Semiautomated Building Extraction Based on CSG Model-Image Fitting

Yi-Hsing Tseng and Sendo Wang

Abstract
Building extraction based on pre-established models has been recognized as a promising idea for acquiring 3D data for buildings from aerial images. This paper proposes a novel building extraction method developed from the concept of fitting CSG (Constructive Solid Geometry) primitives to aerial images. To be practicable, this method adopts a semiautomatic procedure, carrying out high-level tasks (building detection, model selection, and attribution) interactively by the operator and performing optimal model-image fitting automatically with a least-squares fitting algorithm. Buildings, represented by CSG models, can be reconstructed part by part after fitting each parameterized CSG primitive to the edge pixels of aerial images. Reconstructed building parts can then be combined using CSG Boolean set operators. Consequently, a building is represented by a CSG tree in which each node links two branches of combined parts. This paper demonstrates ten examples of building extraction from aerial photos taken at a scale of 1:5,000 and scanned at a pixel size of 25 μm. All of the tests were performed in the prototypical system implemented in a CAD-based environment cooperated with a number of specially designed programs. The process time for each primitive is about 20 seconds and the successful rate of model-image fitting was about 90 percent. Evaluated with some check points, the fitting accuracy was about 0.3 m horizontally and 1 m vertically. The test results are encouraging and promote the theory of model-based building extraction.

Introduction
In response to the development of 3D City Spatial Information Systems for urban planning and management, acquisition of 3D data of city objects has become a topic of increasing importance. This tendency leads to intense research activities aiming for automatic or semiautomatic building extraction from digital aerial images in both the photogrammetry and the computer vision communities (Mohan and Nevatia, 1989; Braun et al., 1995; Englert and Gülich, 1996; Lang and Förstner, 1996; Shufelt, 1999; Vosselman and Veldhuis, 1999; Grün, 2000; van den Heuvel, 2000). While the task of building extraction may differ in terms of image data type and scale, object complexity, required level of detail, and type of product, the common process sequence would be detection, reconstruction, and attribution. Various approaches have been implemented with emphasis on more or less automation with respect to the process sequence.

Model-based building extraction (Sester and Förstner, 1989; Braun et al., 1995; Vosselman, 1998; Brenner, 1999; Fischer et al., 1999; Ameri, 2000; Suveg and Vosselman, 2000; van den Heuvel, 2000) starts with a hypothesis of a building model representing a specified target in the scene and verifies the compatibility between the model and the existing image data. Approaches to model-based building extraction are mostly implemented in a semi-automatic manner, solving the model-image fitting problem based on some high-level information given by the operator. The spatial data of a building object are determined, when model-image fitting is achieved. In contrast to the traditional point-by-point mapping procedure, model-based building extraction features object-based data acquisition. Although the idea and benefits of model-based building extraction have been acknowledged, the working principle is not well established. Therefore, the focus of this study is to establish a practical theory for model-based building extraction.

Building modeling and model-image fitting are the key issues in model-based building extraction. The issue on building modeling is how to establish a set of representative and complete building models. This paper reviews some building model schemes known in the field of digital photogrammetry and discusses how CSG modeling is employed in the proposed method. The issue in model-image fitting is how to develop a computer algorithm that is able to determine the pose and shape parameters of an object model such that the edge lines of the wire frame, as projected into the images, are optimally coincident with the corresponding edge pixels. It is assumed that the image orientations are known and that the pose and shape parameters are approximately determined through an interactive manual process. To deal with this problem, this paper proposes a tailored least-squares model-image fitting algorithm as the key component of the building extraction framework.

In sum, this paper proposes a semiautomatic approach to model-based building extraction from multiple aerial images. This approach is developed with the prospects of releasing the operator from tedious point measurement and efficiently delivering precise and reliable results. Ten buildings were extracted from the test data for demonstration. The experimental results were investigated with regard to model availability, working efficiency, needed constraints, success rate of model-image fitting, and fitting accuracy.

