GIS and CG Integrated System for Automatic Generation of 3-D Urban Models

Kenichi Sugihara　　Yoshitugu Hayashi

【Abstract】3-D Urban Models are 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 money have to be consumed to design the model and to acquire the data for the model. In this paper, we propose the system to generate 3-D Urban Models automatically, utilizing and integrating GIS and CG. 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 Urban Models automatically.

【Keywords】 (1) GIS(Geography Information System) (2) CG(Computer Graphics) (3) Landscape Evaluation (4) 3-D Urban Model (5) Automatic Generation

1. INTRODUCTION

3-D urban models by CG, that is *Spatial data infrastructure*, are the important information background that can be utilized in several fields, for example, landscape evaluation, city planning, civil engineering, education, 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 models can be of great use. However, in order to realize 3-D CG urban models, enormous time and money have to be consumed to acquire the 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 al. 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 and other objects must be restructured. However, the manual creation of the shapes and the mapping of texture data of the objects that composes 3-D urban models 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 objects on GIS such as filtering the vertices of the buildings' polygon, dividing the polygon into primitive ones, etc. and export the coordinates of the polygons' vertices and the attributes data of the objects. Another program on the side of 3-D CG receives the processed data and generates 3-D CG objects 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 urban model generation system is shown in figure 1.

1.Associate Professor. Gifu Keizai University, Department of Business Administration, sugihara@gifu-keizai.ac.jp
2.Professor. Nagoya University, Graduate School of Environmental Studies, yhayashi@genv.nagoya-u.ac.jp

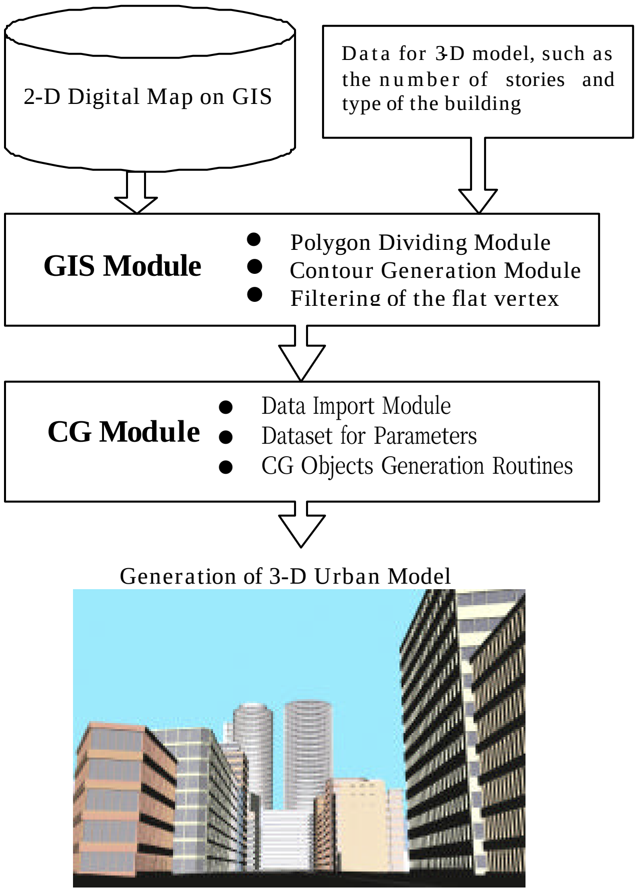


Fig.1 Flow of the Urban Model Automatic Generation System

3. Features of Our System

Our system generates 3-D urban model, depending on the objects (spatial entities by GIS definition) and attached attribute data on 2-D digital map administrated by GIS. For example, after CG module importing the position of vertices of buildings' polygon and attribute data linked to buildings' polygon, 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, color of roof and the type of the building. 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. Since our system generates 3-D urban model after receiving the recognized data from GIS, it may not be called full-automatic but semi-automatic generation. After the object recognition and the import of the position of object and the attribute data, our system executes the generation process. In the generation process, the system estimates and realizes 3-D CG model alternative for the real, because 2-D polygon or polyline data and attached attribute data are the main data source of estimation and they are not complete source of generation. So, we aim at optimal estimation and saving labor that arises from mis-match between estimation and the real.

