AUTOMATIC GENERATION OF 3-D BUILDING MODELS FROM BUILDING POLYGONS ON GIS

Kenichi Sugihara[1], **Yoshitugu Hayashi**[2]

ABSTRACT

When a real urban world is projected into 3-D virtual space, buildings are major objects in this space. To realize a 3-D CG urban model, it is important to generate building models efficiently. A 3-D urban model is an important information infrastructure that can be utilized in several fields, such as landscape evaluation, urban planning, disaster prevention simulation, etc. However, in order to realize a 3-D urban model, enormous time and labor has to be consumed to acquire the spatial data and to design the models. In this paper, we proposed the GIS and CG integrated system that automatically generates a 3-D building model from building polygons on a digital map stored by GIS. Some building polygons contain noise edges to be filtered out. The algorithm for filtering out noise edges and breaking down complicated polygon into primitive ones is presented. By applying the algorithm, we can get a core polygon with smaller vertices, which then can be broken down into a basic L-shaped polygon and rectangles. After breaking down, the integrated system is placing CG primitives i.e. building parts on these polygons to form a 3-D building model.

KEY WORDS

GIS, 3-D urban model, Automatic Generation, Computer Graphics

1.INTRODUCTION

When simulating a current city by a 3-D urban model, buildings are major objects in this model. A 3-D urban model is an important information infrastructure that can be utilized in several fields, e.g., landscape evaluation and city planning, architecture, disaster prevention simulation, etc. In addition, disclosure of information about public projects to the public in order to encourage their participation in city planning is a new application area where a 3-D urban model can be of great use. However, in order to realize a 3-D urban model, enormous time and labor has to be consumed to acquire the spatial data and to design the models. For example, when manually modeling a house with roofs by constructive solid geometry (CSG), one must follow these laborious steps : 1) generation of primitives of appropriate size, such as box, prism or polyhedron that will form parts of a house 2) boolean operation among these primitives to form the shapes of the roof and the house body 3) rotation of parts of the house 4) positioning of parts of the house 5) texture mapping to these parts

[1] Professor, Faculty of Business Administration, Gifu-Keizai University, 5-50 Kitagata-chou Ogaki-city Gifu-Pref., 503-8550 JAPAN, Phone & FAX +81 584-77-3598, sugihara@gifu-keizai.ac.jp

[2] Professor, Graduate School of Environmental Studies, Nagoya University, Furou-chou Chikusa-ku, Nagoya 464-8603 Japan, Phone +81-52-789-2772, Fax +81-52-789-3837, yhayashi@genv.nagoya-u.ac.jp

In order to save the steps mentioned above, we aim at generating 3-D building models automatically from buildings' contours, i.e., building polygons on a 2-D digital map. In the generation process, the sources of 3-D building models are 2-D digital maps stored and administrated by GIS. 2-D digital maps are made by map production companies. Building polygons on a digital map are drawn manually with a digitizer, depending on the aerial photos. In the case of city planning, an urban designer may follow the traditional way of drawing building polygons for the future layout of a city.

Building polygons can be extracted from aerial images by an image recognition software. Aerial images are also important sources of 3-D building models. However, due to the complexity of natural scenes and the lack of performance of image recognition software, fully automated methods cannot guarantee results stable and reliable enough for practical use (Gruen and Wang, 1998). Recently, many approaches for automated and semi-automated extraction and modeling of buildings from aerial images have been proposed (Gruen et al., 2002 ; Suveg and Vosselman, 2002 ; Fischer et al., 1997).

Gruen and Wang (1998) introduced a semi-automated topology generator for 3-D building models: CC-Modeler. Feature identification and measurement with aerial images is implemented in manual mode. During feature measurement, measured 3-D points belonging to a single object should be coded into two different types according to their functionality and structure: boundary points and interior points. After these manual operations, the faces are defined and the related points are determined. Then the CC-Modeler fits the faces jointly to the given measurements in order to form a 3-D building model.

Suveg and Vosselman (2002) presented a knowledge-based system for automatic 3D building reconstruction from aerial images. The reconstruction process starts with the partitioning of a building into simple building parts based on the building polygon provided by 2D GIS map. If the building polygon is not a rectangle, then it can be divided into rectangles. A building can have multiple partitioning schemes. To avoid a blind search for optimal partitioning schemes, the minimum description length principle is used. This principle provides a means of giving higher priority to the partitioning schemes with a smaller number of rectangles. Among these schemes, optimal partitioning is 'manually' selected. Then, the building primitives of CSG representation are placed on the rectangles partitioned.