Related Work
In the last decade, building extraction has been a topic of active research. There are a number of helpful reviews and paper collections available (Grün et al., 1995; Förstner and Plümer, 1997; Grün et al., 1997; Grün and Nevatia, 1998; Förstner et al., 1999). An overview of the different approaches reveals that any method for building extraction should incorporate some sort of...
building models. This concept is important because the projection of 3D objects into 2D images leads to a loss of relevant information for building extraction. Furthermore, the use of a building model is essential if building boundaries in the images are confused with irrelevant information, such as vegetation, cars, and building details. Although existing model-based approaches were motivated by the same concept, they differ with respect to the building modeling scheme employed and the strategy of reconstruction. Therefore, related research work on building extraction can be reviewed based on the categories of employed models and reconstruction strategies. Employed models can be generally categorized into polyhedral, prismatic, parameterized polyhedral, and CSG. However, the categorization of reconstruction strategies is somewhat vague. One might divide them into fully automated and semi-automated methods, but the ways to derive a building shape and to match a model with images are quite diverse.

Building Model Schemes

Polyhedral models are boundary representations of objects, which provide the most flexible approximation for the representation of buildings (Henricsson et al., 1996; Huang and Trinder, 1999; Brenner, 1999; Grün, 2000). However, the use of a general polyhedral model for building extraction is problematic due to the absence of semantic classification of building shapes. Not only could the flexibility of modeling cause the reconstruction procedure to be very tricky, but it is also difficult to make predictions about occluded parts. Introducing geometric or object constraints, such as parallelism, coplanarity, and perpendicularity of object faces, may partly solve those problems. However, it introduces the problems of setting and weighting constraints.

Prismatic models, extended from the traditional 2D mapping concept, allow the representation of buildings with arbitrary ground plans, but are restricted to vertical walls and flat roofs (Nevatia and Price, 1982; Mohan and Nevatia, 1989; Lang et al., 1995; Weidner and Förstner, 1995; Hendrickx et al., 1997). They seem to be a special case of polyhedral models. Although the possible building shape is confined under the circumstance, buildings are not further classified and do not have any specific parameterization. Therefore, prismatic models suffer from the same lack of specific building knowledge as do the general polyhedral models.

Parameterized polyhedral models are representations of objects deduced from a set of predefined polyhedral shapes by giving values to the associated parameters (Jaynes et al., 1997; Shufelt, 1999; Vosselman and Veldhuis, 1999). The predefined shapes can be some common building shapes, such as rectangular box, gable roof, L shape, and T shape. Each shape model is associated with some shape parameters to adjust its length, width, height, arm length, and so forth as well as the pose parameters to determine its location and orientation. Shape models implicitly contain object constraints, so that partially occluded buildings can be fully reconstructed and a reconstructed building implies its classification. The major drawback of parameterized models is the lack of flexibility with respect to diverse building shapes. To establish a complete model database for all kinds of buildings would be almost impossible.

CSG models, instead of modeling a full building block, are composed of a combination of volumetric primitives. It is possible to model a complex building with a very small set of primitives, depending on the level of detail required. A primitive is a predefined simple solid model to determine the intrinsic geometric properties of a building part, and is associated with some transformation parameters to perform scaling, rotation, and translation. Boolean set operations, such as union, intersection, and difference, are provided for the combination of primitives. Complex buildings internally are then represented as a CSG tree in which each node links two branches of combined parts. The leaves of the tree are primitives representing building parts. Because CSG models not only provide flexibility for the representation of buildings but also implicitly contain object constraints and classification, many researchers are tending towards using CSG models for building extraction and model-based close-range photogrammetry (Braun et al., 1995; Englert and Gülch, 1996; Lang and Förstner, 1996; Gülch, 1997; Gülch et al., 1998; Veldhuis, 1998).

Semiautomated vs. Fully Automated Approaches

Automation is always a substantial objective in developing computer technologies. However, automation is not practically valuable if the automated technology does not reliably deliver demanded results. Fully automated building extraction is certainly one of the ultimate goals in digital photogrammetry. Many fully automated approaches have been proposed in the past decade (Hsiao and Wong, 1999; Brenner, 2000), but they are more or less only practical for some special cases (Förstner, 1999). Developing a generic fully automated process is still difficult for many reasons. The difficulty stems from the fact that automated image understanding is still operating at a very rudimentary level. Because the task of building extraction may differ in terms of image data type and scale, object complexity, required level of detail, and type of product, ad hoc image understanding algorithms tend to fail whenever a new situation is encountered. Semiautomated approaches, therefore, are currently attracting more and more attentions, attributed to the pressing need of precise, reliable, and complete 3D data (Lang et al., 1995; Gülch et al., 1998; Chio and Wang, 1999; Rottensteiner, 2000).

