Automatic Generation System for Primitive Building Models by the Integration of CG and GIS
Kenichi Sugihara, Member, IEEE, Yoshitugu Hayashi

Abstract—When the real urban world is projected into the 3-D virtual space, buildings are major objects in this space. To realize the 3-D urban model in this space, it is important to produce building models efficiently. 3-D urban model is the important information infrastructure that can be used for various purposes, such as, simulator for landscape evaluation, community planning and civil engineering. However, in order to realize 3-D urban model, the enormous time and labor have to be consumed to design the model and to acquire the spatial data for the model. In this paper, we propose the system to generate 3-D Primitive Building Models automatically by the Integration of CG and GIS. The programs have been developed to process the 2-D GIS information. Another program on the side of 3-D CG receives the processed data and generates 3-D Primitive Building Models automatically.

Index Terms—3-D Urban Model, Automatic Generation, Computer Graphics, Landscape Evaluation

1. INTRODUCTION

3-D urban model by CG are the important information infrastructure that can be utilized in various fields, for example, landscape evaluation, city planning, civil engineering, architecture, disaster prevention simulation, etc. In addition, disclosure of information about public projects to the public in order to encourage their participation in urban planning is a new application area where 3-D urban model can be of great use. However, in order to realize 3-D urban model, enormous time and labor have to be consumed to acquire the spatial data and to design the models. For example, it takes us thousands of hours to generate a landscape that consists of thousands of buildings, supposing that it takes 30 minutes to make one simple building.

To build the urban model, the image data must be acquired by taking the photographs of the construct in the city and then the spatial data must be extracted from the real world. In the field of the remote sensing in which the image data is acquired, surveyors acquire spatial data using the satellite and the aircraft, the motorcar that are equipped with multi-sensor system. The sensor technology such as laser and digital camera, CCD is improving rapidly and also the individual becomes able to get a satellite photograph, too, and the trials which build the 3-D urban model using the image data in the city were accomplished [1]-[3]. However, it is by the present situation that, in the wide area, a minute urban model is not gotten from the 3-D shape acquired in remote sensing.

Except for models based on remote sensing techniques, in most of the present 3-D urban models, 3-D CG objects are mainly created by the manual operation, using CAD and CG software. However, some efforts exist to integrate CAD and GIS to build 3-D urban models such as those of Gruber [4]. Also, Ueda et tried to generate 3-D CG data of buildings automatically using OpenGL graphic library based on GIS data [5]. However, these efforts to generate 3-D urban models automatically using graphic libraries need a great deal of programming. There are few large-scale and detailed 3-D urban models that have been completed till now.

2. FLOW OF THE AUTOMATIC GENERATION SYSTEM

In order to realize 3-D urban model, the 3-D shapes and material attributes of the buildings must be restructured. However, the manual creation of the shapes and the mapping of texture data to the buildings that composes 3-D urban model require lots of labor and time. Therefore, in our research, a program has been developed using 2-D GIS software components to pre-process the buildings' contours i.e. the building polygons on GIS. Pre-process includes filtering the vertices of the building polygon, dividing the complicated polygon into primitive ones, generating the inside contour to form the walls of the building and exporting the coordinates of the polygons' vertices and the attributes data of the buildings. Another program on the side of 3-D CG receives the processed data and generates 3-D CG building model automatically. The GIS data include the vector spatial infrastructure data (digital maps) of the Geographical Survey Institute and the Digital Residential Maps (Zenrin). The flow of the automatic generation system for building model is shown in Fig.1.

Our system generates 3-D building model, depending on the spatial data of building polygon and attribute data linked with building polygon on 2-D digital map administrated by GIS. For example, after CG module importing the position of vertices of building polygon and attribute data, our system recognizes these data as the design specifications and produces the 3-D CG buildings, following these specs such as the number of stories, the edge length of the polygon and the image num of the roof.

With the rapid progress of remote sensing and photogrammetrical measurement technology, we can get huge amount of digital data, namely, 3-D point cloud that are not structured and to be recognized as a certain object. On the other hand, in GIS, points collection are structured to form polygons or polylines according to geometry, topology and semantics. Our system executes the generation process of 3-D building model after receiving the data that are recognized as building polygon from GIS. In the generation process, the system estimates and realizes 3-D CG model alternative for the real, because 2-D polygon data and attached attribute data are the

Kenichi Sugihara is with the Faculty of Business Administration, Gifu Keizai University, Gifu, Japan (e-mail: sugihara@gifu-keizai.ac.jp).
Yoshitugu Hayashi is with the Graduate School of Environmental Studies, Nagoya University, Aichi, Japan (e-mail: yhayashi@genv.nagoya-u.ac.jp).


Presented at URISA, IAAO (International Association of Assessing Officers)
The 7th Annual Integrating GIS & CAMA 2003 Conference, Ohio
2) Calculation of the component of the normalized forward and backward edge vector.
3) Calculation of the component \((vix, viy)\) of the normalized sum vector of the normalized forward and backward edge vector.
4) If the edge turns right then
   The component of the vector for inside contour is \((vix, viy)\)
   else the component is \((-vix, -viy)\) end if;
5) \(pout().x = pt().x + wid*vix; pout().y = pt().y + wid*viy;\)
6) \(pout().x = pt().x - wid*vix; pout().y = pt().y - wid*viy;\)

![Original building polygon](image)

**Fig. 2 Generation of inside & outside contour**

For circular building polygon that has successive flat points, the radius and center of the circle or the center and the minor and major axes of the ellipse are estimated in GIS module. The circular or elliptic cylinder will be placed to them so that cylindrical building model will be generated in CG module. Since circular polygon has as much as 360 points in GIS, the inside and outside contour are not used for the generation of cylindrical building to avoid to use the bulky data.