These proposals and systems will provide us with a primitive 3D building model with accurate height, length and width, but without details such as windows, eaves or doors. The research on 3D reconstruction is concentrated on reconstructing the rough shape of the buildings neglecting details on the facades such as windows, etc(Zlatanova and Heuvel, 2002). On the other hand, there are some application areas such as urban planning and disaster prevention simulation where the immediate creation and modification of many plausible building models is requested to present the alternative 3-D urban models. When generating these models, automatic modeling of many future buildings is to be desired. In these models, approximate 3D building models with details are sometimes necessary to encourage the public involvement so that people in general can recognize them as buildings of alternative urban plans. Usually and traditionally, urban planners design the future layout of a town by drawing a map. So, it is convenient for urban planners who draw maps if the map can immediately be converted into a 3-D urban model. In our research, we aim at creating 3-D urban models, especially building models automatically from 2D building contours on a digital map. The system automatically generates a 3-D urban model so quickly that it meets the urgent demand to realize another alternative plan.

2. FLOW OF THE AUTOMATIC GENERATION SYSTEM

The automatic generation system consists of GIS application (ArcView, ESRI Inc.), GIS module and CG module as shown in Fig.1. The source of a 3-D urban model is a digital residential map that contains building polygons linked with attributes data such as the number of stories, the type of roof.

The GIS module pre-processes building polygons on the digital map. Pre-process includes filtering out 'noise edge' (noise edge that is between long edges of almost the same direction) and unnecessary vertices of a building polygon, dividing a complicated polygon into primitive ones, generating inside contours to form walls and glasses of a building and exporting the coordinates of polygons' vertices and attributes of buildings. The attributes of buildings consist of the number of stories, the image code of roof, wall and the type code of roof (flat, gable roof, hipped roof, gable roof with longer width, gambrel roof and penthouse). The GIS module has been developed using 2-D GIS software components (MapObjects, ESRI).

The CG module receives the pre-processed data that the GIS module exports, generating 3-D building models. In case of modeling a building with roofs, the CG module follows these steps: 1) generation of primitives whose size are determined by the preprocessed data 2) boolean operation among these primitives to form the shapes of the roof, the building body, etc. 3) rotation of parts of the building 4) positioning of parts of the building 5) texture mapping to these parts according to the attribute received 6) copying the 2nd floor to form the 3rd floor or more in case of building higher than 3 stories. CG module has been developed using Maxscript that controls 3-D CG software (3D Studio VIZ, Autodesk Inc).

