BUILDING RECONSTRUCTION BASED ON MDL PRINCIPLE FROM 3-D FEATURE POINTS

Y. Ishikawa[a], I. Miyagawa[a], K. Wakabayashi[a], T. Arikawa[a]

[a] NTT Cyber Space Laboratory, NTT Corporation, 1-1 Hikarinooka Yokosuka-Shi Kanagawa, 239-0847, Japan
(ishikawa.yuji, miyagawa.isao, wakabayashi.kaoru, arikawa.tomohiko)@lab.ntt.co.jp

Commission III, WG III/7

KEY WORDS: Reconstruction, Urban, Three-dimensional, Method, Building

ABSTRACT:

For 3-D building reconstructions of urban areas, we present a fully automatic shape recovery method that uses 3-D points acquired from aerial image sequences. This paper focuses on shape recovery of flat rooftops that are parallel to the ground. We recover each rooftop from a set of 3-D points located at nearly the same height. Such 3-D point sets are made by merging point sets under the MDL (Minimum Description Length) principle, which finds suitably concise point sets for 3-D building models. Often, only parts of rooftop shapes can be recovered because of the 3-D position errors being generated in the points. To refine the recovered shapes, we merge the parts under a heuristic condition in which shapes will have a pair of orthogonally oriented edges. To optimize parameters and estimate the viability of our method, we defined a success rate, called the cover ratio, as the area in which the recovered shape and a correct shape (given as reference data) overlap to the combined area of the recovered and correct shapes. Experimental results showed that our method achieved a cover ratio of 75.25%, and through improved cover ratio we also confirmed effectiveness of shape refinement. We also found that even if only one-ninth of the reference data could be used in the optimization of parameters, the cover ratio was 70.96%. The experimental results we obtained showed that our point-based method was effective in enabling the recovery of buildings in urban areas.

1. INTRODUCTION

Three-dimensional building models in urban area, i.e. 3-D city models, are used widely to numerically simulate events such as floods, fire spreading, wave propagation and heat convection. Computer graphic models for car navigation have also been rapidly shifting from 2-D to 3-D systems. Accordingly, 3-D city model reconstruction technologies are becoming increasingly in demand.

There are mainly two ways to categorize recovery methods: measurement (aerial or ground) systems and sensing devices (camera or laser profilers). The airborne approach is suitable for reconstruction over a wide area. Especially over the past few decades, a considerable number of studies have been made on recovery methods in which line segments were extracted from aerial images and 3-D information was obtained by stereoscopic vision (Herman, 1986; Henricsson, 1998; Paparoditis, 1998). Not limited to the recovery of 3-D building shapes, line segments have been one of the important primitives in shape reasoning. However, two kinds of line segments, i.e. *edge* and *pattern,* frequently mingle together in images. While edge lines are very useful for contour detection, line patterns on flat surfaces have nothing to do with the contours. This makes it difficult to recover shapes on the basis of line segment connections.

On the other hand, since points are more primitive than line segments, they have several advantages over line segments. For one thing, they make it easer to acquire 3-D positions and to recover 3-D shapes. For example, 3-D shapes can be recovered simply by generating a triangle mesh over 3-D points. Actually, triangle meshes are applied widely in recovering curved surfaces (Terzopoulos, 1991). Application users, however, require a 3-D city model that is not a minute mesh model but a CAD model, i.e., one which is a set of concise polyhedrons.

Recently, it has become easier to acquire high-resolution, high-frame-rate image sequences, and therefore it is becoming feasible to recover building shapes with *Shape from Motion* (Miyagawa, 2000). In this paper, we present a fully automatic method of recovering shapes from 3-D points acquired from aerial image sequences to enable 3-D reconstruction of buildings in urban areas. The paper focuses on shape recovery of flat rooftops being parallel with the ground. Thus we recover each rooftop from a set of 3-D points located at nearly the same height. Such 3-D point sets are made by merging point sets under the MDL (Minimum Description Length) principle (Rissanen, 1984), which finds suitably concise point sets for 3-D building models. Often, only parts of rooftop shapes can be recovered because of 3-D position errors in the points. To refine the recovered shapes we merge the parts under a heuristic condition under which the shapes will have a pair of orthogonally oriented edges after mergence.

