ON THE ACCURACY ASSESSMENT OF LEAST-SQUARES MODEL-IMAGE FITTING FOR BUILDING EXTRACTION FROM AERIAL IMAGES

Sendo WANG[1] Yi-Hsing TSENG[2]

Department of Surveying Engineering, National Cheng Kung University
No. 1 University Road, Tainan 70101
Taiwan, R.O.C.

KEYWORDS: Least-squares Model-image Fitting, Accuracy Assessment, Model-based Building Extraction, Constructive Solid Geometry, Digital Photogrammetry

ABSTRACT: *Model-based building extraction* from aerial images has been an intensive research topic in the field of digital photogrammetry since the last decade. Based on our previous research work, the principle of *Constructive Solid Geometry* (CSG) is applied to model various buildings. Each building is represented by a combination of 3D primitives and each primitive is associated with a set of shape and pose parameters. Building reconstruction is implemented by adjusting the model parameters to fit model with images. We proposed a *Least-squares Model-image Fitting* (LSMIF) algorithm to obtain the optimal fit between model and images. In this paper, the performance of LSMIF is investigated. First, the convergence rate and pull-in range of the algorithm is analyzed. Then, how to use constraints to increase the convergence rate is introduced. Finally, 10 sets real data were tested to assess the theoretical accuracy of parameter determination. In order to assess the empirical accuracy, the results are also compared with manually measured data using an analytical plotter. This study reveals that the function of LAMIF is table and can generate qualified 3D information of building comparable to manually measured data.

1. INTRODUCTION

3D spatial information is essential for a wild range of applications, such as city planning, architecture designing, tourist guiding, or personal navigating system. Although each application requires diverse spatial information, buildings are the very required objects in common (Gülch, 1996). To efficiently extract buildings from aerial images, several full-automated or semi-automated approaches have been proposed by experts both in photogrammetry and computer vision domains (Förstner, 1999; Grün, 2000). Model-based building extraction, extracting buildings by fitting pre-defined volumetric models with images, has been recognized as a convincing approach to efficient building extraction that could be implemented in practice (Braun, *et al.*, 1995; Chapman, *et al.*, 1992; Lang and Förstner, 1996; Veldhuis, 1998).

Model-based building extraction relies on a model-image fitting algorithm to obtain the optimal fit between model and images. Attempts to solve the problem of model-image fitting date back to the work of Sester and Förstner (Sester and Förstner, 1989). By fitting projected model to image, the transformation parameters of a building model are determined using a clustering algorithm followed by a robust estimation of model parameters. This budding research work has marked an important step toward model-based building extraction (MBBE), although the algorithm is restricted to fit a model to single image rather than multiple images. Concurrently developed in the field of computer vision for model-based vision, Lowe (Lowe, 1991) proposed a least-squares model-image fitting to solve for projection and model parameters that will best fit a 3D model to matching 2D image features. Lowe's study set up the fundamental theory of the least-squares model-image fitting (LSMIF) for generic applications. This rigorous fitting algorithm has been recognized as a key to deal with MBBE (Veldhuis, 1998; Vosselman, 1999). However, to apply LSMIF for MBBE, we need to tailor the generic LSMIF theory to meet the specific situations of building modeling and model parameters. Vosselman and Veldhuis (Vosselman and Veldhuis, 1999) modified Lowe's algorithm to improve pull-in range, but they did not clearly define the building models and the fitting functions.

Based on our previous research work (Tseng and Wang, 2000; Wang and Tseng, 2001; Chou, *et al.*, 2001; Lin, *et al.*, 2001), the principle of constructive solid geometry (CSG) is applied to model various buildings. Each primitive is associated with a set of model parameters that can be categorized into shape and pose parameters. Building reconstruction is implemented by adjusting the model parameters to fit model with images. Therefore, an LSMIF algorithm was proposed to fit a primitive to the corresponding building part in the image, which is based on the minimization of the squares sum of the distances from the edge pixels extracted from images to the corresponding edge of the projected primitive.