4. The GIS Module

4.1 Main Function of GIS Module

In terms of geometric point of view, objects are differently classified between GIS and CG. With rigorous definition of 'objects', it is used in the sense of database objects in GIS [7] while, in CG, it is defined as an abstraction that models the state and behavior of entities in a system[8]. Since objects in GIS are related to geometric elements, objects can be treated as the spatial entities that have geometric elements. In GIS, there are only a few objects such as point, polyline and polygon while there are many geometric primitive objects such as box, prism, cylinder, sphere, plane, cone and etc. in CG. In GIS, the rectangular, ellipse, trapezoid and etc. that form the building contour are classified as only one category, that is, polygon. Therefore, it is necessary to convert and pre-process the data of objects on 2-D GIS map for 3-D CG model generation. If building contour has only right angles, it will be subdivided into rectangular and L-shape polygon so that box and prism are placed on these polygon in CG generating process. Another pre-processing is the generation of the inside contour and the outside contour from the building contour given in GIS so that they form the wall and the glass of building. The building walls are formed by extrusion of area between the outside contour and the given contour and the glass of building are formed by extrusion of area between the given contour and the inside contour.

Main function of GIS module is as follows.

1) Exporting coordinates and attributes' data from 2-D GIS to 3-D CG
2) Subdividing building contour (polygon) into rectangular and L-shape polygon if building contour has limited vertices and only right angle
3) Generating the inside and the outside contour in order to form the wall and the glass of the building
4) Filtering out the vertex which has almost 180 degree angle of polygons from 2-DGIS

5) Estimation of the center and the minor and major axes of the ellipse that will be placed on the vertices of circular polygon

4.2 Filtering out the flat vertex

The system calculates the length of edges and the angle of vertices of building polygon for creating 3D buildings. After finding the longest edge in all edges of a polygon, the length of the roof and the box are decided by the longest value. In this searching process, there can be miscalculation caused by polygons that have almost 180 degrees angle vertex (named flat vertex). The flat vertex is meaningless point and should be eliminated. However, in case of the circular building, flat vertices will be in continuation. In our system, the isolated flat vertex will be filtered out and the continual flat vertices will be assigned by circular polygon in the following algorithm.

Algorithm: Flat_vertex_filter(point collection:Pt)

Let 'flat vertex array' contains the number and the position of flat vertex.

Let 'first_count_flag' be the flag for start counting the number of flat vertex in order to count the continual number of flat vertices.

Let 'e_flag' be the flag for erasing the flat vertex. If it is set, the vertex will not be exported.

Input: the coordinates of vertices of polygon (Pt.x , Pt.y), the number of vertices of polygon (nv)

Output: the filtered coordinates of vertices of polygon, the estimated center and the minor and major axes of the ellipse

1) Calculation of angle of all vertices

2) For each vertex of a polygon

3) If angle is almost flat then

4) If first_count_flag is reset then

5) first_count_flag is set and suffix of flat vertex array incremented end if;

6) The number of flat vertex is incremented;

7) If the angle of the last and the first vertex of the series is flat then

8) The first flat vertex array includes the last flat vertex array end if;

9) end if

10) The number of flat vertex and first_count_flag are reset;

11) If flat vertex array contains more than the threshold number of flat vertex for circular object then

12) Estimate the center and the minor and major axes of the ellipse end if.

13) If flat vertex array contains isolated flat vertex then

14) e_flag' is set and the vertex is not be exported;

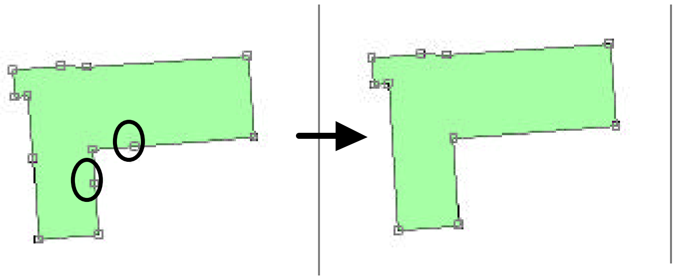


Fig2 The isolated flat vertex filtered out

4.3 Generation of the inside and the outside contour for creating the wall and the glass of building

In order to create the wall and the glass of building, the inside and the outside contour generated from the building polygon are necessary. The building walls are formed by extrusion of the area between the outside contour and the building polygon and the glass of building are formed by extrusion of the area between the building polygon and the inside contour.