Image understanding needs a knowledge base (high-level information) and images (low-level data) to work. Processing images to obtain higher-level data is a bottom-up or data-driven procedure, while conversely, a top-down procedure derives features from high-level information and verifies the correspondence between the derived features and data. Frequently, both bottom-up and top-down procedures are applied in an image understanding approach. Information and data meet in a certain data level for the verification of correspondence.

Model-based building extraction from aerial images is a typical example with building models as the high-level information. In general, fully automated building extraction tends to verify the correspondence in the higher data level than do semiautomated approaches. A hypothesis test or knowledge engineering procedure is required for fully automated methods to determine the most appropriate model with respect to the scene. There is, so far, a lack of theory to implement a robust hypothesis test procedure for building extraction. Semiautomated approaches, therefore, let humans decide which building model should be used to avoid doing this high level task, and perform model-image fitting automatically. Because humans can perform high-level tasks much more reliably than computers, and computers do low-level tasks faster than humans, this cooperation would make semiautomated building extraction practically valuable.

Strategy and Workflow

Based on the CSG principle, buildings are modeled as a combination of volumetric primitives. A primitive may represent a building or a part of building, depending on the complexity of building shapes. Buildings with complex topology can be reconstructed by Boolean operations of the building-part primitives in a generic way. The system should provide a model base in which various parametric primitives are included. Each model is associated with a number of shape parameters and pose parameters. The representation of a building part is implemented by setting the values of shape and pose parameters for the representative model. The operator needs to find an appropriate model from the model base corresponding to the
target (building part). Some interactive user interfaces should be provided for the operator to select a model and to perform an approximate fitting between model and images. Then, optimal model-image fitting is performed by the system for each primitive. Both the manual and automatic fitting processes are achieved by adjusting the shape and pose parameters of a model to fit the target images. Having determined all of the building parts, the whole building can be reconstructed by combining the building parts using generic Boolean operations.

Some constraints and local modifications of primitives need to be introduced during the process of combination. Again, interactive user interfaces should be provided for the operator to specify various constraints and modifications. Similar to the One-eye Stereo System proposed by Englert and Gülch (1996), the system featured in the operation of CSG primitives and does not require stereo viewing. However, matching the CSG primitives of a building sequentially to extracted edge pixels from multiple images makes this system distinct from that approach.

The workflow of this approach includes four stages: model selection, approximate fitting, optimal fitting, and primitive combination (Figure 1). In the first stage, the operator detects buildings in a navigation mode. The operator then needs to analyze the building and to divide the building into parts that can be modeled by the primitives predefined in the model base. The second stage is an interactive procedure to revise the shape and pose parameters of a model to approximately fit the interested building part. This procedure can be done by providing the user a dialog window to adjust the shape and pose parameters and to show the model wireframe on the images for checking. The third stage is an automatic fitting procedure. Starting from the approximate fitting, the optimal fitting is achieved iteratively by using the least-squares model-image fitting algorithm. The final stage again is an interactive procedure to combine building primitives using Boolean operators. Some attachment constraints and local modifications can be specified for the combination process. Through the workflow, the system does not require stereo viewing and point measurements.

Building Modeling and Parameters

Buildings show an amazingly high diversity in structure. Categorizing building into distinct styles for representational graphical models is almost impossible. However, regularities are commonly inherent in most building structures, which allow the description of most buildings using a small set of rules. Furthermore, because buildings are volumetric objects (or solids), solid modeling is the most intuitive approach to the representation of buildings. Those characteristics have suggested the use of CSG modeling to many pioneers (Braun et al., 1995; Lang and Förstner, 1996; Gülch, 1997; Gülch et al., 1998; Veldhuis, 1998). Therefore, CSG modeling was adopted for this study.