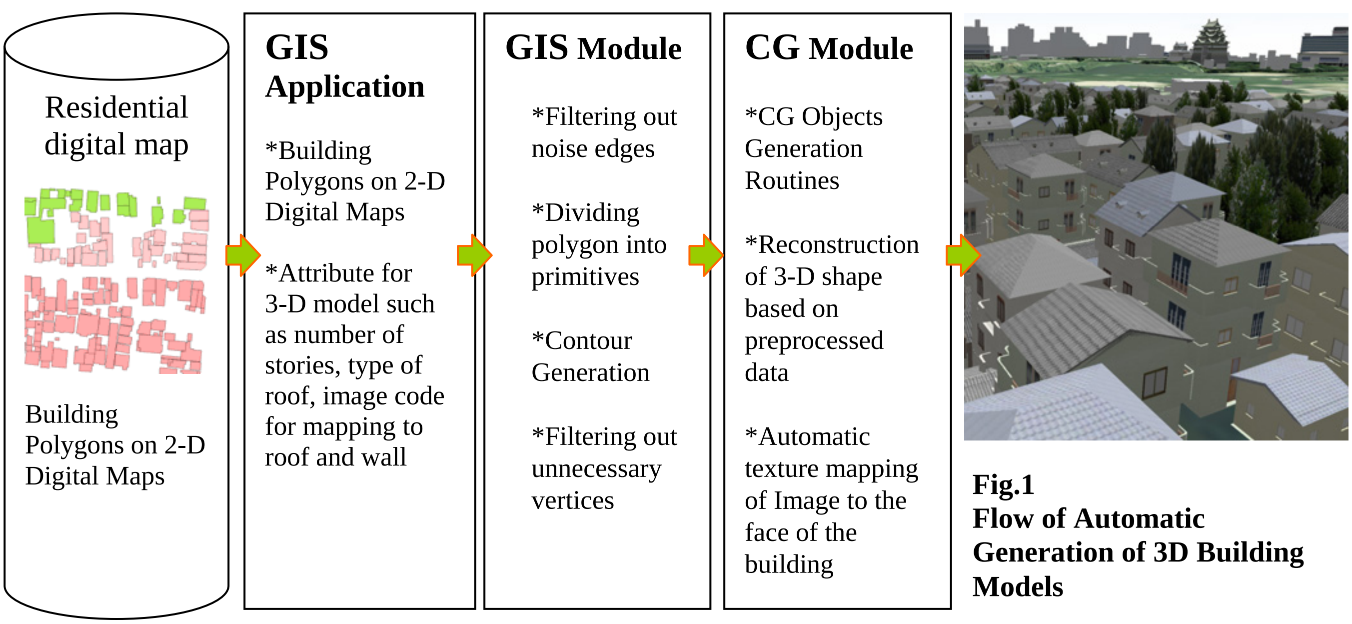

Fig.1
Flow of Automatic Generation of 3D Building Models

3.THE GIS MODULE

3.1 Filtering of ‘noise edge’

Satellite and aircraft equipped with CCD cameras and electromagnetic sensors are daily monitoring and taking photos of the earth. Thanks to the advancement of remote sensor technology, we can easily acquire satellite and aerial imagery. Digital image processing and pattern recognition research of these photos are in full swing. Some image recognition software provides us with image object extraction as shown in Fig.2 (eCognition, Definiens Inc.). On the other hand, at map production companies, many technicians are drawing digital residential maps. The building polygons shown in Fig.3 on the residential map are drawn manually with digitizer, depending on the aerial photos. Since the images that recognition software provides contain too much noise to be filtered out, we use residential maps for the source of a 3-D urban model. These maps show that edges of a building polygon are not always straight lines. Many building polygons contain ’short edges’ that are between long edges of almost the same direction. We call short edge ‘noise edge’.

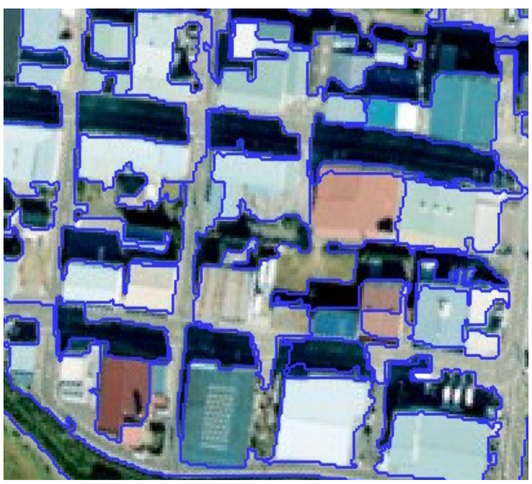

Fig.2 Building contour extracted from aerial photo by pattern recognition (eCognition, Definiens Inc.)

Fig.3 Building polygon on digital residential map drawn by map producer

Noise edge exists because of the shape of roofs or eaves of a house or jaggies of the image recognition system. However, inside the noisy building polygon, there is a combination of boxes whose contours are not jagged. In other words, under the complicated shape of roofs or eaves, there may exist a simple box-combined house. After filtering out noise edges, core polygons are given inside original polygons. The CG and GIS integrated system will generate 3-D roof models from an original building polygon and a 3-D house body model from the core polygon filtered.

The process for filtering out noise edge is executed as follows:

1) The process tries to find the range where normal edges (longer than threshold length) of almost the same direction are in sequence. We call the range ‘same direction range’.

2) Within the same direction range, the average direction of edges is calculated. Since the most inner vertex within the range is in the most right side towards the average direction, the inner product between unit vector orthogonal to the average direction and edge vector is calculated, deciding the most inner vertex. That leads to the most inner line with the average direction in each range.
3) The intersections between the most inner lines in each range are calculated. The set of these intersections will form core polygon.

The algorithm for filtering out noise edge is as follows: In Fig.4, vertices (v_i) of a polygon are numbered clockwise. An edge which connects vertices v_i and v_j is denoted by e_{ij}. For example, e_{23} is the noise edge that is between long edges (e_{12}, e_{34}) of almost the same direction.