The vertex of the inside and the outside contour are constantly distant from the vertex and they are on the line that divides the angle of vertex. The generations of the inside and the outside contour are performed by the algorithm mentioned below. The inside or the outside judgment of contour vertex and the calculation of the component of the normalized angle-bisecting vector are performed here.

Algorithm: Inside & Outside Contour Generation

Let the suffix of the vertex of polygon clockwise.

Let the constant 'wid' the distance between the inside contour vertex and the building polygon vertex

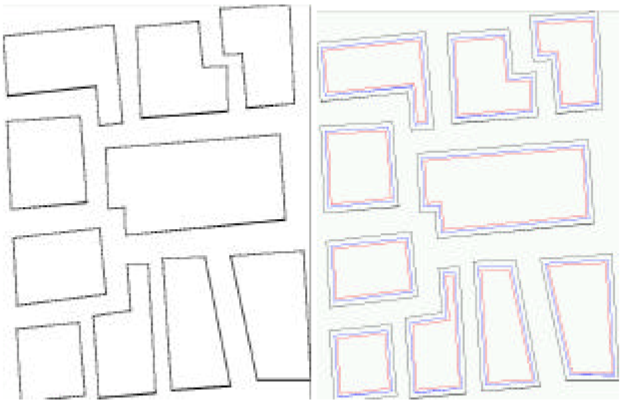
Let pin(),pout() be the vertex of the inside and outside contour respectively.

1) Calculation of outer products of all vertices that decide the turn of the edge, i.e. right turn or left turn.

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
5) The component of the vector for inside contour is (vix,viy) else the component is (-vix,-viy) end if;
*5) pin().x=pt().x + wid*vix ; pin().y=pt().y + wid*viy ;*
6) *pout().x=pt().x - wid*vix ; pout().y=pt().y - wid*viy ;*



The original building polygon The generation of contours

Fig4 The Generation of the inside and the outside contour

For circular building polygons, 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 assigned to them so that cylindrical building will be generated in CG module. In this case, to avoid to use the bulky data of the position of vertices (360 point data for circle), the inside and outside contour are not used for the generation of building.

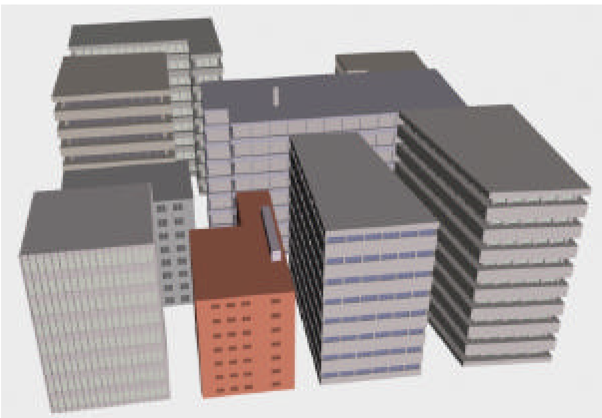


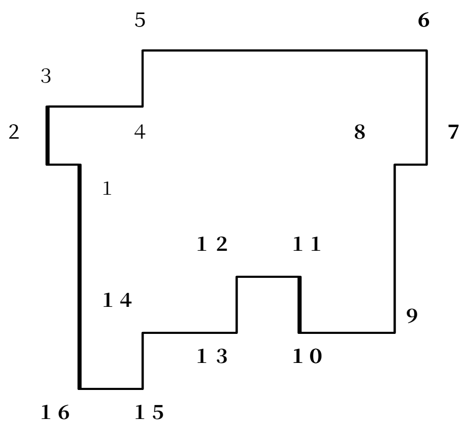
Fig5 The generated 3-D building with the wall and glass and window frame

4.4 Proposed Polygon Representation

For the majority of the building polygons, the angles of the vertices are almost 90 degrees and the number of vertices is limited. This means that any segment of a polygon will have a bending angle of about 90 degrees to the right or to the left relative to the preceding segment. So, it is possible to assume that the most building polygons can be expressed as the combination of several rectangular shapes. Based on this assumption, for the building polygons with the only right angle, we assume that a building polygon can be described by a set of changes in the direction of its segments. 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. When creating a 3-D CG representation of a building, the building polygon in the GIS representation must be divided into several rectangular shapes in order to take advantage of the 3-D CG primitives. This procedure can greatly facilitate the creation of the 3-D CG objects.

