MODEL-IMAGE FITTING USING GENETIC ALGORITHMS FOR BUILDING EXTRACTION FROM AERIAL IMAGES

Yi-Hsing TSENG*
Chih-Chiao LIN**
Sendo WANG***

*Department of Surveying Engineering, National Cheng Kung University
#1 University Road, Tainan 701, China Taipei, R.O.C.

**Department of Surveying Engineering, National Cheng Kung University
#1 University Road, Tainan 701, China Taipei, R.O.C.

***Department of Surveying Engineering, National Cheng Kung University
#1 University Road, Tainan 701, China Taipei, R.O.C.

KEY WORDS: Model-based Building Extraction, Model-image Fitting, Genetic Algorithms.

ABSTRACT: Building extraction with model-image fitting has proved a promising approach to acquiring 3D data of buildings from aerial images. The currently most favorable method for model-image fitting is LSMIF (Least-square Model-image Fitting), which is an iterative approach and much relies on given a good approximation. This paper proposes the use of genetic algorithms (GA) for model-image fitting, so as to relieve the dependence on good approximation. In this paper, buildings are reconstructed part by part by fitting each parameterized CSG (Constructive Solid Geometry) primitive to the edge pixels of aerial images. The shape and pose parameters associated to a primitive provide a link between perception (images) and prior knowledge (primitive) of a building part, so that the fitting method solves the parameters for the optimal fitting. To form a whole building, building parts are 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 will present the theory of the GA method for model-image fitting and analyze its performance as well. Both of the fitting methods will be tested on some demonstrating examples and compared based on the test results.

1. INTRODUCTION

Model-based building extraction (MBBE) from aerial images has been known as a convincing approach to acquiring precise, reliable and complete 3D data of buildings (Braun, et al., 1995; Brenner, 1999; Vosselman and Veldhuis, 1999). Model-image fitting plays the central role in MBBE. It requires a computer algorithm that is able to determine the pose and shape parameters of a building model such that the edge lines of the wire frame, as projected onto the images, are optimally coincided with the corresponding edge pixels. In other words, with known image orientation, the objective of model-image fitting is to find the optimal alignment between the model and overlapped images. This procedure is equivalent to the photogrammetric mapping procedure. It delivers the geometric properties of an object determined by the parameterized model. Attempts to solve the problem of model-image fitting date back to the work of 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. This budding research work has marked an important step toward 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 (1991) proposed a least-squares algorithm 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 (Vosselman and Veldhuis, 1999). However, it relies on a good initial approximation of the unknowns to apply LSMIF. This paper proposes the use of genetic algorithms (GA) for model-image fitting, so as to relieve the dependence on a good unknown approximation.

The GA theory was first proposed by Holland (1975). Thereafter, GA was applied in many optimization problems. They are inspired by the mechanics of natural genetics and natural selection where stronger individuals are likely to survive in competing environment. The unknown parameters will be coded as a finite-length string in the binary form and are regarded as the genes of a chromosome. A string, therefore, represents the potential solution of the optimization problem. By generating a set of strings, the GA

algorithm applies three operators, reproduction (selection), crossover and mutation, to evolve better generations. This paper will present the theory of the GA method for model-image fitting and the design of fitness function. With the demonstration of some test examples, its performance will be analyzed in the aspects of approximation requirement and fitting accuracy compared with the LSMIF method.

In this paper, CSG modeling is employed to represent buildings. Buildings are composed of a combination of volumetric primitives. 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. Based on this building representation, the optimal model-image fitting is the key component in the workflow (Figure 1) of the previously proposed MBBE approach (Tseng and Wang, 2001).