The motivation of this study is to evaluate the performance of the LSMIF algorithm. Ten various buildings, which can be represented by a combination of box and gable-roof primitives, were constructed using the algorithm for testing. Based on the test data, we first investigate the convergence rate and pull-in range to see how often the

[1] Ph.D. student, TEL: +886-6-2370876 ext 834/835, E-mail: sendo@sv.ncku.edu.tw
[2] Associate professor, FAX: +886-6-2375764, E-mail: tseng@mail.ncku.edu.tw

algorithm can deliver correct results. In order to increase the convergence rate, the method to add some constraints into the system is also introduced. The accuracy assessment is performed in both the theoretical and empirical ways. Theoretical accuracy can be estimated by the least-squares theory and empirical accuracy is obtained by comparing the results with manually measured data using an analytical plotter.

2. LEAST-SQUARES MODEL-IMAGE FITTING

2.1 Definition of Primitives

A primitive is a pre-defined simple solid model, which determines the intrinsic geometric property of an object part, and is associated with a set of parameters that can be categorized into shape parameters and pose parameters. The shape parameters describe the shape size of the primitive, e.g., a box primitive has three shape parameters: length (l), width (w), and height (h). Changing the values of shape parameters elongates the primitive in the three dimensions and keeps its shape as a rectangular box. Each primitive would be associated with different shape parameters, e.g., a gable-roof house primitive has an additional shape parameter – roof's height (rh). The pose parameters describe the position and attitude of a primitive in the object space. It is adequate to use 3 translation parameters (dX, dY, dZ) to depict the position and 3 rotation parameters, tilt (t) around Y-axis, swing (s) around X-axis, and azimuth (α) around Z-axis to represent orientation of a primitive. However, walls of the building are supposed to be vertical in most circumstances, the tilt and swing angles can be omitted. Unlike the shape parameters, the four pose parameters are suitable for all kinds of primitives. Figure 1 shows the parameters of a box primitive, the X'-Y'-Z' coordinate system is the model space, and the X-Y-Z coordinate system is the object space. The shape parameters of a gable-roof house primitive are illustrated in Figure 2.

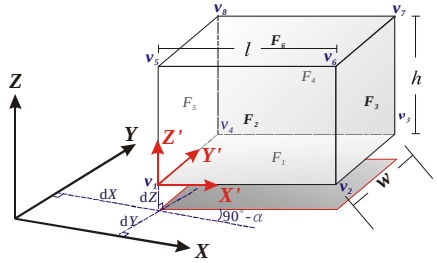


Figure 1: Shape and pose parameters of a box primitive.

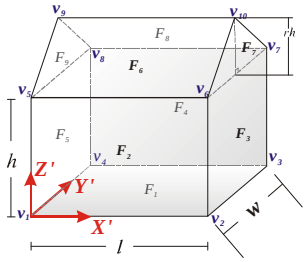


Figure 2: Shape parameters of a gable-roof primitive.

A model can be described by a polyhedral structure. Polyhedron is a composite of faces and each face consists of several vertices. For example, the box primitive, which is a cubic solid in the model space, consists of 6 faces (F_1, F_2, F_3, F_4, F_5, F_6) and 8 vertices (v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8). The model coordinates of the vertices are: v_1=(0,0,0), v_2=(1,0,0), v_3=(1,1,0), v_4=(0,1,0), v_5=(0,0,1), v_6=(1,0,1), v_7=(1,1,1), v_8=(0,1,1). The gable-roof house primitive shown in Figure 2 can be described in the same manner.

2.2 Coordinate Systems

The algorithm performs the fitting in the photo coordinate system. A primitive, however, is defined in the model space. It is necessary to transform a primitive from model space to object space by introducing a set of shape and pose parameters, and then to project the object model onto the photo coordinate system with the known exterior orientation parameters. On the other hand, edge pixels extracted from the images should be transformed to the photo coordinate system for matching. Figure 3 shows the transformation steps of a box from model to photo coordinate system and the edge pixels from image to photo coordinate system.