In Section 2 we show a recovery method based on MDL from 3-D points. In Section 3 we explain refinement of recovered rooftop shapes considering 3-D position errors and recovered shapes. We show experimental results in Section 4. Finally, we summarize the main points of this paper in Section 5.

2. RECOVERY METHOD FROM 3-D POINT SET

Figure 1 depicts an outline of our recovery method. Here, the figure shows the (sub)sections where each step in the recovery process is explained.

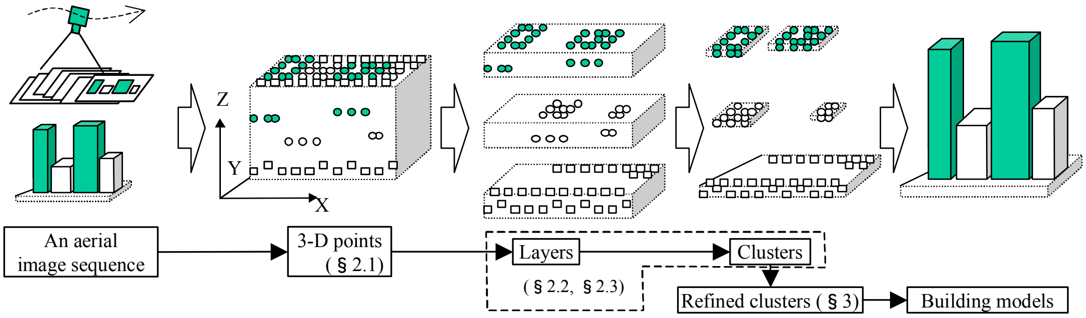

Figure 1. Outline of building recovery method

2.1 Acquisition of 3-D Point Set

A 3-D point set is acquired automatically as follows: detect feature points, track them, acquire 3-D positions, remove outliers, and transform the camera coordinate system into a ground coordinate system (X,Y,Z) based on information of camera position and orientation. For the robust acquisition of 3-D positions we adopt an iterative perspective factorization method (Christy, 1996). In the removal step, 3-D points are projected into all images. For each point we then measure the difference between 2-D positions found by the tracking and those obtained by the projection for all images. Since these differences reflect errors in the tracking process, points which differ significantly are removed from the 3-D point set. Through this process, only highly precise points remain the 3-D point set.

2.2 Layer and Cluster Generation

In this subsection we show our method of recovering shapes from the 3-D point set. Establishing a Z coordinate for each point *height*, the method starts by slicing the point set into *layers*, which are sets of points at almost the same height. After generating the layers, each layer is divided into *clusters*, which are sets of points gathered together on an X-Y plane. A 2-D convex polygon covering each cluster is then formed as a rooftop 2-D shape. Our method recovers each rooftop as a flat roof building, the height of which is the average of the heights of points in each cluster. The recovery process is completed by adding vertical walls to the rooftops.

In contrast to the results obtained using a laser profiler, with our method horizontal distribution of the 3-D points acquired from images is not uniform. This makes it difficult to generate layers and clusters based on a height histogram (Ledur, 1998). The procedures our method follows are given below.

Layer generation. Each initial layer is an individual 3-D point. Let initial layers be $\{L_1, ..., L_n\}$ in order of the height of the points. For each i (i=1,...,n-1), the standard deviation σ_i of the heights of points in L_i and L_{i+1} is calculated. Minimal standard deviation σ_{min} is found from n-1 σ_i. Then L_{min} merges with L_{min+1} and $n = n$-1. For each merge step MDL (defined in Subsection 2.3) is computed. Finally $\{L_1, ..., L_k\}$ whose MDL is minimal is output as the optimal layer set.

Cluster generation. The following process is executed for each layer L_i. Each point in L_i becomes an initial cluster. First of all, distances for all point pairs (p,q) are calculated. Then, in order of the distance lengths of the pairs, the closest two clusters (including the respective points p and q) are checked and are merged if they pass the check. The check begins with creating the smallest convex polygon including all points of the two clusters on the X-Y plane. If the polygon also includes a point belonging to a layer lower than L_i, the two clusters are not merged. If there are no such lower points, the two clusters are merged. This check enables us to form rooftop regions that are consistent with the upper and lower relations of layers.