As an example, a building polygon can be expressed as a set of changes in the direction of its segments as shown in the Figure 6. In this example, the vertices are numbered clockwise. R and L mean a changes in the direction between vertex i and vertex i+1 to the right and to the left, respectively. For the building polygons with the only right angle, 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.

・ Number of the vertices = (Number of right turn segments) + (Number of left turn segments)
・(Number of right turn segments) - (Number of the left turn segments) = 4



LRRLRRRLRRLLRLRR

Fig.6 Example of the Changes in the Direction of the Segments of a Buildings' Polygon

The number of shapes that a polygon can take depends on the number of the vertices of that 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 polygon will be the element 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.

LLRRRRRR LRLRRRRR LRRLRRRR LRRRLRRR

Based on the previous assumptions, the number of the shapes that a building's polygon may take depends on the number of the vertices. The number of these cases can be given by the following formula of circular permutation.

(1) 10 vertices polygon: $\frac{10!}{7!3!} \times \frac{1}{10} = 12$ case

(2) 12 vertices polygon:

3 reiteration patterns $\frac{3!}{2!1!} \div 3 = 1$ case

6 reiteration patterns $\left(\frac{6!}{4!2!} - \frac{3!}{2!}\right) \div 6 = 2$ case

No reiteration pattern

$$\left(\frac{12!}{8!4!} - \frac{6!}{4!2!}\right) \div 12 = 40 \text{ case}$$

Total number of cases: 1 + 2 + 40 = 43 cases

In proportion to the number of vertices, the number of the case will increase by geometric progression.

5. 3-D CG MODULE

5.1 Assignment of Primitives to Polygon

In many cases, CG objects are often created by Boolean operation of 3-D primitives. In our system, buildings are also created by boolean combinations of primitives. So, the building polygons should be divided into primitives and they are assigned by one or more primitives, in this case, rectangle.

The reasons for creating building CG through boolean combinations of primitives are as follows.

1) The properties of the object such as height, length are acquired and controlled easily.
2) The position and direction of the primitive is easily controlled, because the control point of the primitive is clear.
3) Since primitives are expressed in a simple equation, the data sizes of them are small. Especially it is important when generating a lot of buildings.
4) The measurement error of the polygon can be eliminated by assigning primitives to building polygon.
5) The primitive can be converted into NURBS(Non-Uniform Rational B-Spline) that could be complicated geometry.

5.2 Algorithm for assigning building to polygon

The polygon model will be based on the following conventions as the geometric structure.

1) For each pair of nodes, there is at most one edge that connects them directly.
2) Edges must not intersect. If two edges do intersect, then they should be replaced by four edges joining at a node.
3) One edge should not end on another edge. If the begin or end node of one edge is in another edge, another edge should be split in two so that the first edge and the two new edges meet at one node.

The algorithm for assigning buildings and roofs to polygon is started with the algorithm for assigning to polygon that has 6 vertices (6 vertices polygon). 6 vertices polygon will be divided into two rectangles and two boxes(Box1,Box2) are formed on these two rectangles shown at Fig.7. Also, the two roofs(roof1, roof2) are formed and placed on the two boxes.

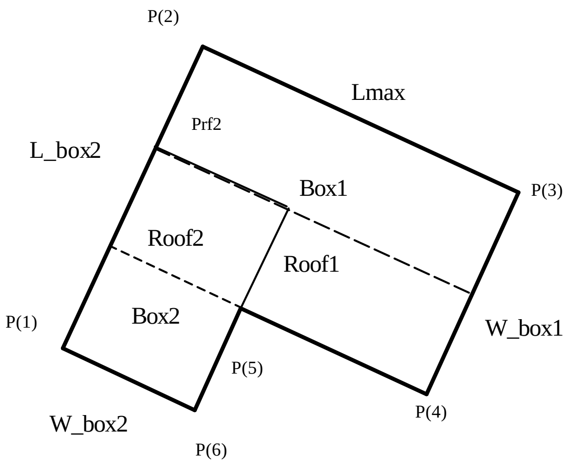


Fig.7 6 vertices polygon divided into Box1, Box2 and assigned by roof1,roof2

The length and direction of the rectangle are supposed to be that of the longer edge of the rectangle.