According to the CSG principle, each primitive should be associated with some parameters to adjust its geometric properties. In this study, those parameters are categorized into shape and pose parameters. The parametric changes would not affect the intrinsic geometric properties. For example, a solid-box primitive is able to represent a rectangular building (or building part) with the shape parameters of length ($l$), width ($w$), and height ($h$), as shown in Figure 2a. By changing the shape parameters, the primitive can be scaled or elongated in each dimension to fit the size of a rectangular building. The primitive will not be allowed to skew, as shown in Figure 2b, unless a skew parameter is introduced. Different primitive models will be associated with different shape parameters, such as the examples listed in Figures 3 and 4. Unlike the shape parameters, pose parameters are not associated with the changes in size or shape, but define the position and orientation of primitives. In a three-dimensional space, it is adequate to use three translation parameters ($dX$, $dY$, $dZ$) and three rotation parameters (tilt, swing, and azimuth ($t$, $s$, $a$)), to depict the position and orientation of an object. However, most buildings should be kept vertical, so that the tilt and swing parameters can be turned off. Therefore, one can use four pose parameters ($dX$, $dY$, $dZ$, $a$) for all kinds of building primitives (Vosselman and Veldhuis, 1999; Suveg and Vosselman, 2000).

A model base, which is a collection of primitives, should be pre-established for building modeling. Because building structures commonly possess some regularity, a small set of...
primitives would be adequate for modeling most buildings. For example, a basic model base may merely contain less than ten models, such as box, wedge, cylinder, and cone (Figure 3). However, for some popular building shapes, which need to be modeled with combined basic primitives, it would be more convenient and efficient to pre-combine basic models to form some advanced models for the model base, for example, models for gable-roof and half-chock buildings (Figure 4).

The structure of each model has to be defined graphically. A model may be described as a polyhedron or a combination of several defined models. A polyhedron is a collection of facets, each facet being a list of vertex indices and each index being a “pointer” into a list of vertices that are defined in a 3D coordinate system. For example, a box model can be graphically depicted in the model coordinate system as shown in Figure 5. The model can be described by the following notation:

$$\text{box} = \{\text{VertexList, FacetList}\}$$

VertexList = \{V1, V2, V3, V4, V6, V7, V8\}

V1 = (0, 0, 0), V2 = (1, 0, 0), V3 = (1, 1, 0), V4 = (0, 1, 0),
V5 = (0, 0, 1), V6 = (1, 0, 1), V7 = (1, 1, 1), V8 = (0, 1, 1)

FacetList = \{F1 = \{V1, V2, V3, V4\}, F2 = \{V1, V5, V6, V2\},...
F6 = \{\ldots\}\}$$

The coordinate systems involved in this approach include model, object, photo, and image coordinate systems. Transformations between coordinate systems can be performed based on associated parameters (Figure 6). A model is defined in the model coordinate system, and can be transformed into the object space in accordance with the shape and pose parameters that represent a building part. The consequence of the transformation will actually show in the changes of the vertex coordinates. However, each vertex is affected by the parameters differently. A box primitive, for example, has the transformation formulas for the eight vertices listed in Table 1. This transformation is equivalent to creating a building object. Through a central projection, the building object can further be transformed into a 2D photo coordinate system in accordance with the known exterior orientation. Furthermore, photo coordinates can be transformed into the image coordinate system in accordance with the interior orientation.

Polyhedron modeling is generally suitable for any model without curved surfaces. Conveniently, one can use a polyhedron to approximate a model with curved surfaces. However, the projected edge lines of the model will be far different from the edge pixels extracted from the images. Under this circumstance, model-image fitting will tend to fail. An appropriate method for the representation and projection of a model with curved surfaces is still needed.

Model-Image Fitting

The principle of model-image fitting is to adjust the shape and pose parameters of a model to fit features extracted from the corresponding images. In this study, features for matching are edge pixels, so that the best fit is achieved by minimizing the sum of the perpendicular distances from the edge pixels to the projected edge lines of the model. Originated with Lowe (1991), the least-squares model-image fitting (LSMIF) is modified to solve this fitting problem. Lowe’s fitting algorithm was developed to solve for the six viewpoint parameters, i.e., the exterior parameters of an image. In this study, however, the viewpoint parameters are known and the model parameters are unknowns to be solved for.

The objective function can be formulated based on the least-squares fit as follows:

$$q = \sum_{i=1}^{j} \sum_{j=1}^{l} \sum_{k=1}^{K} (d_{ijk})^2$$

The summation involves the total number of model edge lines (I), overlapped photos (J), and extracted edge pixels (K). Let an edge line i be projected onto a photo j. The two end points of the projected edge line can be labeled as \(v_{ij}(x_{ij}, y_{ij})\) and \(v_{ij}(x_{ij}, y_{ij})\). If an extracted edge pixel k is from photo j, it is labeled as \(T_{jk}(x_{jk}, y_{jk})\). The distance from the edge pixel to the projected model edge line (Figure 7) can be formulated as

$$d_{ijk} = \sqrt{(y_{ijk} - y_{ij})^2 + (x_{ijk} - x_{ij})^2}$$

Some edge lines may be excluded from the calculation due to self-occlusion. Which edge lines are occluded can be known through the calculation of projection. Given approximate values of the shape and pose parameters, it would also be reasonable that only the edge pixels distributed within the buffer zones (Figure 7) of the projected edge lines are used for calculation. In other words, the buffer zones are used to screen out irrelevant pixels.