The GIS module tries to find out the lower limit of the same direction range (L_range) and the upper limit of the same direction range (U_range). A set of successive edges, (e_{12}, e_{23}, e_{34}, e_{45}, e_{56}) is the same direction range where normal edges of almost the same direction are in sequence. Since this is the first range, L_range is v_1, U_range is v_6 and range_counter=1. In this range, v_4 is the most inner vertex because it is in the most right side towards the average direction calculated. When the module finds out that e_{67} is a normal edge with a different direction, L_range, U_range and range_counter are updated. A set of successive edges, (e_{67}, e_{78},… , e_{16_17}) is the same direction range that starts at v_6 and ends at v_{17}. In this range, v_7 is the most inner vertex. Then, the most inner line is determined. The intersection P2 between the line (range_counter: 1) and the line (range_counter: 2) is calculated. Thus, the core polygon made up of P1 to P6 is formed.

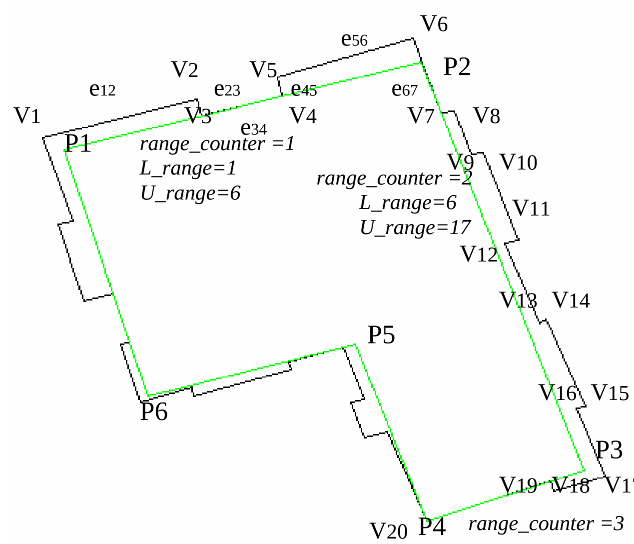

Fig.4 Short edge filtering of building polygon and forming of core polygon

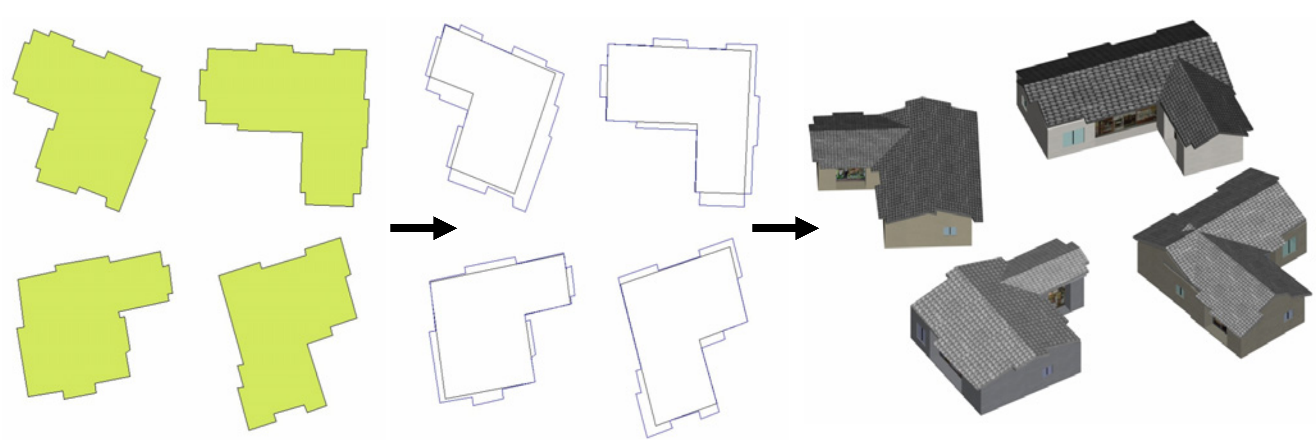

Fig.5 Building polygons that contain noise edges (Left). Core polygons that are filtered out noise edges (Center). Automatically generated 3-D house model (Right). 3-D roof model is generated from original building polygon and 3-D house model from core polygon

The GIS application stores and administrates building polygons with noise edges. The GIS module executes filtering out noise edges and the CG module receives these pre-processed data. Based on the position of vertices of the original building polygon and core polygon and attribute data, the CG module generates 3-D roof models from the original building polygon and 3-D house body models from the core polygon. The process is shown in Fig.5.