2.3 MDL-based Determination of the Number of Layers

To prevent layers from being divided too much in the layer generation process, our proposed method determines the number of layers based on the MDL principle as follows:

Suppose that the 3-D point set is divided into k layers $\{L_1, \cdots, L_k\}$ and $m(i)$ clusters $\{C_{i,1}, \cdots, C_{i,m(i)}\}$ are generated from each layer L_i (i=1,...,k). As a result, the 3-D point set is divided into sets of clusters $\{\{C_{i,1}, \cdots, C_{i,m(i)}\} \mid i=1,...,k\}$. This means that the 3-D point set is expressed by rooftop models recovered from the cluster sets. Given the point set and the parameter k, the cluster set $\{\{C_{i,1}, \cdots, C_{i,m(i)}\} \mid i=1, \cdots, k\}$ can be determined. For this reason we regard k as the degree of freedom of the rooftop models. k should be determined by balancing the simplicity of the models with the conformity of the points to the models.

The MDL principle (Rissanen, 1984) is used in many research areas such as pattern recognition (Leclerc, 1990). Given a model and data, suppose that code length M is the sum of the two code lengths M_m and M_d required for description of the model and the data. Based on the MDL principle, the model that minimizes M is the optimal model for describing data. For our problem, we define M as Equation (1):

$$\begin{aligned} M &= M_m + M_d \\ &= \ln {}_{n-1}C_{k-1} + \frac{m}{2} \ln n \\ &\quad + \frac{1}{2\sigma^2} \sum_{i=1}^{k} \sum_{j=1}^{m(i)} \sum_{p \in C_{i,j}} \left(z(p) - z(C_{i,j}) \right)^2 \qquad (1) \end{aligned}$$

where n is the number of 3-D points, m is the number of clusters, σ is the standard deviation of the height of points in one cluster (we assume σ is constant over all clusters), Z(p) is the height of a point p, Z(C) is the average of the heights of points in a cluster C. The derivation of Equation (1) is shown in Appendix A. We employ the parameter k that minimizes M as the number of layers.

3. CLUSTER REFINEMENT BASED ON CLUSTER SHAPE AND POSITION ERROR

3.1 Relaxation of *Exclusive Constraint*

As mentioned in Subsection 2.2, clusters are merged under a constraint such that a convex polygon of any cluster in a layer L may not cover any point included by any layer lower than L. In this paper we call the constraint the *exclusive constraint*. As pointed out in Subsection 2.1, points that have large projection errors are removed from the 3-D point set. After the removal, the 3-D position of the point set still contains some errors. Unless the exclusive constraint is relaxed, therefore, clusters are insufficiently merged and only fragments of rooftop shapes are obtained.

The optimization method *Regularization* is adopted in previous works in which the 3-D surface is recovered from 3-D points (e.g. Grimson, 1983). Suppose that D is a gap estimate function between 3-D points and a surface and S is a shape estimate function of the surface. Under these assumptions, the optimal surface is found by minimizing $E = D + \lambda S$, where λ is a parameter. Minimization of E is often solved using numerical analysis methods for differential equations or using search methods, especially Dynamic Programming. For our method, however, clusters cannot be expressed in numerical form and cannot be independent of the generation process. Therefore neither the use of numerical analysis nor that of search methods is effective for our method.

We do not apply the Regularization framework rigidly to our method. Based on estimate functions similar to D and S, we decide whether or not to relax the exclusive constraint and merge clusters. Parameters in the estimate functions are determined by optimization for reference data. To achieve the optimization, it is important to reduce the number of parameters in the estimate functions. We do not employ the parameter λ and we hold the number of parameters to a minimum, that is, we introduce only one parameter into the D and S functions.

3.2 Penalty *D* for Breaking Constraint

It is assumed that on the X-Y plane a convex polygon Pc includes all points in a cluster C in a layer L and a point q_i in a layer lower than L. As a 3-D model the cluster C expresses a prism whose shape on the X-Y plane is Pc and whose height is Z(C), and then q_i is located inside the prism (see Figure 2).