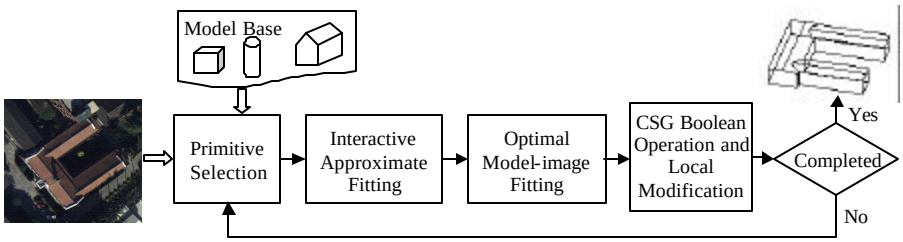

Figure 1: The workflow of the proposed MBBE approach.

2. 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. A model may be described as a polyhedron or a combination of several defined models. Each primitive is associated with a set of parameters that can be categorized into **shape** parameters and **pose** parameters. 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 2. Different primitive models will be associated with different shape parameters. Unlike the shape parameters, pose parameters are not associated to the changes in size or shape, but define the position and orientation of a primitive in the object space. In a three-dimensional space, it is adequate to use 3 translation parameters (dX, dY, dZ) and 3 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 4 pose parameters (dX, dY, dZ, a) for all kinds of building primitives (Vosselman and Veldhuis, 1999).

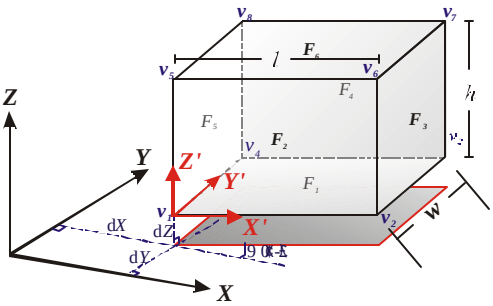

Figure 2: The graphical description of a box primitive and associated shape and pose parameters.

2.2 Fitting Principle and Coordinate Systems

The principle of model-image fitting is to adjust shape and pose parameters so as to fit the boundary lines (after projection) of a model with the corresponding edge pixels extracted from the images. The coordinate systems involved in this approach include **model**, **object**, **photo**, and **image** coordinate systems (Figure 3). Transformations between coordinate systems can be implemented based on the associated parameters indicated in Figure 4. Through the coordinate transformation, the fitting can be performed in the photo coordinate system.

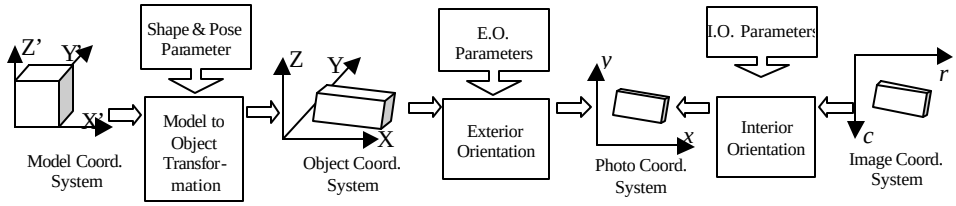

Figure 3: Coordinate systems involved in the fitting algorithm and their relationship.

A primitive is defined in the model coordinate system, and can be transformed into the object space in accordance with the shape and pose parameters to represent a building part. The transformation sequence should be shaping, rotation, and translation. The transformation will only alter the vertex coordinates. However, each vertex is affected by the parameters differently. A building object can be further transformed into a 2D photo coordinate system in accordance with the known exterior orientation through a central projection. This transformation can be implemented by using the collinearity condition equations to project all of the vertices onto the photo plane. Furthermore, extracted edge pixels in the image coordinate system can be transformed into the photo coordinate system in accordance with the interior orientation. The model-image fitting becomes possible through those geometrical transformation processes.