In Equation 1, \(q\) is a function of the unknowns of shape and pose parameters. A necessary condition for \(q\) to be minimum is

$$\frac{\partial q}{\partial p_i} = 0; \text{ for all unknown parameters } p_i$$

Equation 3 forms the normal equations of the least-squares solution. In practice, we take the derivative of Equation 2 at each unknown parameter, and form the normal equations using matrix operations. Equation 2 is a non-linear function with respect to the unknown parameters. For the box primitive, Equation 2 can be expressed as a function as follows:
The linearized equations can be expressed in matrix form as  \( \mathbf{V} = \mathbf{A} \mathbf{X} - \mathbf{L} \), where \( \mathbf{A} \) is the matrix of partial derivatives, \( \mathbf{X} \) is the vector of the increments, \( \mathbf{L} \) is the vector of approximations, and \( \mathbf{V} \) is the vector of residuals. The objective function actually can be expressed as  \( \mathbf{q} = \mathbf{V}^T \mathbf{w} \). For each iteration, \( \mathbf{X} \) can be solved for by the matrix operation  \( \mathbf{X} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{V} \).

The buffer size plays an important role in the solution. Using a narrow buffer confines the solution to a small range of convergence, i.e., a small pull-in range, so that it requires good initial values to obtain correct results. Using a wider buffer would offer the larger pull-in range, but it also increases the probability of using irrelevant edge pixels. It is a dilemma as to whether to use a narrow or a wide buffer. To keep its balance, we initiate the computation with a large buffer (say, 30 pixels) and gradually decrease the buffer size with respect to the iterations down to the minimum buffer size (say, five pixels). The procedure of decreasing buffer would work for most cases. However, further investigations are required to know how much the buffer size influences the solution and what is the best way to manipulate the buffer size.

Inadequate relevant image features, affected by irrelevant features or noise, or given bad initial approximations, may lead the computation to a wrong solution. One way to fix this problem is to introduce certain constraints to the solution. Constraints are especially needed if the existing edge pixels are not adequate to resolve some unknowns. In this case, the constrained parameters should be determined by additional observations. Constraints are also needed when the working model is attached to a previously determined model as described in the next section. The constraints can be formulated for each parameter. For instance, the constraint for \( w \) can be

\[
(w_{\text{obs}} - w^0) + d_w = \Delta w
\]

in which \( w_{\text{obs}} \) is the observed value, \( w^0 \) is the initial value, and \( d_w \) is the residual of the parameter. The constraint equations can be combined with Equation 5 to find the solution with constraints. Constraints to the parameters are not necessarily
firmly fixed. Elastic constraints can be implemented by assigning appropriate weights to Equation 6.

**Boolean Set Operations and Local Modification**

In this study, building parts are determined one by one, which should be combined to form a complete building using Boolean set operations, such as union (∪), intersection (∩), and difference (−). Attachment of building parts is a case of union that needs special care. When attached building parts are determined independently, they may not be connected well but may have a discrepancy or overlap in between due to fitting errors or uncertainties. An attachment operation could incorporate some constraints in the fitting process or make some local modifications to ensure that a connection between two primitives makes sense. In order to maintain the intrinsic geometric properties, adjustment for attachment should work on shape or pose parameters.

Attachment constraints for combining building primitives can be categorized into facet-to-facet, edge-to-edge, and orientation alignment constraints. A facet-to-facet constraint is needed for side-by-side connections of two building primitives, such as the example in Figure 8a. The way to implement this constraint is to set the \( dZ \) parameter of the top building part as \( dZ_{\text{top}} = dZ_{\text{bottom}} + h_{\text{bottom}} \). When two primitives need to connect side by side, and some edges of the connected facets need to overlap, edge-to-edge constraints should be applied. Figures 8b and 8c are examples requiring edge-to-edge constraints, which can be implemented by combining a facet-to-facet constraint and an orientation constraint, e.g., \( a_{\text{left}} = a_{\text{right}} \) or \( a_{\text{top}} = a_{\text{bottom}} \). Orientation alignment is also frequently required to obtain a reasonable connection, for example, when a building with the structure of stacking-up boxes inherits orientation alignment among the boxes as shown in Figure 8d. In this study, building parts are reconstructed sequentially, so that the attachment constraints can be derived from a previously reconstructed part and be set for the new part by using the LSMIF constraint as described in Equation 6.