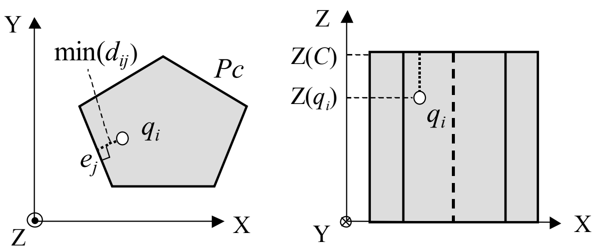

Figure 2. A point q_i breaking *exclusive constraint*

For each edge e_j of the Pc, horizontal distance d_{ij} to q_i is computed, and min(d_{ij}) is the minimal value of d_{ij}. Then the *depth* of q_i for C is defined as smaller than horizontal *depth* min(d_{ij}) or vertical *depth* Z(C)-Z(q_i) (see Equation (2)).

$$d_i = \min\left(Z(C) - Z(q_i),\ \min_{e_j \in Pc}(d_{ij}) \right) \tag{2}$$

Using d_i, Equation (3) defines probability $P(C, q_i)$ that the prism expressed by C represents as an actual building, where L is the peripheral length of Pc and α_1 is a parameter.

$$P(C, q_i) = \exp(-\frac{d_i}{\alpha_1 L}) \tag{3}$$

Suppose that Pc includes n points $\{q_1, ..., q_n\}$. Assuming that $P(C, q_i)$'s are independent of each other for all q_i, probability $P(C)$ of the rooftop shape being represented by Pc is the following:

$$P(C) = \prod_i P(C, q_i) = \exp\left(-\frac{1}{\alpha_1 L} \sum_i d_i \right) \tag{4}$$

Letting α_2 be a threshold of $P(C)$, we can transform a condition $P(C) \geq \alpha_2$ into $\sum d_j / L \leq \alpha_1 \log(\alpha_2)$. For penalty that C includes the points $\{q_1, ..., q_n\}$ in the lower layer, an estimate function $D(C)$ is defined as Equation (5),

$$D(C) = \frac{1}{L} \sum_j d_i \tag{5}$$

Finally, the exclusive constraint is relaxed by allowing $D(C)$ to reach a threshold $\alpha (= \alpha_1 \log(\alpha_2))$.

3.3 Shape Estimation Function *S*

Most rooftop shapes in urban areas have right angles. This feature is often exploited for extracting rooftop shapes. Thus for cluster C we design the shape estimation function $S(C)$ so that $S(C)$ takes a larger value as an angle θ_{ij} of an edge pair (e_i, e_j) in Pc becomes closer to a right angle. Note that (e_i, e_j) is taken from not only a pair of adjacent edges but also every edge pair in Pc. Even though Pc does not have a right angle, the positioning of the edge pair is evaluated highly in $S(C)$ if Pc has orthogonally oriented edge pairs.

$S(C)$ is defined as Equation (6), which is an average of an exponential function with θ_{ij} weighted by $l_i l_j$, where l_i is the length of e_i and β is a parameter.

$$S(C) = \frac{\sum_{(i,j), i \neq j} l_i l_j \exp(-\left(\frac{\theta_{ij} - \pi/2}{\beta} \right)^2)}{\sum_{(i,j), i \neq j} l_i l_j} \tag{6}$$

From Equation (6), if Pc has orthogonally oriented pairs of long edges, $S(C)$ becomes high, and if Pc is an excessively slender rectangle, $S(C)$ becomes low. Thus we consider $S(C)$ as an effective function in selecting shapes for rooftops.