2.3 Least-Squares Model Image Fitting (LSMIF)

The LSMIF method solves the shape and pose parameters so that the sum of the perpendicular distances from the edge pixels to the projected model edge lines is minimized. The summation involves the total numbers of model edge lines (I), overlapped photos (J), and extracted edge pixels (K). Let a model edge line i be projected onto a photo j. The two end points of the projected edge line can be labeled as $v_{ij1}(x_{ij1}, y_{ij1})$ and $v_{ij2}(x_{ij2}, y_{ij2})$. If an extracted edge pixel k from photo j, it is labeled as $T_{jk}(x_{kj}, y_{kj})$. The distance from the edge pixel to the projected model edge line (Figure 5) can be formulated as Equation (1), and the objective function is to minimize the sum of squares of the distance as Equation (2). Equation (3) shows the necessary condition for q to be minimum, which forms the normal equations of the least squares adjustment. Some model edge lines may be excluded from the calculation due to self-occlusion, which can be found automatically through the calculation of projection. By providing approximate values of the shape and pose parameters, it would also be reasonable that only the edge pixels distributed within a buffer zone (Figure 4) of the projected edge lines are used for calculation. The use of buffer enables us to screen out irrelevant pixels.

$$d_{ijk} = \frac{\left|(y_{ij1} - y_{ij2})x_{jk} + (x_{ij2} - x_{ij1})y_{jk} + (y_{ij2}x_{ij1} - y_{ij1}x_{ij2})\right|}{\sqrt{(x_{ij1} - x_{ij2})^2 + (y_{ij1} - y_{ij2})^2}} \quad (1)$$

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

$$\frac{\partial q}{\partial p_l} = 0 \text{; for all unknown parameters } p_l. \quad (3)$$

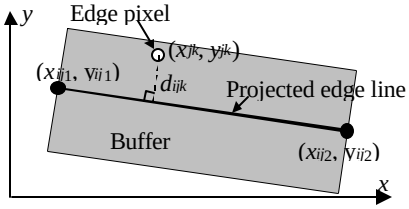

Figure 4: Distance from an edge pixel to the projected edge line and the buffer zone.

In practice, we take derivative of Equation (1) at each unknown parameters, and form the normal equations using matrix operations. Equation (1) is a non-linear function with respect to the unknown parameters. For the box primitive, Equation (2) can be rewritten as $d_{ijk} = F_{ijk}(w, l, h, \alpha, dX, dY, dZ)$. The typical solution of a non-linear least squares adjustment is to apply the Newton's method. Using the first order Tyler expansion, the non-linear equation can be linearized as the function of the increments of parameters. Given a set of initial approximations, the unknown parameters are updated iteratively by the calculated increments. The linearized form can be expressed as:

$$d_{ijk} = \left(\frac{\partial F_{ijk}}{\partial w}\right)_0 \Delta w + \left(\frac{\partial F_{ijk}}{\partial l}\right)_0 \Delta l + \left(\frac{\partial F_{ijk}}{\partial h}\right)_0 \Delta h + \left(\frac{\partial F_{ijk}}{\partial a}\right)_0 \Delta a + \left(\frac{\partial F_{ijk}}{\partial dX}\right)_0 \Delta dX + \left(\frac{\partial F_{ijk}}{\partial dY}\right)_0 \Delta dY + \left(\frac{\partial F_{ijk}}{\partial dZ}\right)_0 \Delta dZ + F_{ijk0} \quad (4)$$

In Equation (4), F_{ijk0} is the approximation of the function F_{ijk}, and (w, l, h ...) are the increments of unknown parameters. The linearized equations can be expressed as a matrix form: $\boldsymbol{V}=\boldsymbol{AX}-\boldsymbol{L}$, where $\boldsymbol{A}$ is the matrix of partial derivatives; $\boldsymbol{X}$ is the vector of the increments; $\boldsymbol{L}$ is the vector of approximations; and $\boldsymbol{V}$ is the vector of residuals. The objective function actually can be expressed as $q=\boldsymbol{V}^T\boldsymbol{V}$. For each iteration, $\boldsymbol{X}$ can be solved by the matrix operation: $\boldsymbol{X}=(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{L}$. The iteration normally will converge to the correct

answer. However, inadequate relevant image features, affected by irrelevant features or noise, or given bad initial approximations may lead the computation to a wrong answer.