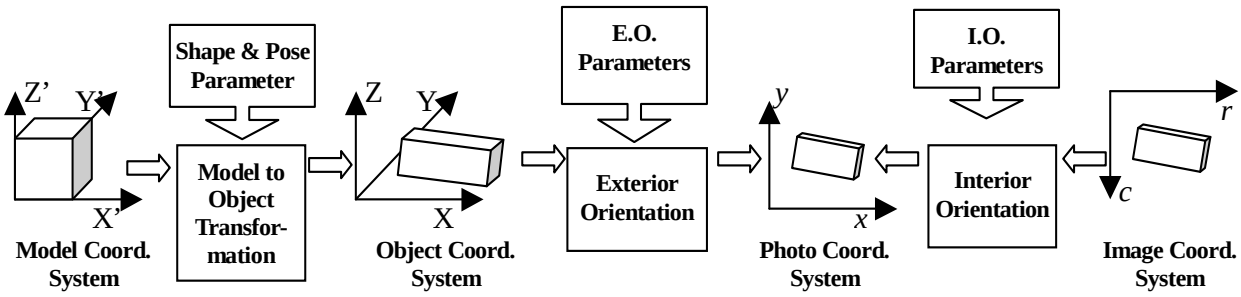


Figure 3: Coordinate systems involved in the fitting algorithm and their relationship.

The primitive defined in the model space is a simple unit solid, e.g. a box primitive is a unit cube which width, length, and height are all equal to 1. The shape parameters will elongate or shorten the box to the correct size, and the pose parameters will rotate and move the box to the correct attitude and position in the object space. Table 1 lists the change of coordinates of the 8 vertices. Each vertex in the object space can be projected to the photo by the collinearity condition equations with the known exterior orientation elements:

$$x = -f \cdot \frac{r_{11}(X - X_0) + r_{12}(Y - Y_0) + r_{13}(Z - Z_0)}{r_{31}(X - X_0) + r_{32}(Y - Y_0) + r_{33}(Z - Z_0)}; \qquad y = -f \cdot \frac{r_{21}(X - X_0) + r_{22}(Y - Y_0) + r_{23}(Z - Z_0)}{r_{31}(X - X_0) + r_{32}(Y - Y_0) + r_{33}(Z - Z_0)} \tag{1}$$

where (x, y) are the photo coordinates of a vertex
(X, Y, Z) are the object coordinates of a vertex
(X_0, Y_0, Z_0) are the object coordinates of the perspective center
f is the focal length of the camera
$r_{11}, r_{12}, r_{13}, r_{21}, \ldots$ are coefficients of the rotation matrix defined by the exterior orientation

Table 1: Vertex coordinates from model space to object space.

Vertex No.	Model Space Coordinate	Multiply Shape Parameters	After Rotation	After Translation (Object Space Coordinate)
v_1	(0, 0, 0)	(0, 0, 0)	(0, 0, 0)	(dX, dY, dZ)
v_2	(1, 0, 0)	(w, 0, 0)	($w$cosα, $w$sinα, 0)	($w$cosα+dX, $w$sinα+dY, dZ)
v_3	(1, 1, 0)	(w, l, 0)	($w$cosα-$l$sinα, $w$sinα+$l$cosα, 0)	($w$cosα-$l$sinα+dX, $w$sinα+$l$cosα+dY, dZ)
v_4	(0, 1, 0)	(0, l, 0)	(-$l$sinα, $l$cosα, 0)	(-$l$sinα+dX, $l$cosα+dY, dZ)
v_5	(0, 0, 1)	(0, 0, h)	(0, 0, h)	(dX, dY, h+dZ)
v_6	(1, 0, 1)	(w, 0, h)	($w$cosα, $w$sinα, h)	($w$cosα+dX, $w$sinα+dY, h+dZ)
v_7	(1, 1, 1)	(w, l, h)	($w$cosα-$l$sinα, $w$sinα+$l$cosα, h)	($w$cosα-$l$sinα+dX, $w$sinα+$l$cosα+dY, h+dZ)
v_8	(0, 1, 1)	(0, l, h)	(-$l$sinα, $l$cosα, h)	(-$l$sinα+dX, $l$cosα+dY, h+dZ)