3.4 Constraint Relaxation Based on Cluster Shapes

We relax the exclusive constraint based on $S(C)$. It is better for the cluster shape to be estimated at final step in the generation process, but there are too many combinations of clusters for mergence to enumerate all clusters at the final step. Therefore, clusters are initially generated with a *rigid* exclusive constraint, i.e., without relaxation. After this step only fragments of a rooftop shape can be obtained. In this step every cluster pair (C_i, C_j) for which C_i and C_j have cancelled mergence by the exclusive constraint is listed in the order in which the cancellation happens. Secondly, suppose that C is a cluster including all points of C_i and C_j on the X-Y plane. Then $S(C)$ is evaluated and it is decided whether or not we should relax the exclusive constraint and merge the two clusters. These evaluations and decisions are conducted in the order in which they appear on the cancellation list. We call the second process *cluster refinement*. The mergence is accepted if Condition (7) is satisfied, where max() expresses a function which finds a maximum:

$$D(C_{ij}) \le \alpha \ \textit{and} \ \max(S(C_i), S(C_j)) \le S(C_{ij}) \qquad (7)$$

Using this condition, if a mergence brings a rooftop shape including orthogonally oriented long edge pairs, the exclusive constraint is relaxed and the mergence is executed.

4. EXPRERIMENTAL RESULTS

4.1 Definition of Recovery Ratio

In our method parameters σ, α and β are used in the calculation of MDL and in judging whether to relax the exclusive constraint. We determine the parameters by maximizing the reference data recovery ratio. In this case the recovery ratio definition is inseparably connected to the determination of the parameters. Thus we here explain the definition of recovery ratio that we have used in our experiments.

Recovery ratio is often defined based on the number of recovered buildings. Judgment whether each building is recovered or not, however, cannot avoid being subjective to some degree. In particular, judging shapes recovered by our method is often difficult, because polygons of the point-based method are more likely to have many vertices than in the line-segment based method. Therefore we evaluate the recovered shapes objectively and quantitatively by comparing them with reference data.

For each target object on the ground a rooftop polygon P is obtained manually on images as reference data. On the other hand, a rooftop polygon Pc is supposed to be recovered from a cluster C that expresses the rooftop of the object. *Cover ratio* $E(P;Pc)$ of P for Pc is defined as $100 * S_{P\cap C} / (S_P + S_C - S_{P\cap C})$, where S_P, S_C and $S_{P\cap C}$ are respectively the areas of P, Pc, and the overlap of P and Pc. If two or more Pc lap over one reference polygon P, let maximal $E(P;Pc)$ be cover ratio $E(P)$ for P. $E(P)$ can be computed automatically with this rule.

Reconstructed 3-D city models are often applied for visualization or numerical simulation. In many cases shape errors of larger buildings have more negative influence on the applications. For this reason we employ an area weighted average of $E(P_i)$ as *cover ratio* $E(\{P_i\})$ for a set of reference polygons $\{P_i\}$. Assuming that S_{Pi} is the area of P_i, $E(\{P_i\})$ is given by $\Sigma S_{Pi} E(P_i) / \Sigma S_{Pi}$. In the following section, we regard the cover ratio as the recovery ratio.

4.2 3-D Point Set and Reference Data

4.2.1 Acquisition of 3-D Point Set: Each 3-D point set was acquired from an aerial image sequence that consists of 80 frames (1290×1080 pixels). In selecting nine urban area scenes, feature points were detected in the center area of the middle frame in each sequence. The center area was 1320×360 pixels, which was roughly equivalent to 220×60 meters on the ground. The feature points were tracked over all 80 frames by the Lucas-Kanade technique (Bradski, 2000). 3-D points were projected into all 80 frames, and then points that had a projection error of more than three pixels were removed as outlier. In the end we obtain 4756.1 points on average per sequence. Figure 3a is a frame example for the feature point detection, in which the white-lined rectangle expresses the detection area. We also selected eight other dense urban area scenes as shown in Fig. 3a. Figure 3b shows a 3-D point set acquired from a sequence for which the middle frame is Fig. 3a. In Fig. 3b the more black there is, the higher the point is.