**Experiments**

**Implementation**

The proposed approach is implemented in a CAD-based environment combined with a model-image fitting program. The methods for model selection, approximate fitting, and visualization are implanted in the AutoCAD™ system using Visual Basic for Application (VBA) programming. The LSMIF is currently an independent function developed using C code. However, it would be possible to combine all of the processes into a single system.

Figure 9 shows the designed working environment. The graphical user interface allows the operator to zoom and view overlapped images in the image windows and to pick a suitable primitive from the model icons listed in the left column. Therefore, the operator can perform model selection and approximate fitting in this environment and visually supervise the fitting procedure. Each primitive is associated with an anchor point which is a point located at the origin of the model coordinate system. By specifying the projected positions of the anchor point (the circled corner of the model shown in Figure 9) on the images, the approximate location of the model in the object space can be determined. Furthermore, one can obtain the approximate scale and orientation of the model by specifying the two neighbor corners of the anchor point in one image. On the screen, the lower-left and lower-right windows show the top and perspective views of the model, respectively, in the object space. The model wire frame is also superimposed on the images. The approximate fitting results serve as the input data to the LSMIF. The LSMIF updates the shape and pose parameters to obtain the best fit between model and images.

**Experiments of Model-Image Fitting**

The first demonstrated example is reconstructing a building modeled by a box primitive. Figure 10a shows the image pair...
TABLE 2. THE QUANTITATIVE ASSESSMENT OF THE FITTING RESULTS OF THE BOX PRIMITIVE

<table>
<thead>
<tr>
<th>Parameters</th>
<th>( l ) (m)</th>
<th>( w ) (m)</th>
<th>( h ) (m)</th>
<th>( \theta ) (degree)</th>
<th>( dX ) (m)</th>
<th>( dY ) (m)</th>
<th>( dZ ) (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial values</td>
<td>7.235</td>
<td>24.738</td>
<td>15.221</td>
<td>6.0965</td>
<td>169208.838</td>
<td>2544552.792</td>
<td>23.495</td>
</tr>
<tr>
<td>Differences</td>
<td>-0.118</td>
<td>-0.175</td>
<td>0.385</td>
<td>0.0970</td>
<td>0.147</td>
<td>0.169</td>
<td>0.101</td>
</tr>
</tbody>
</table>

Assessment, the results are also compared with the model parameters derived from manually measured data with an analytical plotter. Table 2 shows the data of the model parameters of the initial state and the final fitting, the checking data, and the differences between the fitting results and the checking data. The largest error appeared in the parameter of building height \( h \), mainly due to self-occlusion in the images. The edge lines of the top and bottom facets should determine the building height, but only one edge line of the bottom facet can be seen in the images.

The second example is reconstructing a gable-roof building. Figure 11a shows the image pair superimposed with the extracted edge pixels and the projected wire frames of the gable-roof primitive after approximate fitting. Figure 11b shows the fitting results of the 

Test and Accuracy Assessment

The test data are drawn from a set of aerial photos covering the National Cheng Kung University campus. The camera focal length is 305 mm and the flight height was about 1500 m, so that the photo scale is about 1:5,000. The photo endlap is about 60 percent and sidelap is about 30 percent. The photos were digitized with a pixel size of 25 \( \mu \)m. In the test, ten buildings were reconstructed using the corresponding stereo pairs of the images. Figure 12 shows the results of the ten examples. With regard to model availability, most of the buildings can be properly represented by a combination of box and gable-roof primitives. In fact, building modeling implies a certain simplification of the real building shape. Although the models used in the examples 8 and 9 (Figures 12h and 12i) do not quite fit the real building shape, to a certain extent the representations are reasonable, especially when very simplified building modeling is required. One certainly can elaborate the building representation if there are more suitable primitives available. Except for the first example, the other buildings are modeled with two or more than two primitives. It can be seen that the CSC modeling is very adaptive to complex buildings. The proposed model-image fitting function is quite efficient. For each primitive, it takes about 20 seconds to go through the procedure. It is much faster than point-by-point manual measurement. The successful rate of the LSMIF process is about 90 percent (24 out of 27 primitives). By introducing proper constraints, all cases can be solved and proper topology between connected primitives can be maintained. From the extracted spatial information of buildings, the coordinates of building corners were derived to com-