3. MODEL-IMAGE FITTING USING GENETIC ALGORITHMS

The first step to apply a GA is to set up the functions of the following components: the chromosome representation and genetic operators, fitness function, as well as the algorithm design including the creation of the initial population, termination rule, population size, crossover rate (P_c), and mutation rate (P_m), etc. The following subsections explain how those components are designed for model-image fitting.

3.1 Chromosome Representation and Genetic Operators

An optimization problem conventionally is modeled as a mathematic function of a set of parameters. In a GA, the parameters are coded as a finite-length binary string imitating a chromosome. Therefore, a chromosome represents a possible solution of the problem. A fitness value can be computed with the mathematic function to evaluate how good the solution provided by a chromosome, so that the selection of superior chromosomes can be performed based on the comparison of fitness values. A GA begins with the creation of a randomly generated initial population, and then the population evolves according to the genetic operators, such as reproduction (selection), crossover and mutation. The evolution of is repeated until a desired termination rule is reached.

Reproduction is a selection process in which individual strings are kept to produce the next generation in accordance with their fitness values. In this paper, we applied the commonly knownselection mechanism developed by Holland (1975) for the reproduction process, in which the probability of selection, P_j, for each individual j is defined by:

$$p_j = f_j / \sum f_j, \quad j = 1, 2, \ldots, n \tag{5}$$

The crossover process takes a pair of chromosomes and yields two offsprings according to an operation rate, and the mutation process alters one chromosome to produce a single new solution. The mutation process should only happen in a small probability. Figure 6 illustrates the concept of two commonly used operators: the uniform-crossover (Figure 5.a) and the bit mutation (figure 5.b). The offspring are generated form the parents, based on a randomly generated crossover mask, so that the offspring contain a mixture of genes from the parents.

The GA procedure can be summarized as the following:

(1) Starts, generation $t = 0$.
(2) Generate a population P_0 of N individuals and respective fitness values, generation $t = t + 1$.
(3) Select the chromosomes in the population P_{t-1} for crossover and mutation operations.
(4) Evaluate P_t.
(5) Repeat step 4 to 6 until termination.
(6) Ends the procedure.

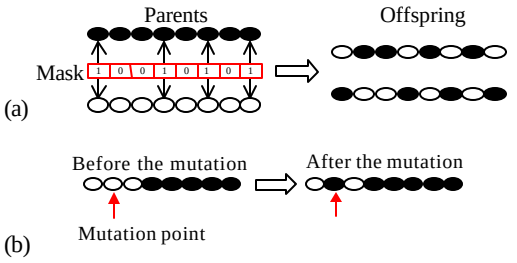

Figure 5. Diagrams of (a) the uniform-crossover and (b) the bit mutation mechanism.

3.2 Fitness Functions

The design of the fitness function is essential for a GA to search for the optimal solution. The intuitive fitness function is the least-squares solution, i.e., minimizing the function in Equation (2). However, this calculation is based on an assumption that edge pixels screened using a buffer of a model edge line should correspond with the real edge line of the object. This assumption is valid only when the parameters are approximately known, so that it works for the LSMIF solution. It is not the case when a GA is applied. A set of randomly generated values of the parameters could be quite different from the correct numbers. Under this circumstance, how many correctly correspondent edge pixels captured by the buffer becomes an important factor. When a chromosome represents a set of parameter values far different from the correct numbers (i.e., few correspondent pixels are found), it may lead to two extreme cases. The first case is that a large number of irrelevantpixels are caught, and the calculation of Equation (2) would result in a large value indicating a bad fitting. However, the other case is that very few pixels are caught, and the calculation results in a small value indicating a good fitting. The former case makes a correct indication, but the later case will mislead the solution. Therefore, an elaborate fitness function is requited.