3.2 Proposed Polygon Representation

After filtering out the noise edges of a building polygon, the number of vertices of the polygon are reduced. For most of the building polygons filtered, the angles of vertices of a polygon are almost 90 or 270 degrees and the number of vertices is small. Such a building polygon can be replaced by a combination of rectangles. When following edges of a polygon clockwise, an edge turns to the right or to the left by 90 degrees. So, it is possible to assume that a building polygon with right angles only can be expressed as a set of its edges' turning direction. The polygon with 10 vertices (10 vertices polygon) shown in Fig.6 is expressed as a set of its edges' turning direction, i.e., RLRRLRRRLR where R and L mean a change of an edge's direction to the right and to the left, respectively.

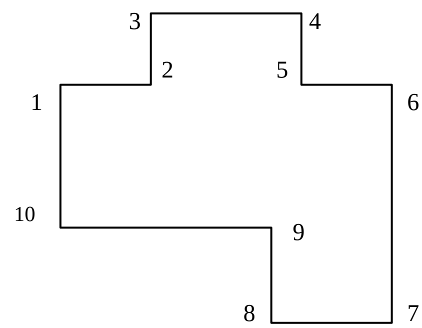

Fig.6 Building polygon is expressed as a set of its edges' turning direction ; RLRRLRRRLR

The number of shapes that a polygon can take depends on the number of vertices that a polygon has. By the formula of circular permutation, we can calculate the number of the shape patterns that a polygon may take. In the case of a 6 vertices polygon, the edges' direction set of the polygon is LRRRRR. Since the left turn edge appears only once in 6 vertices polygon, the shape pattern is unique, that is, L-shape. This L-shaped polygon is the basic polygon into which a polygon with more vertices is divided.

In the case of an 8 vertices polygon, the following four kinds of polygon shape pattern are possible: LLRRRRRR, LRLRRRRR, LRRLRRRR, LRRRLRRR. Four cases can be calculated by the following formula of circular permutation.

(1) **8 vertices polygon**:

No reiteration pattern

$$\left(\frac{8!}{6!2!} - \frac{4!}{3!1!}\right) \div 8 = 3$$

4 reiteration patterns

$$\frac{4!}{3!1!} \div 4 = 1$$

Total number of cases: 3 + 1 = 4 cases

(2) **10 vertices polygon**:

$$\frac{10!}{7!3!} \times \frac{1}{10} = 12 \text{ cases}$$

(3) **12 vertices polygon**:

3 reiteration patterns

$$\frac{3!}{2!1!} \div 3 = 1$$

6 reiteration patterns

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

No reiteration pattern

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

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

As for 10 and 12 vertices polygon, the number of shape pattern is also calculated by the same formula. In proportion to the number of vertices, the number of shape pattern will increase by geometric progression.

The advantage of this RL expression is as follows.

1) The shape of a polygon with right angles only can be described as a simple RL expression. That means a RL expression specifies the shape of a polygon.

2) This expression decides from which vertex a dividing line is drawn. From ‘L’ vertex, a dividing line is drawn so that a polygon with more than 8 vertices is broken down into rectangles and a L-shaped polygon. This is explained in the next section.

3.3 How to Divide A Polygon

A RL expression is useful in deciding the vertex from which a dividing line is drawn. In order to break down a polygon with more than 8 vertices into rectangles and a L-shaped polygon, a dividing line is drawn from ‘L’ vertex. As for a polygon consisting of more than 8 vertices, the polygon will be divided into a center area and attached branches. The polygon with right angles only is supposed to be expressed as a dataset of turning direction of its edges. In case that a dataset take L (Left turn) after or before consecutive R (Right turn), we assume this pattern as a branch. In other words, we take notice of the vertex that turns conversely. From this vertex, the dividing line is drawn. For example, ‘*RRL*’ pattern is recognized as a branch. From ‘L’ vertex, a dividing line is drawn to the backward direction in terms of vertices numbered clockwise. Also, ‘*LRR*’ pattern is recognized as a branch. From ‘L’ vertex, a dividing line is drawn to the forward. These dividing lines are sure to have intersections with one of the ‘RR’ edges, resulting in dividing into rectangles and the rest of a