The algorithm for assigning to 6 vertices polygon is as follows:

1) *After finding the longest edge(Lmax) among 6 edges, it will be the length and direction of Box1.*

2) *Between the two edges that are adjacent to Lmax, longer edge will be the length(L_box2) and direction of Box2.*

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 (Roof2) will be*

$$\left(L\_box2 - \frac{W\_box1}{2} \right).$$

7) *The point Prf2 that is the position control point of Roof2 will be calculated according to the following expression.*

In case that distance of P(1)&P(2) >= distance of P(3)&P(4)

$$Prf2 = P(1) + (P(2) - P(1)) * \frac{L\_box2 - 0.5 * W\_box1}{L\_box2}$$

In case that distance of P(1)&P(2) < distance of P(3)&P(4)

$$Prf2 = P(1) + (P(2) - P(1)) * \frac{0.5 * W\_box1}{L\_box2}$$

Fig.8 shows the examples that the above algorithm is applied to 6 vertices polygon.

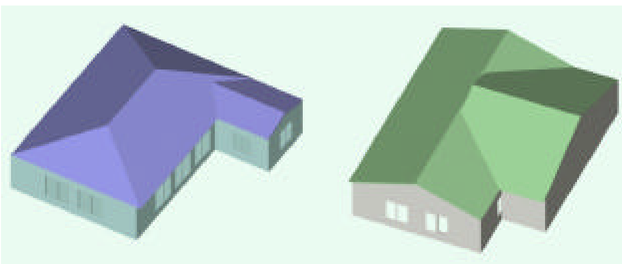
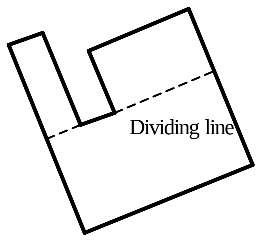


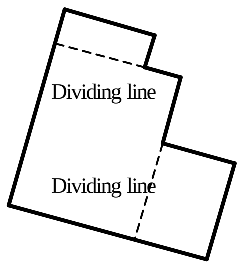
Fig. 8 Examples of assignment of two Boxes and two roofs to 6 vertices polygon

5.3 How to divide the polygon

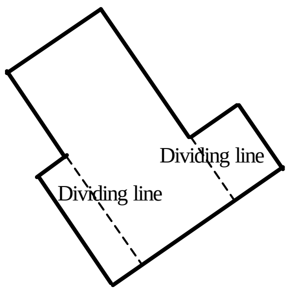
According to the polygon consisting of more than 6 vertices, the polygon will be divided into the center area and attached branches. The polygon is supposed to be expressed as the dataset of turning direction of edges. In case that dataset take L (Left turn) after consecutive R(Right turn), we assume this pattern as the branch. For example, *RRL* will be recognized as the branch. In other words, we take notice of the vertex that turns conversely. From this vertex, the dividing line will be drawn to the backward direction in terms of the index order of the vertices. The vertex is indexed in clockwise in terms of centroid of the polygon. After being broken down to 6 vertices polygon, the algorithm for assigning to 6 vertices polygon will be applied to broken-down 6 vertices polygon. The more the vertices of the polygon are, the more the number of cases of how to pick out the vertices are. First of all, we have adapted this dividing algorithm to polygon consisting of 8 vertices (8 vertices polygon). Since 8 vertices polygon takes 4 kinds of shape patterns, the algorithm has been applied to respectively.



LLRRRRRR type



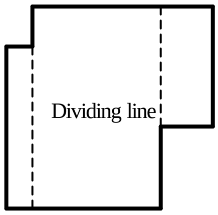
LRLRRRRR type



LRRLRRRR type

Fig 9. 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 after the consecutive vertex that turns to the right. From this vertex, the dividing line will be drawn to the backward direction. From the next vertex that turns to the left, the dividing line will be drawn to the forward direction.



LRRRLRRR type

Fig 10. 8 vertices polygon that has reiteration pattern

In the type that has reiteration pattern, the module also looks up the vertex that turns to the left. From this vertex, the dividing line will be drawn to the backward direction. From the next vertex that turns to the left, the dividing line will be drawn to the backward direction.