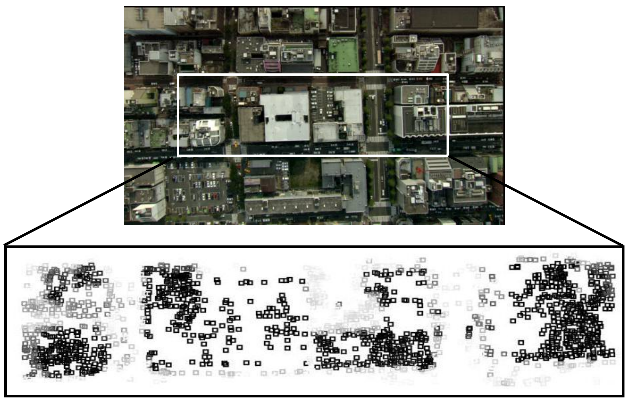

Figure 3: An example of an image frame (3a:upper) and 3-D points (3b:lower)

4.2.2 Collection of Reference Data: From the middle frames of each of the nine sequences we manually obtained reference data, which was a 2-D rooftop shape of man-made objects on the ground satisfying the following conditions: a) every shape was a convex polygon, b) area was over 1000 pixels (≒28 m^2), c) height was over 10 meters. In the end we obtained 96 polygons as reference data. There were 10.7 polygons on average per scene, and the number of polygons in one scene is 18 at the maximum and six at the minimum.

4.3 Evaluation Results of Recovery Method

We evaluated our method in both the cases where the exclusive constraint was relaxed and where it was not. We call the former case the *Relax* method and the latter case the *Rigid* method. To optimize the parameters in the estimate functions we utilized the Powell method, which is one type of direction set method (Press, 1992).

4.3.1 Optimizing With All Reference Data: The Rigid method uses only one parameter σ contained in Equation (1). On the other hand, the Relax method has two additional parameters α and β. We optimized the parameters with all reference data and generated clusters using both the Rigid and Relax methods. The cover ratio with the clusters are shown in Table 4,

	σ [m]	α	β	*E* [%]
Rigid	2.96	—	—	68.18
Relax	1.80	0.106	0.285	75.25
Difference	-1.16	—	—	7.07

Table 4. Optimization and estimation results.

The Rigid and Relax methods achieved cover ratios of 68.18% and 75.25% respectively. Considering that the recovery was executed using only information about the 3-D point position, these results show the effectiveness of the 3-D point-based method. In particular, the Relax method raised the cover ratio to more than 7% that of the Rigid method, which confirms the effectiveness of the cluster refinement described in Subsection 3.4. We consider values of σ in Table 4 as estimated values of the standard deviation of the height of points included in one cluster. Considering that the height of Japanese buildings per floor is about 4m according to statistics, the values seem valid. And this result shows that layer division based on the MDL principle worked well.

Layers and clusters are generated so that M in Equation (1) is minimal. If σ is small, M_d has more influence on M than M_m does. This means that height error $Z(p)$-$Z(C)$ is regarded as more important than the number of clusters m. Considering that σ of the Relax method is smaller than that of the Rigid method, the Relax method acquired fragments of accurate shapes before relaxation. Then, by relaxing the exclusive constraint, the fragments were merged. Finally, more accurate shapes were recovered with the Relax method than with the Rigid method. Figure 5 shows the difference between the cover ratios achieved by the Rigid and Relax methods for all 96 reference data items, where the horizontal axis is the cover ratio of the Rigid method and the vertical axis is that of the Relax method.

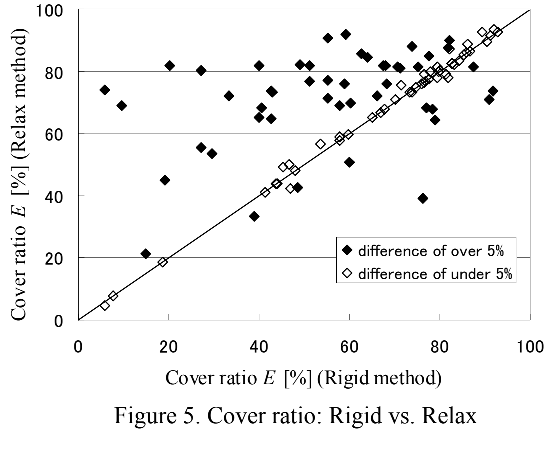

Figure 5. Cover ratio: Rigid vs. Relax

In Figure 5, the points on the diagonal line show cases where there was no difference between the two methods, while the points in the upper-left area show cases where the Relax method achieved better results. The difference was less than 5% in exactly half of the cases (48 cases), while in many other cases the cover ratio was highly improved. In other words, the Relax method tended to maintain the cover ratio of the Rigid method or to greatly improve it in cases where improvement is possible. Condition (7) means two clusters are merged only if shape estimation is definitely improved. Thus, it can be said that the Relax method tends to relax the exclusive constraint in a conservative manner. And this tendency can be seen in evaluating it.