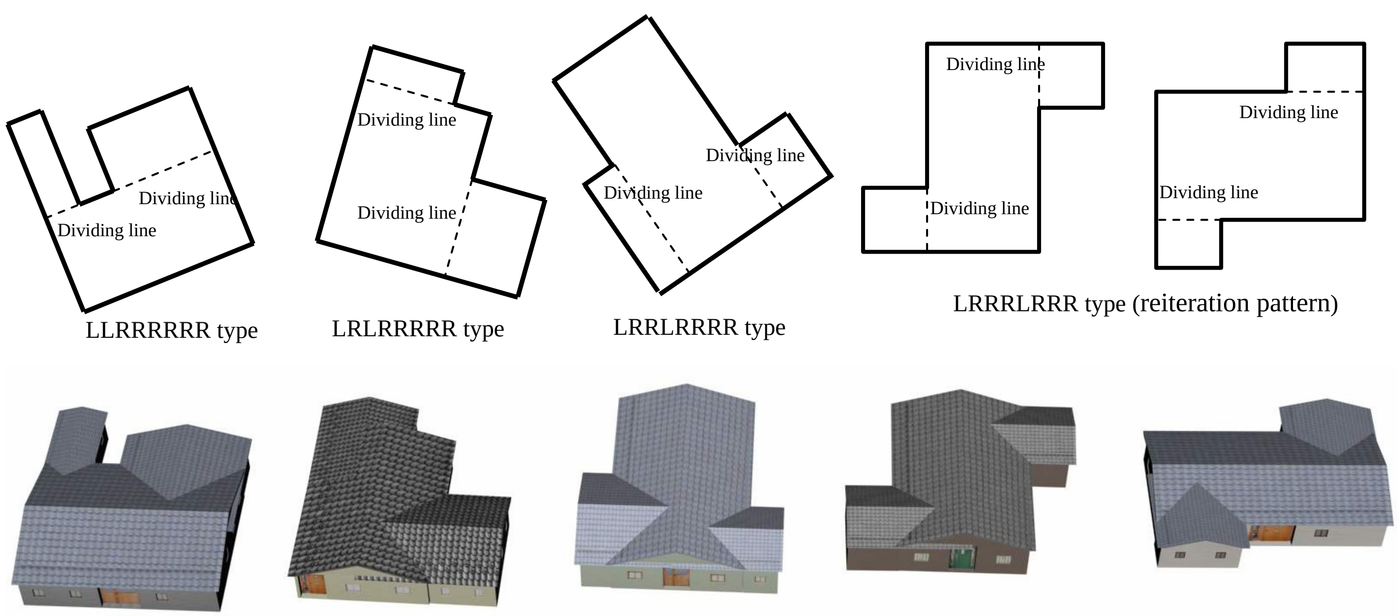

Fig.7 8 vertices polygon divided into a central area and branches (upper)
3-D building model automatically generated from the divided polygon (below)

polygon as branches and a center area.

At first, we have applied this dividing algorithm to an 8 vertices polygon. Since an 8 vertices polygon takes four types of shape pattern according to RL expression, the algorithm has been applied to four types respectively. Fig.7 shows three types that have no reiteration pattern. In these types, the module finds out a 'L' vertex after the 'RR' vertices. From this 'L' vertex, a dividing line is drawn to the backward direction. From another 'L' vertex, a dividing line is drawn to the forward.

There is the pattern that has the reiteration pattern of 'LRRR' shown at right two polygons in Fig.7. These two patterns cannot be distinguished by a RL expression. We have to consider the length of the preceding edges of the 'L' vertex. These two patterns are distinguished by the sum of the length of two edges preceding 'L' vertex. From the two 'L' vertices, dividing lines are drawn to the same direction. After dividing into branches and the rest of a polygon, the coordinates of 6 vertices of the polygon are provided as a set of coordinates of 6 vertices polygon. The algorithm for assigning to L-shaped polygon is applied to this set of coordinates.

Fig.8 shows that the dividing algorithm mentioned is applied to 8 vertices polygons. The houses in the first and second columns from the left belong to 'LRRRLRRR'. The houses in the third column belong to 'LRLRRRRR'. The houses in the fourth column belong to 'LLRRRRRR'. The houses in the right end column belong to 'LRRLRRRR'. The houses in the first row from the bottom have a gable roof. The houses in the second row have a hipped roof. The houses in the third row have a gable roof whose width is longer than roof length.