TABLE 3. THE QUANTITATIVE ASSESSMENT OF THE FITTING RESULTS OF THE GABLE-ROOF PRIMITIVE

<table>
<thead>
<tr>
<th>Parameters</th>
<th>( l ) (m)</th>
<th>( w ) (m)</th>
<th>( h ) (m)</th>
<th>( \theta ) (degree)</th>
<th>( dX ) (m)</th>
<th>( dY ) (m)</th>
<th>( dZ ) (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fitting Results</td>
<td>8.928</td>
<td>31.997</td>
<td>10.246</td>
<td>1.398</td>
<td>92.581</td>
<td>169346.466</td>
<td>2544057.869</td>
</tr>
<tr>
<td>Check Data</td>
<td>9.053</td>
<td>32.119</td>
<td>9.670</td>
<td>1.500</td>
<td>92.710</td>
<td>169346.673</td>
<td>2544057.928</td>
</tr>
<tr>
<td>Differences</td>
<td>-0.125</td>
<td>-0.122</td>
<td>0.576</td>
<td>-0.102</td>
<td>-0.129</td>
<td>-0.207</td>
<td>-0.059</td>
</tr>
</tbody>
</table>
Figure 12. (a) to (j) sequentially show the ten test results. For each example, the stereo images are shown with superimposed model wire frames in the upper, and the top and perspective views in the lower. (Color version at www.asprs.org.)
pare with manual measurements of the corresponding points. Table 4 shows the average and RMS differences of the coordinates. Large differences appeared mostly due to confusing edges or self-occlusion. This empirical accuracy analysis provides evidence that the new approach generates qualified 3D data of buildings.

Conclusions

The proposed model-based building extraction from aerial images performs well and exhibits its potential in acquiring 3D data of buildings. The following aspects characterize the proposed approach:

- It uses a semi-automatic procedure to combine the human ability of image understanding with the number-crunching capacity of computers;
- In contrast with the traditional point-by-point digitization mapping process, this approach promotes an object-by-object data acquisition procedure;
- It does not require stereo viewing or measurement;
- It employs CSG modeling so that complex buildings can be modeled with a small set of primitives; and
- The final products are CSG building representations, which have the prospects of providing the fundamental data for a 3D city spatial information system.

The experiment results demonstrate the reliability of the method. Most buildings were successfully reconstructed in spite of information loss or confusion due to occlusion or mixed with other objects. The test shows a 90 percent success rate of reconstruction. Checked with manually measured data, the accuracy of the test results was also evaluated. The fitting accuracy is about 0.3 m in horizontal, which conforms to the requirement of large-scale mapping. The vertical accuracy is about 1 m, much worse than the horizontal accuracy due to the small base-height ratio of the stereo images (about 0.3). The fitting process may not work if the bottom edges of the building are unclear or occluded. Under this circumstance, constraints on the parameters $dZ$ or $h$ are needed. In sum, the overall performance of the proposed method proves the advantage of using models for building reconstruction. The test results are encouraging and reveal the potential of promoting the proposed method as a practical system.

Acknowledgments

This research project was sponsored by the National Science Council of the Republic of China under NSC grants 90-2211-E-006-103 and 91-2211-E-006-092. The authors are grateful to the referees for their constructive comments on the original manuscript.

References


TABLE 4. The Average and RMS Differences of the Building-Corner Coordinates Derived from Building Extraction Method and Manual Measurement

<table>
<thead>
<tr>
<th></th>
<th>X (m)</th>
<th>Y (m)</th>
<th>Z (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Diff.</td>
<td>0.161</td>
<td>0.007</td>
<td>0.047</td>
</tr>
<tr>
<td>RMS Diff.</td>
<td>0.330</td>
<td>0.277</td>
<td>1.034</td>
</tr>
</tbody>
</table>


(Received 21 August 2001; accepted 29 July 2002; revised 26 September 2002)