2.3 Edge Feature Extraction

The transformed model edges are supposed to fit with the edge pixels extracted from the corresponding images. In this paper, edge pixels are extracted using the algorithm of the polymorphic feature extraction module (FEX) developed by Förstner and Fuchs (Förstner, 1994, Fuchs, 1995, Fuchs and Förstner, 1995), which can extract interest points, edges, and regions simultaneously. The extraction module consists of three steps: (1) *Classification* of the pixels into the three feature types, (2) *Localization* of the features, (3) *Approximation* of lines and blob boundaries by analytic functions. By providing the known interior orientation parameters of each image, edge pixels extracted by FEX can be transformed to the photo coordinate system.

The user interface of our semi-automatic approach allows the operator to resize, rotate, and move a model to fit the corresponding building images approximately. This procedure would offer a set of approximate values for the shape and pose parameters, so that the discrepancy between the projected edges of the initial primitive and the extracted edge pixels should be small. Therefore, it is reasonable that the fitting algorithm uses only the edge pixels distributed in a buffer zone of the projected edges shown as Figure 4.

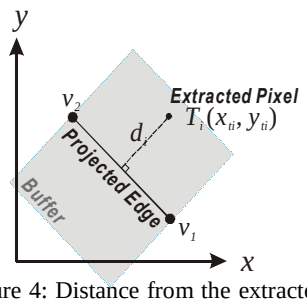


Figure 4: Distance from the extracted pixel to the projected edge and the effective buffer zone.

2.4 Objective Function and Least-squares Adjustment

That the fitting condition we are looking for is the projected edge exactly falls on the building edges in the image. It means that the distance d_i is considered as a discrepancy and is expected to be zero. Therefore, the objective of the fitting function is to minimize the distance between the extracted edge pixels T_i and the projected edge v_1v_2. The projected edge is composed of the projected vertices $v_1(x_1, y_1)$ and $v_2(x_2, y_2)$. Suppose there is an extracted edge pixel $T_i(x_{ti}, y_{ti})$ located inside the buffer, the distance d_i from the point T_i to the edge v_1v_2 can be formulated as the following equation:

$$0 + d_i = \frac{\left|(y_1 - y_2)x_{ti} + (x_2 - x_1)y_{ti} + (y_2x_1 - y_1x_2)\right|}{\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}} \quad \text{where} \quad i \quad \text{is the index of extracted edge pixels.} \tag{2}$$

The photo coordinates $v_1(x_1, y_1)$ and $v_2(x_2, y_2)$ are functions of the shape, pose, and exterior-orientation parameters, where only the exterior-orientation parameters are known. E.g., a box primitive will have 7 unknowns: w, l, h, α, dX, dY, and dZ. Every extracted edge pixel in the buffer may form an objective function as Equation (2). Since d_i represents the discrepancy between model and image, the objective of fitting is to minimize the squares sum of the distances:

$$\Sigma d_i^2 = \Sigma [F_i(w, l, h, \alpha, \mathrm{d}X, \mathrm{d}Y, \mathrm{d}Z)]^2 \rightarrow min. \tag{3}$$

Appling the least-squares adjustment to the fitting function, the distance d_i is considered as the residual errors v_i. Therefore, Equation (2) should be rewritten as follows:

$$0 + v_i = F_i(w, l, h, \alpha, \mathrm{d}X, \mathrm{d}Y, \mathrm{d}Z) \tag{4}$$

Equation (4) is nonlinear and the unknowns cannot be calculated directly. In order to solve the unknowns using Newton-Rapheson method, Equation (4) should be differentiated with respect to the unknown parameter. Then, it may be expressed in the linearized form as:

$$v_i = \left(\frac{\partial F_i}{\partial w}\right)_0 \Delta w + \left(\frac{\partial F_i}{\partial l}\right)_0 \Delta l + \left(\frac{\partial F_i}{\partial h}\right)_0 \Delta h + \left(\frac{\partial F_i}{\partial \alpha}\right)_0 \Delta\alpha + \left(\frac{\partial F_i}{\partial dX}\right)_0 \Delta dX + \left(\frac{\partial F_i}{\partial dY}\right)_0 \Delta dY + \left(\frac{\partial F_i}{\partial dZ}\right)_0 \Delta dZ + F_{i0} \quad (5)$$