We propose a fitness function, which takes the number of edge pixels in the buffer, the length of projected edge line, as well as the sum of the perpendicular distances from the edge pixels to the projected edge line into account. First, for each visible model edge line, a fitness fraction s_i is calculated by Equation

(6) as a contribution to the fitness function. This follows the basic idea of least squares fitting. Then, each s_i will be modified based on the number of edge pixels n_i found inside the correspondent buffer. The modification will be made based on a consideration of the number of pixels expected and the maximum contribution of an s_i, n_i^{Max} and $s_{i_{\max}}$, which are functions of the length of the projected edge line, L_i. For example, n_i^{Max} can be L_i divided by pixel size and $s_{i_{\max}} = L_i / \sum L_i$. Equation (7) shows the modification proposed. Therefore, the final fitness value is the summation of the modified fitness fraction s_i^* of each visible model edge line in images. Equation (8) shows the fitness function of the model parameters represented by vector x, in which N is the total number of visible edge lines in all of the involved images.

$$s_i = 1 - \left(\frac{\sum_{j=1}^{n_i} d_{ij}^2}{n_i} \Big/ buffer^2 \right) \qquad (5)$$

$$\begin{cases} s_i^* = s_i \times \left(n_i / n_i^{Max} \right) \times s_{i_{\max}}, & 3 < n < n_i^{Max} \\ s_i^* = s_i \times s_{i_{\max}}, & n \geq n_i^{Max} \\ s_i^* = 0, & n \leq 3 \end{cases} \qquad (6)$$

$$F(x) = \sum_{l=1}^{N} s_l^* \qquad (7)$$

The range of fitness value of equation (7) is within [0,1]. The GA is, therefore, designed to search for the maximum value of the fitness function, which corresponds to find the optimal solution of the parameters.

3.3 Algorithm Design

- **Searching Ranges of the Model Parameters**

A GA is usually designed to seach for the optimal solution within a seaching range of each parameter. Approximate values are needed to define the searching ranges. If wide searching ranges are used, the approximations can be very rough. On the contrary, good approximations are required. However, the wider searching range, the larger ammount of calculation needed and the higher properbility to find a wrong solution. In general, a GA does not rely on good approximations as LSMIF does. In this study, 4 cases of searching range assigment were tried for each test: ± 3 (case ***a***) ± 5 (case ***b***) ± 8 (case ***c***) ± 10 (case ***d***). The unit of the seaching will be meter for length and degree for angle.

- **Genetic Operators, and Genetic Parameters.**

In this study, the searching range of each parameter is coded into an 8-bit binary string. A chromosome is then composed of all binary strings of the parameters. For example, a box model has 7 parameters (w, l, h, $á$, dX, dY, dZ), which can be coded into a binary string. The precision of parameter determination will be the function of the searching range and the string length. For instance, the precision of the case ***d*** will be 0.078 m for length and 0.078 deg for angle. Roulette wheel selection is used to choose individuals from the population generated by crossover. Elitism mechanism (ref) is used for the generational replacement, which ensures the preservation of the fittest subpopulation for the next generation. In our tests, 10 percent of the best individuals were preserved for the next generation. To keep the amount of variation for the explored solution and consider the convergence speed of fitness value, the uniform-crossover is applied. The termination rule is simply a setting of a specified maximum number of generations. Based on the tests, the maximum generation number can be set as inbetween 200 and 1000 in consideration of applied searching range. For the setting of the crossover rate (P_c), and mutation rate (P_m), we adopt the suggestion proposed by Goldberg (1989), i.e., P_c and P_m are in the ranges of [0.6,1.0] and [0.001,0.1] respectively.

4. EXPERIMENTS

Several model-image fitting tests, for the box and gable-roof primitives, have been studied. The test aerial photos have the image scale of 1 to 5000, and were scanned into digital images with 25 m resolution (12.5 cm on the ground). The exterior orientation parameters of the photos were determined by aerial triangulation. In this section, we demonstrate two examples first, for a box and gable-roof primitives respectively, in which the performance of calculation convergence is analyzed. Then, how the occlusion problem affects the solution is investigated. Finally, the fitting accuracy of the results from the applications of GA and LSMIF will be compared.

