Automatic Generation of 3D Building Models for Sustainable Development

Kenichi Sugihara^{1*}

1 School of Information Media, Gifu Keizai University

** Corresponding Author, Email: sugihara@gifu-keizai.ac.jp*

Received 5 April, 2014; Accepted 12 September, 2014

Key words: 3D urban model, automatic generation, GIS (Geographic Information System), CG (Computer Graphics), 3D building model, straight skeleton.

Abstract: 3D city models are important in urban planning for sustainable development. Urban planners draw maps for efficient land use and a compact city. 3D city models based on these maps are quite effective in understanding what, if this alternative plan is realized, the image of a sustainable city will be. However, enormous time and labour has to be consumed to create these 3D models, using 3D modelling software such as 3ds Max or SketchUp. In order to automate the laborious steps, a GIS and CG integrated system that automatically generates 3D building models is proposed based on building polygons (building footprints) on a digital map. In either orthogonal or non-orthogonal building polygons, the new system is proposed for automatically generating 3D building models with general shaped roofs by straight skeleton computation.

1. INTRODUCTION

3D city models, such as what is shown in Figure 1, right, are important in urban planning for sustainable development; urban planners draw maps for efficient land use and a compact city. 3D city models based on these maps are quite effective in understanding what, if this alternative plan is realized, the image of a sustainable city will be. To facilitate public involvement for sustainable development, 3D models simulating the town in the future can be of great use. However, enormous time and labour has to be consumed to create these 3D models, using 3D modelling software such as 3ds Max or SketchUp. For example, when manually modelling a house with roofs by Constructive Solid Geometry (CSG), one must use the following laborious steps:

(1) *Generation of primitives of appropriate size, such as box, prism or polyhedron that will form parts of a house.* (2) *Boolean operations are applied to these primitives to form the shapes of parts of a house such as making holes in a building body for doors and windows.* (3) *Rotation of parts of a house.* (4) *Positioning of parts of a house.* (5) *Texture mapping onto these parts.*

Figure 1. Pipeline of Automatic Generation for 3D Building Models

In order to automate these laborious steps, we are proposing a GIS and CG integrated system that automatically generates 3D building models (Sugihara and Kikata, 2012) based on building polygons or building footprints on a digital map shown above in Figure 1 left, which shows most building polygons' edges meet at right angles (orthogonal polygon). A complicated orthogonal polygon can be partitioned into a set of rectangles. The proposed integrated system partitions orthogonal building polygons into a set of rectangles and places rectangular roofs and box-shaped building bodies on these rectangles. In order to partition complicated orthogonal building polygons, a useful polygon expression (RL expression: edges' Right & Left turns expression) and a partitioning scheme was proposed (Sugihara, 2009) for deciding from which vertex a dividing line (DL) is drawn.

In the digital map, however, not all building polygons are orthogonal. In either orthogonal or non-orthogonal polygons, we propose the new system for automatically generating 3D building models with general shaped roofs by straight skeleton computation defined by a continuous shrinking process. We call this kind of roof 'a straight skeleton roof'. In this paper, the algorithm for shrinking a polygon and forming a straight skeleton are clarified and we propose the new methodology for constructing roof models by assuming the third event: 'simultaneous event' in addition to 'edge event' and 'split event' and, at the end of the shrinking process, some polygons are converged to 'a line of convergence'.

2. RELATED WORK

Since 3D urban models are important information infrastructure that can be utilized in several fields, the research on the creation of 3D urban models are in full swing. Various types of technology, ranging from computer vision, computer graphics, photogrammetry, and remote sensing, have been proposed and developed for creating 3D urban models.

Using photogrammetry, Gruen et al. (1998; 2002) introduced a semi-automated topology generator for 3D building models: CC-Modeler. Feature identification and measurement with aerial stereo images is implemented in manual mode. During feature measurement, measured 3D 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 3D 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 a 2D GIS map. If the building polygon is not a rectangle then it can be divided into rectangles. A polygon 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, using photogrammetry, 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 façades such as windows, et cetera (Zlatanova and Heuvel Van Den, 2002).

On the other hand, there are some application areas such as urban planning and game industries where the immediate creation and modification of many detailed building models is requested to present the alternative 3D urban models. Procedural modelling is an effective technique to create 3D models from sets of rules such as L-systems, fractals, and generative modelling language (Parish and Müller, 2001).