In Equation (5), F_{i0} is the approximation of the function F_i evaluated by the given initial value of the 7 unknowns - shape and pose parameters. $\left(\frac{\partial F_i}{\partial w}\right)_0$, $\left(\frac{\partial F_i}{\partial l}\right)_0$, etc., are the partial derivatives of F_i with respect to the indicated unknowns evaluated at the initial approximations. Δw, Δl, Δh, etc., are increments of the unknowns applied to the initial approximations.

Each extracted edge pixel can form such an equation for the corresponding projected edge. Therefore, a visible edge may have a group of observation equations. All of these equations can be summarized and expressed by the matrix form: $\boldsymbol{V}=\boldsymbol{AX}-\boldsymbol{L}$, where $\boldsymbol{A}$ is the matrix of partial derivatives and $\boldsymbol{X}$ is the matrix of increments of shape and pose parameters. Once an edge pixel is added, it adds a row to matrix $\boldsymbol{A}$, $\boldsymbol{V}$, and $\boldsymbol{L}$. A box primitive, which has up to 9 edges are visible, may have 9 groups of equations for one single photo. It is easy to solve the matrix $\boldsymbol{X}$ by the matrix operation: $\boldsymbol{X}=(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{PA})^{-1}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{PL}$. If the weights of all unknowns are assumed equal, $\boldsymbol{P}$ is an identical matrix. The solution of $\boldsymbol{X}$ is the increments of shape and pose parameters. If any element of matrix does not meet the requirement of the threshold, adding the increments on the previous parameters to regenerate the new matrix $\boldsymbol{A}$ and $\boldsymbol{L}$, then implement the adjustment again. The iteration continues until the increments converge below the thresholds or diverge beyond the limits. When the iteration converges, the primitive parameters are determined and the model is fitted to the image. If a building is visible on multiple photos, it can provide more observation equations. These redundant observation equations sometimes are very useful especially when an edge is occluded on one photo but visible on another photo. All of the observations on different photos join the least-squares adjustment to solve the model parameters simultaneously

3. CORRECTNESS AND ACCURACY ASSESSMENT

In order to study the performance of the LSMIF algorithm, we choose 10 various buildings in the NCKU campus from the digitized aerial photos. Except building 1 is modeled by one single primitive, the others are composed of two or more than two primitives. For accuracy assessment, an experienced photogrammetric operator measured all of the visible building corners.

3.1 Pull-in Range Test

The pull-in range is the range from the initial value to the convergent value of each parameter. It can be taken as a reference of the maximum error that the fitting function can tolerate, or the least accuracy the initial value should achieve. The part 1 of building 5 (Figure 5) was chosen for the pull-in range test. A box primitive is selected to fit the building part, and the shape and pose parameters have been solved by LSMIF. The pull-in-range of each parameter is tested on by one by adding a certain amount of error onto the correct number until the iteration divergent. The pull-in ranges of the 6 parameters are shown in Figure 6.

Figure 5: The building and the box primitive for the pull-in range test.

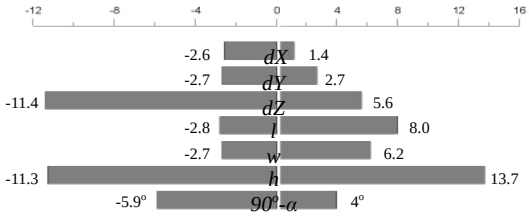
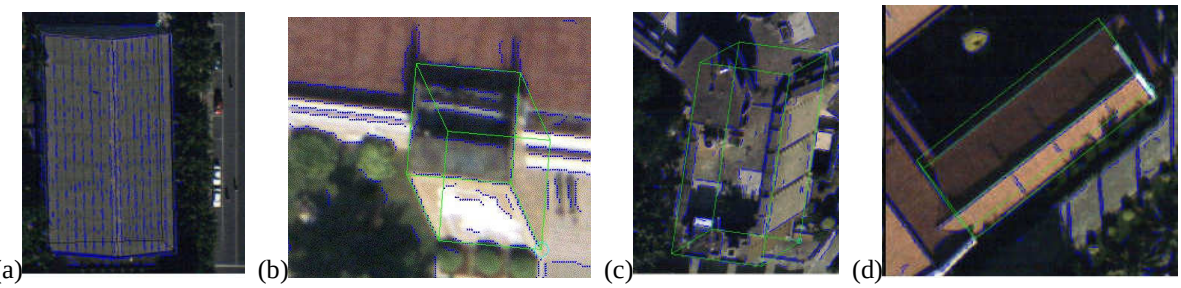