4.1 Examples of Model-image Fitting Using GA

The first example is fitting a box primitive to a stereo image pair. For the GA parameter setting, the population size is 100, P_c =0.8, and P_m =3/56. Within 500 generations, using any case of the test searching ranges can converge to the correct answer. For the searching range case ***d***, Figure 6.a and 6.b show the 5 best solutions of the first and 500th generation respectively. Figure 6.c shows the

curves of the average and the best fitness values with respect to the generation number. Figure 6.d shows the fitting result and the 4 check points. The second example is fitting a gable-roof primitive to a stereo image pair. For the GA parameter setting, the population size is 100, P_c =0.8, and P_m =5/56. Perhaps, because the additional parameter (*rh*) is highly correlated with the *dZ* and *h* parameters and the roof tile of the building presents a lot of ignoring edge pixels, the narrowest searching range (case ***a***) is required to obtain correct solution. It means that it needs a good approximation for the solution. Nevertheless, this approximation requirement is still not so strict as required in using LSMIF, which usually has a pull-in range of ± 2 m for each parameter. Figure 7 shows the fitting result and the 4 check points.

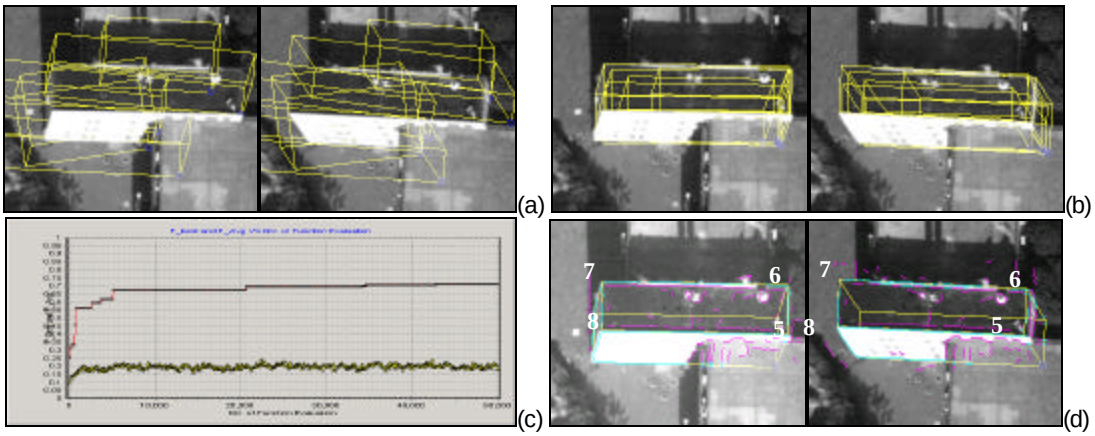

Figure 6: An example of fitting a box primitive to a stereo pair. (a) The 5 best solutions of the first generation; (b) The 5 best solutions of the 500th generation; (c) The curves of the average and the best fitness values with respect to the generation number; (d) The fitting results from the calculation of 500 generations.

4.2 Occlusion Example

Building edges are frequently occluded in an image, and some of them may be critical in determine a certain parameter. Consequently, model-image fitting will be failed. One way to fix this problem is to predetermine those weak parameters using other observations. For example, the bottom edges of a building are frequently invisible in an aerial image, so that the *dZ* parameter cannot be well determined in the solution. Under this circumstance, this parameter should be predetermined, so that we can set a very narrow searching range for the parameter in the GA depending on how accurate the parameter is observed. This is similar to set a weight constraint on a certain parameter in LSMIF (Tseng and Wang, 2001). Figure 8 shows an example that needs 2 constraints for fitting. The first constraint is given for the *dZ* parameter due to the bottom edges of the building are occluded, and the second constraint is given for the *w* parameter to prevent the fitting to the shadow edge indicated in Figure 8.