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 6 vertices polygon will be applied to this candidate coordinates set. In case of 8 vertices polygon, there are 2 sets of candidate coordinates.

Here are the examples that the above algorithm is applied to 8 vertices polygon.

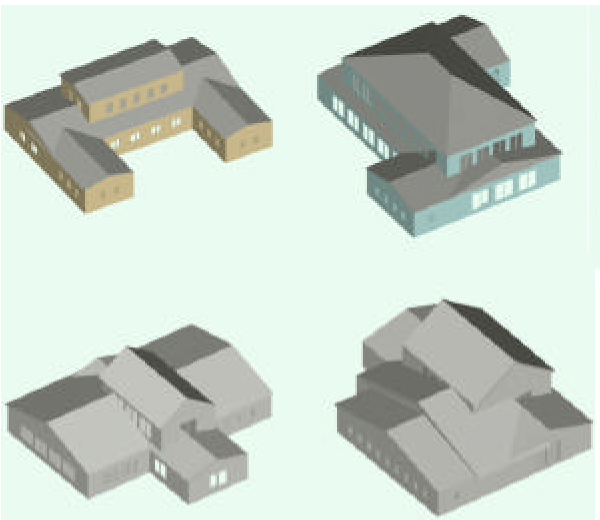


Fig 11. Examples of assignment of three Boxes and three roofs to 8 vertices polygon

5.4 Building Generation Module

CG module that controls 3-D CG software will automatically generate the component of the city model, based on the data imported from the GIS module. The components are created through Boolean operations of various CG objects. CG objects are defined as the instances of CG primitive class, such as box or prism, shown Fig 12.

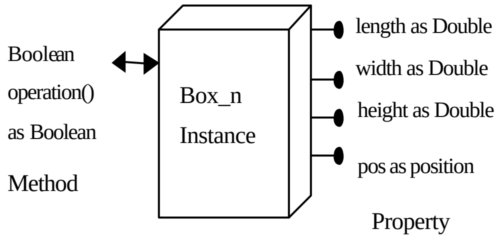


Fig 12. Data structure of Box object

The shape and the position of the CG object is controlled by setting the value to properties.

CG module will generate the component of the city, according to the type of building, the number of vertex of polygon. The type of building is classified into several houses, such as the house with gable roof, the house with hipped roof or the house with the lean to roof. Although the number of parameters for specifying the shape of house is many, we cannot have too many fields in GIS, considering the amount of thousands of building that will be automatically generated. So the number of parameters is limited and we aim at generating the simple shaped houses. With respect to the house with the gable roof, the parameter sets are defined as follows.

1) height of Box_n 2) standard interval of windows at longer edge of Box_n 3) standard interval of windows at shorter edge of Box_n 4) horizontal (or vertical) ratio of window width to standard interval at longer edge of Box_n 5) horizontal (or vertical) ratio of window width to standard interval at shorter edge of Box_n
6) vertical position of window 7) gradient of the roof_n
8) ratio of top edge to the length of the hipped roof
9) ratio of the length of Box3(second floor part) to the length of Box1 10) ratio of the width of Box3 to the length of Box1 11) position of Box3 to Box1

Even if the number of parameters is limited, the parameters are necessary for building units (from Box1 to Box3) that constitute the building. Since the number of parameter is many in the end, parameter sets are placed in the program as the data table. The dataset is delivered from the table, according to the attribute data from GIS. The shape of the building is decided through the fieldwork or the survey of the photograph from the sky. The surveyor selects the most resemble building among the candidates that are generated from various parameter sets. For example, the houses that have the different set of parameters and are generated from the same polygon have resulted in a variety of houses shown in Figure 13.

Fig 13. Houses that have different set of parameters and are generated from the same polygon

6. CONCLUSION

In this paper, we proposed the system that automatically generates 3-D urban model, based on the data from the GIS that accumulates a variety of attributes of shape, the position of the buildings. The system we developed consists of two modules, GIS module, CG module. Main roles of GIS module are to divide the building polygons into primitives and to export attribute data that GIS accumulates. CG module will create buildings, houses with roofs and roads, based on the coordinates and the attributes data of the objects that GIS module exports.