Müller et al. (2006) have created an archaeological site of Pompeii and a suburbia model of Beverly Hills by using a shape grammar that provides a computational approach to the generation of designs. They import data from a GIS database and try to classify imported mass models as basic shapes in their shape vocabulary. If this is not possible, they use a general extruded footprint together with a general roof obtained by the straight skeleton computation defined by a continuous shrinking process (Aichholzer et al., 1995). However, there is no digital map description or in-depth explanation about how the skeleton is formed and applied to create roofs in their paper.

More recently, image-based capturing and rendering techniques, together with procedural modelling approaches, have been developed that allow buildings to be quickly generated and rendered realistically at interactive rates. Bekins et al. exploit building features taken from real-world capture scenes. Their interactive system subdivides and groups the features into feature regions that can be rearranged to texture a new model in the style of the original. The redundancy found in architecture is used to derive procedural rules describing the organization of the original building, which can then be used to automate the subdivision and texturing of a new building. This redundancy can also be used to automatically fill occluded and poorly sampled areas of the image set.

Aliaga et al. extend the technique to inverse procedural modelling of buildings and they describe how to use an extracted repertoire of building grammars to facilitate the visualization and modification of architectural structures. They present an interactive system that enables both creating new buildings in the style of others and modifying existing buildings in a quick manner.

Vanega et al. interactively reconstruct 3D building models with the grammar for representing changes in building geometry that approximately

follow the Manhattan-world (MW) assumption which states there is a predominance of three mutually orthogonal directions in the scene. They say automatic approaches using laser-scans or LIDAR data, combined with aerial imagery or ground-level images, suffering from one or all of low-resolution sampling, robustness, and missing surfaces. One way to improve quality or automation is to incorporate assumptions about the buildings such as MW assumption. However, there are lots of buildings that have cylindrical or general curved surfaces, based on non-orthogonal building polygons.

By these means of interactive modelling, 3D building models with plausible detailed façades can be achieved. However, the limitation of this modelling is the large amount of user interaction involved ([Nianjuan and Loong-Fah, 2009](#)). When creating 3D urban models for urban planning or facilitating public involvement, 3D urban models should cover lots of the involved citizens' and stakeholders' buildings. This means that it will take an enormous amount of time and labour to model a 3D urban model with hundreds or thousands of buildings.

Thus, a GIS and CG integrated system that automatically generates 3D urban models immediately is proposed, and the generated 3D building models that constitute 3D urban models are approximate geometric 3D building models that citizens and stakeholders can recognize as their future residence or real-world buildings.

3. PIPELINE OF AUTOMATIC GENERATION

As the pipeline of automatic generation is shown in Figure 1, the source of a 3D urban model is a digital residential map that contains building polygons. The digital maps are stored and administrated by GIS application (ArcGIS, ESRI Inc.). The maps are then pre-processed in the GIS module, and the CG module finally generates the 3D urban model.

To streamline the building generation process, the knowledge-based system was proposed for generating 3D models by linking the building polygons to information from domain specific knowledge in GIS maps: attributes data such as the number of storeys and the type of roof.

Pre-processing in the GIS module includes the procedures as follows: (1) *Filter out an unnecessary vertex whose internal angle is almost 180 degrees.* (2) *Partition or separate orthogonal building polygons into sets of rectangles.* (3) *Generate inside contours by straight skeleton computation for placing doors, windows, fences and shop façades which are setback from the original building polygon.* (4) *Form the straight skeleton for the general shaped roof.* (5) *Rectify the shape of the polygons so that there are no gaps or overlaps between geometric primitives such as rectangles.* (6) *Export the coordinates of polygons' vertices, the length, width and height of the partitioned rectangle, and attributes of buildings.*

The attributes of buildings, shown in Figure 1, left below, consist of the number of storeys, the image code of the roof, wall and the type of roof (flat, gable roof, hipped roof, oblong gable roof, gambrel roof, mansard roof, temple roof, and so forth). The GIS module has been developed using 2D GIS software components (MapObjects, ESRI).

As shown in Figure 1, the CG module receives the pre-processed data that the GIS module exports, generating 3D building models. In the GIS module, the system measures the length and gradient of the edges of the

partitioned rectangle. The CG module generates a box of the length and width, as measured in the GIS module.