Figure 6: The pull-in range of the each parameter.

Due to the influence of irrelevant information to the building or self-occlusion, the pull-in ranges could be case by case. In this case, one can find that the two parameters: *dZ* and *h*, have better pull-in ranges. It is because the 4 edges on the roof are extracted clearly and successfully, which provides a good constraint to the parameter *h*, especially in the positive direction. There are also 2 bottom edges are clearly visible in the left photo, which provides the constraint to *dZ* and *h* both in the negative direction. The pull-in range of *dZ* in positive direction is not as well as *h* is due to the noise pixels extracted on the roof of part 2 of the same building (lower-right in the photos). The parameter *dX* has a narrowest pull-in range among other parameters. It is because some trees occlude the bottom edge on the right side of the building, and the boundary of trees is extracted as edge pixels. When the *dX* increases, the right edge of the box primitive will be fitted to the boundary of trees and results in the wrong solution.

3.2 Correctness Analysis and Constraints

Because the LSMIF algorithm fits the model to the extracted edges in images, the correctness of fitting depends on the correctness of the edge extraction. Any edge pixels do not formed by the building may cause incorrect fitting results. Figure 7 illustrates some of these circumstances.



(a) (b) (c) (d)

Figure 7: Four examples of incorrect fitting results caused by (a)the texture of the roof, (b)the shadow of the building, (c)the balconies, and (d)the gable-roof is not symmetric.

There are two approaches to fix incorrect fittings. The first approach is to give a set of very accurate initial parameters with the use of a narrow buffer. However, it may not work if the real edge is occluded or ambiguous. Besides, the use of narrow buffer tends to cause LSMIF divergent if the buffer does not cover the wanted edge pixels. The second approach is to add constraints to the LSMIF. If a certain parameter has been known from other data source, one can treat the parameter as an observation and add an observation function to the least-squares adjustment. By raising the weight of the observation, the parameter will tends to be fixed. This is especially useful in the situation of information loss. For example, shadows or other buildings have occluded the bottom edges of the primitive. If the nearby ground height can be measured, the parameter of *dZ* can be determined and fixed. Figure 8 shows an instance in which the bottom edges are invisible but the fitting can be carried out successfully by adding constraints to *h* and *dZ*. Therefore, the building can be extracted completely. This is one of the great benefits that make MBBE superior to the traditional stereo point-by-point measurement.

In the case study, the 10 buildings can be decomposed into 23 building parts and each can be modeled either by box or gable-roof primitives. The fitted primitive is projected onto the images as a wire-frame for checking. Table 2 lists the checking results. The extraction of building 4 was incorrect because the bottom and the wall edges are almost invisible. But this can be improved by introducing the constraints of *dZ* and *h* into the LSMIF. The percentage of correct results is about 89%, which proves the flexibility of LSMIF.

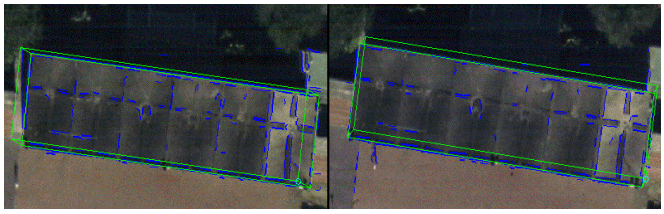


Figure 8: The fitting results with constraints.