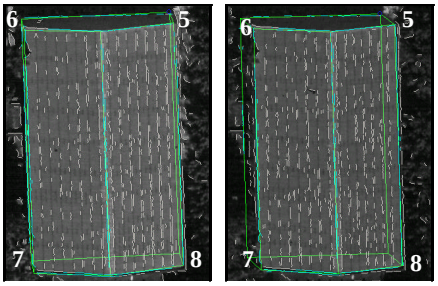

Figure 7: The example of fitting a gable-roof primitive.

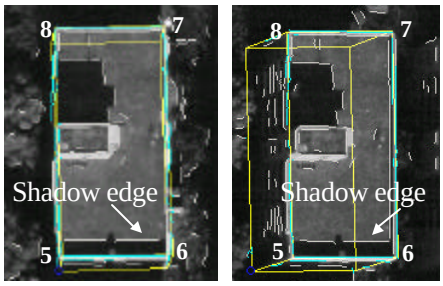

Figure 8: An example of fitting needs constraints.

4.3 Accuracy Assessment

To assess the fitting accuracy, the derived 3D coordinates of the building corners from the fitting results are compared with manually measured data. From each test, the 4 topside corners of the buildings with index number 5~8 (as shown in Figures 8.d, 9 and 10) are compared. Since the GA is not a least-squares solution, the fitting results are different from the LSMIF results. To show the differences, we also list the data of LSMIF accuracy assessment together. Table 1 shows the average and RMS differences of each coordinate. In general, the horizontal accuracy is about 0.5m, and the height accuracy is about 1m. The height accuracy is lower than the horizontal accuracy due to the small base-height ratio (about 0.3) of the

images.

Table1: Accuracy assessment by comparing the derived building-corner coordinates from the GA and LSMIF results with manual measurements.

Coordinate	ΔX (m)		ΔY (m)		ΔZ (m)	
Method	GA	LSMIF	GA	LSMIF	GA	LSMIF
Average Diff.	0.075	0.226	-0.055	-0.084	-0.724	-0.334
RMS Diff.	0.417	0.360	0.371	0.309	0.730	0.583

5. CONCLUSIONS

A GA is applied for model-image fitting to reconstruct buildings from aerial images. The design of the fitness function is the key to success. The proposed fitness function has been validated by several sets of test data. The GA method has proved itself capable of solving the model-image fitting problem, and achieves the fitting accuracy as good as LSMIF. However, the GA solution is not quite stable. That the number of generations needed for an evolution to reach the optimal solution is also case by case. Furthermore, the GA needs much more computation time than LSMIF. Further study is required to improve the GA for a practical application.

ACKNOWLEDGEMENTS

This research project was sponsored by the National Science Council of Republic of China under the grants of NSC 90-2211-E-006-103 and NSC 91-2211-E-006-092.

REFERENCES

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

Brenner, C., 1999. Interactive Modeling Tools for 3D Building Reconstruction, Photogrammetric Week '99, Stuttgart, 47, pp. 23-34.

Fua, P., 1996. Model-based Optimization: Accurate and Consistent Site Modeling , *International Archives of Photogrammetry and Remote Sensing,* Vol. 31, Part B3, 222-233.

Förstner, W., 1994. A Framework for Low Level Feature Extraction, *Computer Vision -ECCV '94*, Vol. 2, Springer-Verlag, 383-394.

Goldberg, D. E., 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Pub. Com. Inc., USA.

Hollen, J.H., 1975. *Adaptation in Natural and Artificial Systems*, Ann Arbor: The University of Michigan.

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

Sester, M. and Förstner, W. , 1989. Object Location Based on Uncertain Models, *Mustererkennung 1989*, Informatik Fachberichte 219, Springer-Verlag, 457-464.

Tseng, Y.H. and Wang, S., 2001. Experiments on CSG Model-based Building Extraction from Aerial Images, *Proceedings of the 22nd Asia Conference on Remote Sensing*, pp. 1191-1196.

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