In the case of modelling a building with roofs, the CG module follows these steps: (1) *Generate primitives of appropriate size, such as boxes, prisms or polyhedra that will form the various parts of the house.* (2) *Boolean operations applied to these primitives to form the shapes of parts of the house, for example, making holes in a building body for doors and windows, making trapezoidal roof boards for a hipped roof and a temple roof.* (3) *Rotate parts of the house according to the gradient of the partitioned rectangle. The roof boards, in addition, are rotated so as to align them according to the parameter: the slopes of the roofs.* (4) *Place parts of the house, depending on the position of the rectangle partitioned.* (5) *Texture mapping onto these parts according to the attribute received.* (6) *Copy the 2nd floor to form the 3rd floor or more in case of buildings higher than 3 storeys.*

The length and width of a box as a housing body are decided by the rectangle partitioned or separated from a building polygon in the GIS module. Also, the length of a thin box as a roof board is decided by the rectangle partitioned while the width of a roof board is decided by the slope of the roof given as a parameter. The CG module has been developed using Maxscript that controls 3D CG software (3ds MAX, Autodesk Inc).

4. STRAIGHT SKELETON COMPUTATION FOR ROOF GENERATION

Aichholzer et al. (1995) introduced the straight skeleton defined as the union of the pieces of angular bisectors traced out by polygon vertices during a continuous shrinking process in which edges of the polygon move inward, parallel to themselves at a constant speed. The straight skeleton is unexpectedly applied to constructing general shaped roofs based on any simple building polygon, regardless of their being rectilinear or not.

As in the shrinking process shown in Figure 2 on the following page, each vertex of the polygon moves along the angular bisector of its incident edges. This situation continues until the boundary changes topologically. According to Aichholzer et al. (1995), there are two possible types of changes:

(1) **Edge event:** *An edge shrinks to zero, making its neighbouring edges adjacent now.*

(2) **Split event:** *An edge is split, i.e., a reflex vertex runs into this edge, thus splitting the whole polygon. New adjacencies occur between the split edge and each of the two edges incident to the reflex vertex.*

A reflex vertex is a vertex whose internal angle is greater than 180 degrees. All edge lengths of the polygon do not always decrease during the shrinking process. Some edge lengths of a concave polygon will increase. For example, as shown by 'ed1' and 'ed2' in Figure 2(a), the edges incident to a reflex vertex will grow in length. If the sum of the internal angles of two vertices incident to an edge is more than 360 degrees, then the length of the edge increases, otherwise the edge will be shrunk to a point (node). This shrinking procedure is uniquely determined by the distance d_{shri} between the two edges from before and after the shrinking procedure.

The distance e_d_{shri} is the d_{shri} when an edge event happens in the shrinking process. e_d_{shri} for the edge (ed_i) is calculated as follows:

$$\mathbf{e_d}_{\text{shri}} = \frac{L_i}{(\cot(0.5 * \theta_i) + \cot(0.5 * \theta_{i+1}))}$$

where L_i is the length of ed_i , and θ_i & θ_{i+1} are the internal angles of vertices incident to ed_i .

When $0.5 * \theta_i + 0.5 * \theta_{i+1} < 180$ degrees, i.e., the sum of the internal angles of two vertices incident to an edge is less than 360 degrees, an edge event may happen unless the edge is intersected by an angular bisector from a reflex vertex and a split event happens.

Figure 2 from (a) to (c) shows a shrinking process for a non-orthogonal concave polygon, the polygon just before a split event, and the polygon being split into two polygons after a split event happens. Figure 2(d) shows a set of polygons shrinking at the constant interval and nodes by an edge event and a split event, and nodes by a collapse of a triangle into a point.

Figure 2(e) shows the straight skeleton defined as the pieces of angular bisectors traced out by polygon vertices. Figure 2(f) shows the roof model automatically generated. Since the straight skeleton partitions the interior of a polygon with n vertices (n -gon) into n monotone polygons, each roof board that constitutes the roof model is formed based on these partitioned ‘interior monotone polygons’.

Figure 2. Shrinking process and a straight skeleton, a roof model automatically generated

Figure 3, below, shows how a split event happens, and how the position of the node having arisen by the split event is calculated. The position of the node is given by the intersection of two angular bisectors: one from the reflex vertex and the other bisector situated between the intersected edge and one of two edges incident to the reflex vertex.