3-D urban model reflecting the real 3-D world offers us the important tool to display the results of simulation, experiments and alternative plan. This Model can act as the simulator to realize the alternative ideas of urban planning virtually or to examine the current zoning system, building regulations and building agreements. The system automatically generates 3-D Urban Model that is quite effective not only for specialist of urban planning but also for residents, citizen or students in understanding what will be built, what image of the town will be or what if this alternative plan is realized.

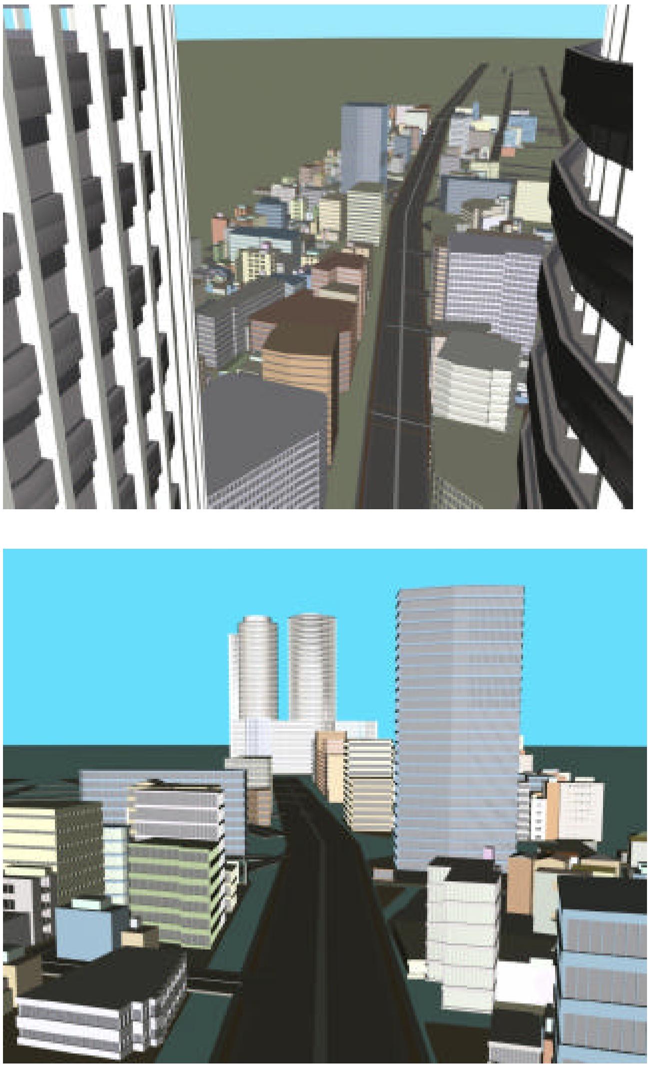


Fig 14. Automatically generated 3-D Urban Model (Sakura Boulevard, Central Towers, Nagoya)

REFERENCES

[1] Mikito Notomi, Shiro Ozawa, 1998. "Modeling of Urban Scene by Motion Analysis," The Transaction of the Institute of Electronics, Information and Communication Engineers, Vol.J81-D-2,No.5,pp872-879

[2] Chunxiao Li, 1998. " Building 3-D Information Acquisition from 2-D Spatio-Temporal-Image," UM3 (UrbanMulti-Media/3-Dmapping) '98, pp118-124

[3] Ryosuke SHIBAZAKI, 1998. " Automatic Object Extraction from Simulated High Resolution Satellite Imagery," UM3 (Urban Multi-Media /3-D Mapping)'98, pp1-6

[4] Michael GRUBER, 1998, "The Cyber-City Concept from 2-D GIS to the Hypermedia Database", UM3(Urban Multi-Media/3-D Mapping)'98, pp1-6 , UM3

[5] Minoru Ueda, Takashi Hoshi, 1999. "An Efficient Data Acquisition and Database Design for Urban Landscape Simulation," The Transaction of the Japan Society of Civil Engineers, Vol.8

[6] Martien Molenaar, 1998. "An Introduction to the Theory of Spatial Object Modelling" Taylor&Francis Ltd, 1Gunpowder Square, London, pp.2-3

[7] Will Schroeder, Ken Martin, Bill Lorensen, 1998. "Visualization Toolkit 2nd Edition An Object-Oriented Approach to 3D Graphics", pp619 , Prentice Hall PTR Upper Saddle River,NJ07458