4.3.2 Optimizing With Partial Reference Data: In actual applications, only a small reference data set can be used to optimize parameters. Therefore, widespread use is made of the *Cross Validation* estimation method (Shahraray, 1989). This method as follows. First, reference data is divided into n sets $\{A_1,\ldots,A_n\}$, and then parameters are optimized with A_i. Next, clusters are generated with the optimized parameters. Finally, the cover ratio of reference data without A_i is computed. In this process, we call parameter optimization *learning* and computation of cover ratio *test*.

We performed Cross Validation for i=1,...,n and calculated the average of n test results. For n=2 the size of every A_i was 48 (=96/2), and for n=4 the size of every A_i was 24±1. When n was 9, the size of A_i was not uniform; it varied between 6 and 18 because each A_i was assumed to be a set of reference data housed within each of nine scenes. Figure 6 shows learning and test experimental results, where dashed and solid lines respectively express learning and test cover ratio.

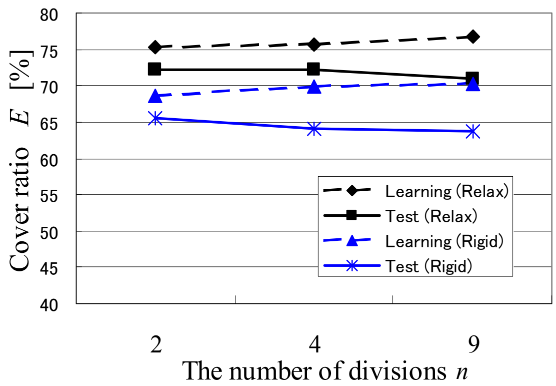

Figure 6. Cross Validation Results

As the figure shows, as n becomes higher the learning cover ratio improves, in contrast to the test cover ratio, which decreases due to the reduced amount of learning data. Even for an n value of 9, however, the Relax method achieved a cover ratio of 70.96%, which was not much worse than 75.25% result obtained when the parameters are optimized with all reference data.

5. CONCLUSIONS

In this paper we proposed a fully automatic method of recovering shapes from a 3-D point set to reconstruct a of 3-D city model of buildings in urban areas. The features of the proposed method are 1) it determines the number of layers based on the MDL principle and 2) it refines clusters based on the cluster shapes. We estimated the validity of our method with 3-D point sets and reference data acquired from nine aerial image sequences in a dense urban area. When parameters were optimized with all reference data, the Relax method achieved a cover ratio of 75.25%, confirming the effectiveness of cluster refinement in comparison to the Rigid method. Furthermore, a cover ratio of 70.96% was achieved even if only one-ninth of the available reference data could be used for optimization. Experimental results showed that our point-based method recovers the shapes of buildings in urban areas effectively.

APPENDIX A. DERIVATION OF CODE LENGTH *M*

In this section we elaborate on the derivation of M ($=M_m+M_d$) in Equation (1) with the same symbols used in Subsection 2.3. We consider only Z coordinates of all points as information being encoded and transmitted. Then Z(p) for every point p is divided into three kinds of information: **a**. a selection of cluster C that includes p, **b**. the height h of the model recovered from C, and **c**. the difference between h and Z(p).

The code length that is necessary to divide a point sequence of n points into k layers is $\log_2 {}_{n\text{-}1}C_{k\text{-}1}$. Once layers are determined, clusters can be obtained with X and Y coordinates. It follows that $\log_2 {}_{n\text{-}1}C_{k\text{-}1}$ is long enough to encode information **a**. And according to the MDL principle (Rissanen, 1984), $(m/2) \log_2 n$ is a code length sufficiently long for information **b**, considering that m is the number of Z(C) which are parameters of models. Finally, we define code length for model description M_m as $(\ln {}_{n\text{-}1}C_{k\text{-}1} + (m/2) \ln n) / \ln 2$.