Figure 3. How a split event happens, and how the position of the node is calculated

For some polygons, such as those shown in Figure 4, the event may differ from the two events mentioned. In this research, it is proposed to add this third event, 'the simultaneous one', in which the above two events happen simultaneously. This event happens at an orthogonal part of the polygon as shown in Figure 4, in which event a reflex vertex runs into the edge, but the other split polygon is collapsed into a node since an edge event happens in the split polygon at the same time. Figure 4 shows a shrinking process of an orthogonal polygon. In the process, as shown in Figures 4(b) and (c), the system detects 'the simultaneous event' by checking if pt_{ia} (vertex) is on ed_{ib} (edge) or pt_{ib} is on ed_{ia} where pt_{ia} and pt_{ib} are the vertices next to two vertices coherent by the edge event, and ed_{ia} and ed_{ib} are the edges adjacent to these two coherent vertices.

Aichholzer et al. (1995) demonstrated three edge events let a triangle collapse to a point in the last stage of each split polygon as shown in Figure 2(d). In this paper, it is proposed to add the case in which two edge events let a rectangle collapse to a line segment ('a line of convergence') in the last stage, a rectangle whose opposite sides are the same and the shortest $e_{d_{shri}}$.

Since a line segment does not have an area, it is not shrunk anymore. The central area of an orthogonal polygon, shown in Figure 4(d), shows a line of convergence to which the shrinking polygon (rectangle) are converged.

After calculating $e_{d_{shri}}$ for all edges and finding the shortest of them, the shrinking process may proceed until d_{shri} reaches the shortest $e_{d_{shri}}$ found. In the process, a split event may happen and the polygon will be divided. So, during shrinking to the shortest $e_{d_{shri}}$, our system checks if a line segment of an angular bisector from a reflex vertex intersects another edge of the polygon or not. If an edge is found intersected, the system calculates the position of the node by the split event as shown in Figure 3(d). However,

Figure 4. Shrinking process and a straight skeleton for simultaneous events

one edge will be intersected by several angular bisectors from several reflex vertices. Among the several reflex vertices, the reflex vertex that gives the shortest d_{shri} will be selected for calculating the position.

After any type of event happens and the polygon changes topologically, we are left with one or more new polygons which are shrunk recursively if they have a non-zero area. At that moment, the system recalculates the length of each edge and internal angle of each vertex in order to find the shortest d_{shri} for the next events.

5. APPLICATION

Here are the examples of 3D building models automatically generated by the integrated system. Figure 5 shows the examples of 3D building models automatically generated by straight skeleton computation from non-orthogonal building polygons. To ease the discussion, Aichholzer et al. exclude degeneracies caused by special shapes of polygon, for example, a regular polygon.

In this paper, we deal with the degenerate cases in which more than three edges are shrunk to a point. Ideally, simultaneous n edge events cause a regular n -gon to collapse to a point but it is difficult to draw such a perfect regular n -gon. Accordingly, the system rectifies the shape of the regular n -gon so as to let n edge events occur at the same time. Figure 5, centre, shows the 3D dodecagon building model automatically generated based on the degeneracy of 12 edges being shrunk to only one node.

Figure 5. Non-orthogonal building footprints and 3D building models automatically generated by straight skeleton computation.

Based on digital maps, we propose the GIS and CG integrated system for automatically generating 3D building models. To generate real-world 3D urban models, the 3D shapes and material attributes of buildings and other objects need to be reconstructed. In the reconstructing process, the image data will be acquired by taking photographs of the objects in the city. But, when thinking of the future layout of the compact city and sustainable development, we cannot take photos of the future of the city and planning roads. Usually and traditionally, urban planners design the town layout for the future by drawing the maps, using GIS or CAD packages.

There may be several plans (digital maps) for urban planning. If the integrated system immediately converts these maps into 3D city models, the system surely supports the researchers and urban planners investigating the alternative idea.

Figures 6 and 7 show the digital map: the city layout proposed by an urban designer and a 3D urban model automatically generated. The area is called Meijo district in Nagoya City with a population of 2 million, the 4th largest city in Japan. This area is in almost the central area of Nagoya City and is located near Nagoya Castle. There are three- or four-storey apartment houses for government officials and citizens run by the municipal government. The area as a whole was selected as one of the areas for a city revitalization project by the Japanese government.