3.3 Accuracy Assessment

The fitting results have been compared with the manual measurement data. For comparison, the 3D vertex coordinates of each primitive are derived from the shape and pose parameters. After eliminating the incorrectly fitted edges, the coordinate difference between LSMIF and the manual measurement represents the empirical accuracy of fitting. The mean values and standard deviations of the differences are listed in Table 3. According to the standard deviations of the differences, the empirical accuracy in each coordinate will be: 33cm(X), 28cm(Y), and 103cm(Z). Figure 9 depicts the distribution of the differences on the X-Y plane. The mean values are close to zero, so that no systematic errors are expected. The empirical accuracy in X-Y plane is about 40cm. ΔZ has a larger standard deviation because most of the bottom edges of the buildings are invisible. Therefore, the empirical accuracy in 3D was downgraded to about 1m.

Table 2: The correctness check of LSMIF.

Building ID	Part ID	Primitive Type	Vertex Number	LSMIF Results	
				Correct	Incorrect
1	1	Gable-roof	6	6	0
2	1	Box	4	4	0
	2	Box	4	4	0
3	1	Box	5	5	0
	2	Box	5	5	0
4	1	Box	6	2	4
	2	Gable-roof	4	0	4
5	1	Box	4	4	0
	2	Box	4	4	0
6	1	Box	4	4	0
	2	Box	4	4	0
7	1	Gable-roof	6	6	0
8	1	Box	2	2	0
	2	Box	2	2	0
	3	Box	2	1	1
9	1	Gable-roof	5	5	0
	2	Box	4	4	0
	3	Box	4	2	2
	4	Box	4	4	0
	5	Box	5	5	0
10	1	Box	4	4	0
	2	Box	4	4	0
	3	Box	4	4	0
Total			96	85	11
Percentage				88.54%	11.46%

Table 3: The statistics of differences from manual measurement to LSMIF results.

	ΔX	ΔY	ΔZ	$\sqrt{\Delta X^2 + \Delta Y^2}$	$\sqrt{\Delta X^2 + \Delta Y^2 + \Delta Z^2}$
Maximum (m)	0.983	0.723	3.034	1.072	3.047
Minimum (m)	-0.733	-0.985	-1.932	0.045	0.138
Average of Absolute Values (m)	0.291	0.223	0.799	0.397	0.942
Average (m)	0.161	0.070	0.047	0.397	0.942
Standard Deviation (m)	0.3296	0.2770	1.0335	0.2400	0.6233

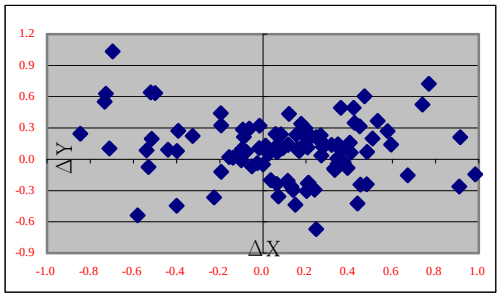


Figure 9: The error distribution on the X-Y plane.

4. CONCLUSIONS

A least-squares model-image fitting algorithm is introduced and evaluated for the use of model-based building extraction from aerial images. This algorithm iteratively fits the pre-defined primitive to the aerial photos and solves the optimal model parameters with known orientation parameters of photos. It is able to use multiple photos simultaneously, which is helpful when some edges are visible in one image and others are visible in the second or other image. By adding constraints, such as ground height and roof height, the building can be entirely extracted even its bottom is occluded. Furthermore, the LSMIF extracts buildings object by object, so that it would be more efficient than point-by-point measurements.

The performance of the LSMIF algorithm is evaluated by analyzing the pull-in range of parameters, the correctness and the empirical accuracy of the fitting. In the pull-in range test, it shows that there is at least a ±2m pull-in range for each parameter in average. It means that the operator does need to provide very accurate initial approximation of parameters for fitting in common situations. According to the results of 10 study cases, the fitting algorithm is functioning very well and efficient. The empirical horizontal accuracy is comparable to manual measurements. The vertical accuracy is low due to shadows and occlusion. The accuracy and reliability would be able to be improved if more overlapped photos are used or more constraints are introduced. In general, the LSMIF delivers satisfying results and has a convincing potential to be applied for MBBE.