Next we explain data description length M_d. Let $h_{i,j}$ be the height of the upper surface of a 3-D model recovered from a cluster $C_{i,j}$; and assume that probability $Q(p)$ for the height error $\delta Z(p)=Z(p)-h_{i,j}$ ($p\in C_{i,j}$) obeys Normal Distribution whose standard deviation is σ. To minimize code length we should employ Z($C_{i,j}$) as $h_{i,j}$ because Z($C_{i,j}$) is the maximum likelihood estimator for $h_{i,j}$. According to Shannon's information theory, if the occurrence probability of data is P, code length $-\log_2 P$ is long enough to encode the data. Assuming that $Q(p)$ are independent for all point p, minimal code length $M_d(i,j)$ for cluster $C_{i,j}$ is the following:

$$M_d(i,j) = -\log_2\left(\prod_{p\in C_{i,j}} Q(p)\right) = \frac{1}{2\ln 2\sigma^2}\sum_{p\in C_{i,j}}\left(Z(p)-Z(C_{i,j})\right)^2 + const. \quad (8)$$

$M_d(i,j)$ is the sum of minimal code lengths for information **c** for all points in $C_{i,j}$. We can obtain M_d using the non-constant part of $M_d(i,j)$ as Equation (9):

$$M_d = \frac{1}{2\ln 2\sigma^2}\sum_{i=1}^{k}\sum_{j=1}^{m(i)}\sum_{p\in C_{i,j}}\left(Z(p)-Z(C_{i,j})\right)^2 \quad (9)$$

Finally, M is obtained removing ln2, the common constant coefficient of M_m and M_d.

REFERENCE

Bradski, G., R. and Pisarevski, V., 2000. Intel's Computer Vision Library: Applications in calibration, stereo, segmentation, tracking, gesture, face and object recognition. In: *CVPR2000*, pp.796-797.

Christy, S., and Horaud, R., 1996. Euclidean Shape and Motion from Multiple Perspective Views by Affine Iterations. In: *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol.18, No.11, pp.1098-1104.

Grimson, W., E., L. 1983. An Implementation of a Computational Theory of Visual Surface Interpolation. *Computer Vision Graphics and Image Processing*, Vol. 22, pp. 39-69.

Henricsson, O., 1998. The Role of Color Attributes and Similarity Grouping in 3-D Building Reconstruction. In: *Computer Vision and Image Understanding*, Vol. 72, No. 2, pp. 163-184.

Herman, M., 1986. Repersentation and Incremental Construction of a Three-Dimensional Scene Model. In: *Techniques for 3-D Machine Perception*, pp.149-183.

Leclerc, Y., G, 1990. Region Grouping Using the Minimum-Description-Length Principle. In: *Image Understanding Workshop*, pp. 473-481.

Ledur, R., Maresch, M., and Lesher, C., 1998. 3-D Reconstruction of Urban Environments from Dense Digital Elevation Models. In: *Image Understanding Workshop*, pp. 589-595.

Miyagawa, I., Nagai, S., and Sugiyama, K., 2000. Shape Recovery from Hybrid Feature Points with Factorization Method. *ISPRS 2000*, TC III-01-05, Amsterdam, The Netherlands.

Henricsson, O., 1998. The Role of Color Attributes and Similarity Grouping in 3-D Building Reconstruction. In: *Computer Vision and Image Understanding*, Vol. 72, No. 2, pp. 163-184.

Paparoditis, N., 1998. Building Detection and Reconstruction from Mid- and High-Resolution Aerial Imagery. In: *Computer Vision and Image Understanding*, Vol. 72, No. 2, pp. 122-142.

Rissanen, J., 1984. Universal coding, information, prediction and estimation. *IEEE Trans. Infomation Theory*, 30(4), pp. 629-636.

Terzopoulos, D. and Vasilescu, M., 1991. Sampling and Reconstruction with Adaptive Meshes. In: *CVPR 1991*, pp. 70-75.

Press, W., H., Flannery, B., P., Teukolsky, S., A., and Vetterling, W., T., 1992. *Numerial Recipes in C*. Cambridge University Press, Cambridge, pp. 412-420.