Figure 6 shows the alternative plan proposed by an urban designer, which is planned as low density in the middle area for condominiums with green spaces and urban facilities, and high density in both side areas for official housing and public housing. In the proposed alternative plan, as shown in Figure 7, terrace houses sharing green spaces in large courtyards are laid out for viewing Nagoya castle. The courtyards are enclosed by apartments, providing a shared park-like space for residents. Figure 7 also shows buildings are increasing in height away from the castle for each of the building's top floor residents to see Nagoya castle.

These alternative plans are created by merging generated 3D building models into three-dimensional maps (MAP CUBE) provided by Pasco Co., Ltd, an aerial survey company in Japan.

6. CONCLUSION

Figure 6. Proposed layout for the town and an automatically generated 3D urban model: low density in the middle area for condominiums with green spaces and urban facilities, and high density in both side areas for official housing and public housing

Figure 7. Proposed layout for the town and an automatically generated 3D urban model: terrace houses sharing green spaces in large courtyard, laid out for viewing Nagoya castle

For everyone, a 3D urban model is effective in understanding what, if this alternative plan is realized, the image of a sustainable city will be. Traditionally, urban planners design the city layout for the future by drawing building polygons on a digital map. This integrated system, depending on the building polygons, generates a 3D urban model so instantly that it meets the urgent demand to realize another alternative urban plan.

If given digital maps with attributes being inputted, as shown in the ‘Application’ section, the system automatically generates two hundred 3D building models within less than 20 minutes by a personal computer with an Intel Core i7 CPU 3.5GHz processor. In either orthogonal or non-orthogonal polygons, we propose the new system for automatically generating general shaped roof models by straight skeleton computation. In this paper, we propose the new methodology for constructing roof models by assuming the third event: simultaneous events in addition to two events and, at the end of the shrinking process, some rectangles converge into a line of convergence. Thus, the proposed integrated system succeeds in automatically generating alternative city plans.

The limitation of the system is that automatic generation is executed based only on ground plans or top views. There are some complicated shapes

of buildings whose outlines are curved or even crooked. To create these curved buildings, the system needs side views and front views for curved outline information. Future work will be directed towards the development of methods for the automatic generation algorithm to model curved buildings by using side views and front views.

REFERENCES

- Aichholzer, O., Aurenhammer, F., et al. (1995). "A novel type of skeleton for polygons", *Journal of Universal Computer Science*, 1 (12), 752–761.
- Carlos, V.A., Daniel, A.G., et al. (2010). "Building reconstruction using Manhattan-world grammars", *Computer Vision and Pattern Recognition (CVPR)*, 2010 IEEE Conference, 358- 365.
- Daniel, B.R. and Daniel, A.G. (2005). "Build-by-number: rearranging the real world to visualize novel architectural spaces", *Visualization*, 2005. VIS 05. IEEE, 143-150.
- Daniel, A.G. Paul, R.A., et al. (2007). "Style Grammars for interactive Visualization of Architecture", *Visualization and Computer Graphics*, IEEE Transactions , 13, 786-797.
- Gruen, A. and Wang, X. (1998). "CC-Modeler: A topology generator for 3-D city models", *ISPRS Journal of Photogrammetry & Remote Sensing*, 53(5), 286-295.
- Gruen, A. et al. (2002). "Generation and visualization of 3D-city and facility models using CyberCity Modeler", *MapAsia*, 8, CD-ROM.
- Müller, P., Wonka, P. et al. (2006). "Procedural modeling of buildings", *ACM Transactions on Graphics*, 25, (3), 614–623.
- Nianjuan, J.P.T. and Loong-Fah, C. (2009). "Symmetric architecture modeling with a single image", *ACM Transactions on Graphics - TOG*, 28(5).
- Parish, I.H.Y. and Müller, P. (2001). "Procedural modeling of cities", *Proceedings of ACM SIGGRAPH 2001*, ACM Press, E. Fiume, Ed., New York, 301-308.
- Sugihara, K. (2009). "Automatic generation of 3d building models with various shapes of roofs", *ACM SIGGRAPH ASIA 2009*, Sketches DOI: 10.1145/1667146.1667181.
- Sugihara, K. and Kikata, J. (2012). "Automatic Generation of 3D Building Models from Complicated Building Polygons", *Journal of Computing in Civil Engineering, ASCE* (American Society of Civil Engineers), DOI: 10.1061/(ASCE)CP.1943-5487.0000192.
- 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.