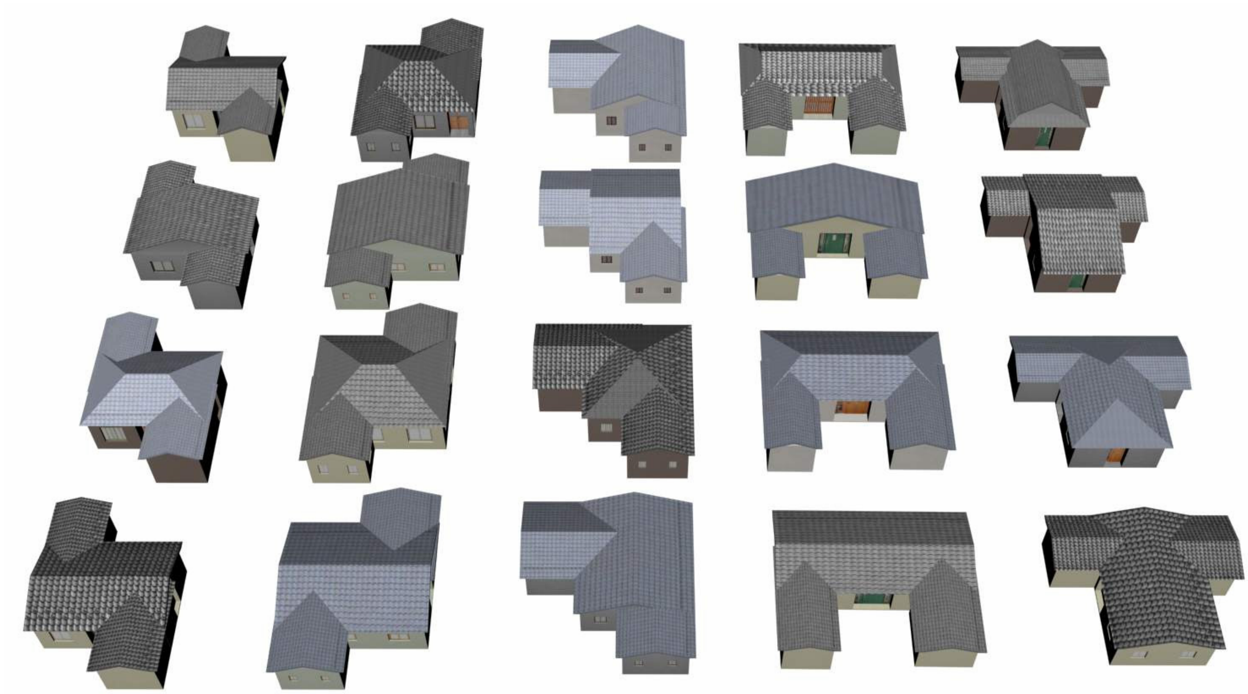

Fig.8 3-D building models with roofs generated from 8 vertices polygons

4. APPLICATION OF THE SYSTEM AND CONCLUSION

A 3-D urban model reflecting a real 3-D world offers us an important tool to display the results of a simulation of urban planning, disaster prevention and public works projects. For residents, citizens or even students as well as the specialist of urban planning, a 3-D urban model is quite effective in understanding what will be built, what image of the town will be or what if this alternative plan is realized. This model can act as a simulator to realize alternative ideas of urban planning virtually or to examine a current zoning system, building regulations and building agreements.

Here is an example of a proposed 3-D urban model produced by the automatic generation system. Fig.9 shows an aerial photo of the center area of Ogaki city where is declining because of motorization. There is a castle, but it is surrounded by buildings and is not outstanding. Fig.10 shows a 3-D urban model of the center area of Ogaki city at present. Fig.11 and Fig.12 show proposed 3-D urban models where the castle is surrounded by 2-story houses with roofs so as to be seen from the area around the castle.

Fig.9 Aerial photo of Ogaki castle, Ogaki city, Japan (Chunichi press, 2000)

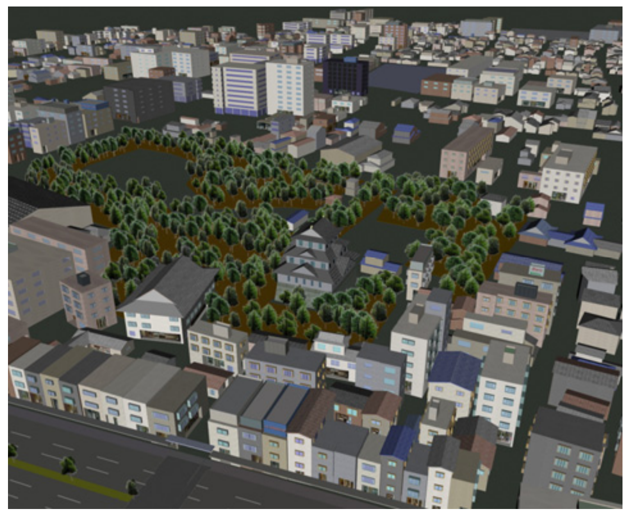

Fig.10 Automatic generated 3-D Urban Model simulating the area around the Ogaki castle