3.3 Proposed Polygon Representation

For the majority of the building polygons, the angles of the vertices are almost 90 or 270 degrees and the number of vertices is limited. The building polygon that has limited num. of vertices and only almost 90 or 270 degrees angles is replaced by the combination of rectangles at GIS module. The vertices and segments of the polygon are numbered clockwise. When following the segments of the polygon clockwise, any segment of a polygon will have a bending angle of 90 degrees to the right or to the left relative to the preceding segment. So, it is possible to assume that the building polygon with almost 90 or 270 degrees angles can be expressed as a set of changes of the segments' direction.

An example of a building polygon is expressed as a set of changes as shown in the Fig. 4. In this example, R and L mean a change of the segments' direction to the right and to the left, respectively.

![Building Polygon](image)

**This polygon can be expressed as RLRRLLRRLLR.**

R means the turn to the right. L means the turn to the left.

**Fig. 4 Expression of a building polygon as a set of changes in the direction of its segment**

For the building polygons that can be described as above RL expression, the following relationship stands up among the number of the vertices, the number of the right turn segments and the number of the left turn segments.

\[(\text{Num. of right turn segments}) + (\text{Num. of left turn segments}) = \text{Num. of the vertices} \]

\[(\text{Num. of right turn segments}) - (\text{Num. of left turn segments}) = 4\]

The number of shapes that a polygon can take depends on the number of the vertices that polygon has. By the formula of circular permutation, we can calculate the number of the patterns of shape that polygon may take according to the number of the vertices of the polygon. In case of a six vertices polygon, the direction set of the polygons is LRRRRR. Since the left turn segment appears only once, the shape pattern is unique, that is, L-shape. This L-shaped polygon is the basic polygon into which the polygon with more vertices is subdivided.

In case of an eight vertices polygon, the following four kinds of polygons are possible.
3) Between the two edges that are adjacent to Lmax, shorter edge will be the width (W_box1) of Box1.
4) The edge that is adjacent to L_box2 and is not Lmax will be the width of Box2 (W_box2).
5) The length of Box2 will be reduced to (L_box2 - W_box1).
6) The length of the roof of Box2 (Rooft2) will be
   \[ \frac{W_{\text{box1}}}{2} \]
7) The point Prf2 that is the position control point of Rooft2 will be calculated according to the following expression.

   In case that \( P(1)P(2) >= P(3)P(4) \)
   \[ \text{Prf2} = \frac{P(1) + (P(2) - P(1)) \times \frac{L_{\text{box2}} - 0.5 \times W_{\text{box1}}}{L_{\text{box2}}}} \]

   In case that \( P(1)P(2) < P(3)P(4) \)
   \[ \text{Prf2} = \frac{P(1) + (P(2) - P(1)) \times \frac{0.5 \times W_{\text{box1}}}{L_{\text{box2}}}} \]

Fig.6 shows the examples that the above algorithm is applied to L-shaped polygon.

**Fig. 6** Examples of assignment of Roofs and Boxes to 6 vertices polygon

4.3 How to divide the polygon

According to the polygon consisting of more than 8 vertices, the polygon will be divided into the center area and attached branches. The polygon is supposed to be expressed as the dataset of turning direction of edges. In case that dataset take L (Left turn) after or before consecutive R (Right turn), we assume this pattern as the branch. In other words, we take notice of the vertex that turns conversely. From this vertex, the dividing line will be drawn. For example, "*RRL*" pattern will be recognized as the branch. From 'L' vertex, the dividing line will be drawn to the backward direction in terms of the vertices that are numbered clockwise. Also, "*LRRR*" pattern will be recognized as the branch. From 'L' vertex, the dividing line will be drawn to the forward. After the polygon being broken down to L-shaped polygon and rectangles, the algorithm for assigning to L-shaped polygon will be applied to broken-down L-shaped polygon. At first, we have adapted this dividing algorithm to the polygon with 8 vertices (8 vertices polygon). Since 8 vertices polygon takes 4 kinds of shape according to RL expression, the algorithm has been applied to 4 cases respectively.

**Fig. 7.** 3 shape types of 8 vertices polygon that have no reiteration pattern

These 3 types have no reiteration pattern. In these types, the module looks up the vertex that turns to the left ('L') after the consecutive R (Right turn), from the vertex, the dividing line will be drawn to the backward direction. From the next vertex that turns to the left ('L'), the dividing line will be drawn to the forward direction.

**Fig. 8.** 8 vertices polygon that has reiteration pattern

In the type that has reiteration pattern of 'LRRR', there are two shapes of the polygon as shown Fig.8. In the RL expression, from the two vertices that turn to the left ('L'), the dividing line will be drawn to the same direction. We distinguish two types by the sum of the length of the two edges before 'L' vertex. In this case, these two patterns cannot be distinguished by the RL expression. We have to consider the length of the preceding edges of 'L' vertex.

After the intersections between the edge and the dividing line are calculated, the branches are divided from the center area. The coordinates of six vertices of pruned polygon are given as the candidate coordinates set. The algorithm for assigning to