REFERENCES

Braun, C., T. H. Kolbe, F. Lang, W. Schickler, V. Steinhage, A. B. Cremers, W. Förstner and L. Plumer, 1995. Models for Photogrammetric Building Reconstruction, *Computers & Graphics*, 19(1), pp. 109-118.

Chapman, D. P., A. T. D. Deacon, A. A. Hamid and R. Kotowski, 1992. CAD Modeling of Radioactive Plant - The Role of Digital Photogrammetry in Hazardous Nuclear Environments, *Proceedings of The XIXth Congress of The International Society for Photogrammetry and Remote Sensing*, Vol. XXIX, pp. 741-753.

Chou, H., Y.-H. Tseng and S. Wang, 2001. Least-squares CSG Model-image Fitting, *Proceedings of the 20th Conference on Surveying Academy and Applications*, Chungli, Taiwan, Vol. 20.

Förstner, W., 1994. A Framework for Low Level Feature Extraction, *Computer Vision*, Springer-Verlag, pp. 383-394.

Förstner, W., 1999. 3D-City Models: Automatic and Semiautomatic Acquisition Methods, *Photogrammetric Week '99*, Stuttgart, Vol. 47, pp. 291-304.

Fuchs, C., 1995. Feature Extraction, *Second Course in Digital Photogrammetry*, IFP University Bonn, pp. p.3-1~p.3-36.

Fuchs, C. and W. Förstner, 1995. Polymorphic Grouping for Image Segmentation, *The 5th ICCV'95*, Boston, pp. 175-182.

Grün, A., 2000. Semi-automated Approaches to Site Recording and Modeling, *Proceedings of The XIXth Congress of The International Society for Photogrammetry and Remote Sensing*, Amsterdam, Netherlands, Vol. XXXIII, pp. 309-318.

Gülch, E., 1996. Extraction of 3D Objects from Aerial Photographs, Report No. Grant 50TT9536, BMBF/DARA GmbH, Bonn.

Lang, F. and W. Förstner, 1996. 3D-City Modeling with a Digital One-eye Stereo System, *Proceedings of The XVIIIth Congress of The International Society for Photogrammetry and Remote Sensing*, Vienna, Vol. XXXI, pp. 415-420.

Lin, W., Y.-H. Tseng and S. Wang, 2001. On the Model Establishment and Operation for CSG Model-based Building Extraction, *Proceedings of the 20th Conference on Surveying Academy and Applications*, Chungli, Taiwan, Vol. 20.

Lowe, D. G., 1991. Fitting Parameterized Three-Dimensional Models to Images, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5), pp. 441-450.

Sester, M. and W. Förstner, 1989. Object Location Based on Uncertain Models, *Mustererkennung 1989*, Springer Verlag, pp. 457-464.

Tseng, Y.-H. and S. Wang, 2000. CAD-Based Photogrammetry for 3D City Model Reconstruction, *The Congress of The Chinese Geographic Information System Society*, Tainan, Taiwan, Vol. pp. 40-48.

Veldhuis, H., 1998. Performance Analysis of Two Fitting Algorithms for the Measurement of Parameterised Objects, *International Archives of Photogrammetry and Remote Sensing*, Vol. 32, pp. 400-406.

Vosselman, G., 1999. 3D Measurements in Images Using CAD Models, *Proceedings of the 5th Annual Conference of ASCI*, Heijden, Netherlands, pp. 449-456.

Vosselman, G. and H. Veldhuis, 1999. Mapping by Dragging and Fitting of Wire-Frame Models, *Photogrammetric Engineering & Remote Sensing*, 65(7), pp. 769-776.

Wang, S. and Y.-H. Tseng, 2001. Least-squares Model-image Fitting for Model-based Building Extraction, *Proceedings of the Asia GIS 2001*, Tokyo, Japan, pp. 1-8.