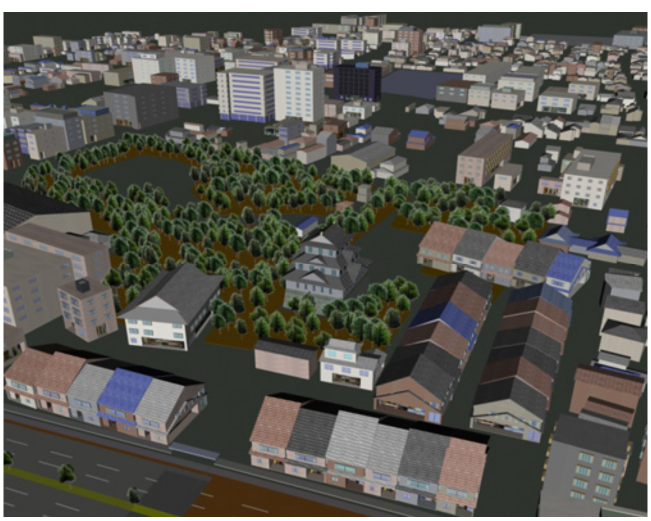

Fig.11 Proposed 3-D Urban Model (Ogaki castle surrounded by 2-story houses so as to be seen from the area around the castle)

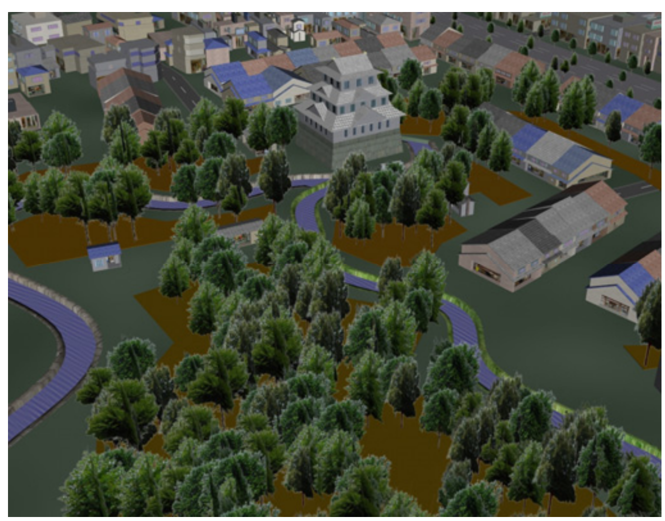

Fig.12 Proposed 3-D Urban Model (The river that restores its natural meander plays an important role for the revitalization of a city.)

In this paper, we proposed a system that automatically generates a 3-D building model from noisy building polygons. By applying noise edge filtering algorithm and then polygon dividing algorithm, we can get core polygon with smaller vertices, which then can be divided into a basic polygon and rectangles. After breaking down into a L-shaped polygon and rectangles, the module is placing primitives i.e. building parts on these polygons to form a 3-D building model.

Future work will be directed towards the development of methods for:

1) importing elevation data from DEM or some other source of remote sensing in stead of using the number of stories inputted by consultant companies
2) dividing algorithm that will be applied to 10 or more vertices polygon, although the number of vertices of most polygons filtered on a digital map is less than 10
3) generating a building with balcony or veranda whose building body is made from a core polygon and whose balcony is from an original building polygon

REFERENCES

Aerial photo of Ogaki castle, Ogaki city (2000) in Furusato Aerial photo collection, Chunichi press.

Fischer, A., Kolbe,T. H., Lang, F. (1997) Integration of 2D and 3D Reasoning for Building Reconstruction using a Generic Hierarchical Model, *Workshop on Semantic Modeling for the Acquisition of Topographic Information from Images and Maps SMATI*, **5**/97, 1-21

Gruen, A. and Wang, X. (1998) CC Modeler: A topology generator for 3-D city models, *ISPRS J. of Photogrammetry and Remote Sensing*, **53**, 286-295.

Gruen, A. and et al. (2002) Generation and visualization of 3D-city and facility models using CyberCity Modeler, *MapAsia*, 8.

Suveg, I. and Vosselman, G. (2002) Automatic 3D Building Reconstruction. *Proceedings of SPIE*, **4661**, 59-69.

Zlatanova, S. and Heuvel van den, F.A. (2002) Knowledge-based automatic 3D line extraction from close range images, *International Archives of Photogrammetry and Remote Sensing*, **34**, 233 - 238