Automatic Generation of 3D Building Models with General Shaped Roofs by Straight Skeleton Computation

Kenichi Sugihara[1] and Zhen-jiang Shen[2]

[1] Gifu Keizai University, Ogaki-City, Gifu-Pref., Japan
[2] Kanazawa University, Kanazawa City, Japan
sugihara@gifu-keizai.ac.jp; shenzhe@t.kanazawa-u.ac.jp

Abstract

3D city models are important in several fields, such as urban planning and gaming industries. However, enormous time and labor has to be consumed to create these 3D models, using a 3D modeling software such as 3ds Max or SketchUp. In order to automate laborious steps, we are proposing a GIS and CG integrated system for automatically generating 3D building models, based on building polygons (building footprints) on digital maps. Digital maps shows most building polygons' edges meet at right angles (orthogonal polygon). 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. 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 and, at the end of the shrinking process, some polygons are converged to a line.

1. Introduction

3D urban models, such as the one shown in Fig.1, are important in urban planning and in facilitating public involvement. To facilitate public involvement, 3D models simulating a real or near future cities by a 3D CG can be of great use. However, enormous time and labour has to be consumed to create these 3D models, using 3D modeling softwares such as 3ds Max or SketchUp. For example, when manually modeling 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.

In order to automate these laborious steps, we are proposing a GIS and CG integrated system that automatically generates 3D building models, based on building polygons or building footprints on a digital map shown in Fig.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 for deciding from which vertex a dividing line (DL) is drawn.

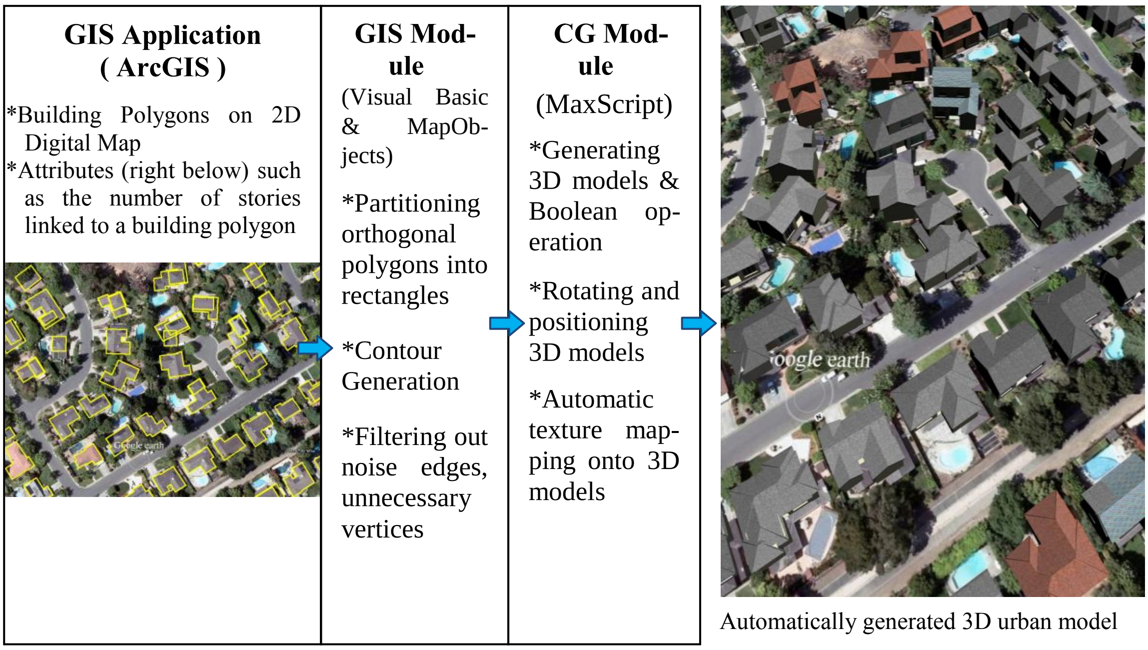


Fig.1. Pipeline of Automatic Generation for 3D Building Models

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 (Aichholzer 1995). 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 researches on creations of 3D urban models are in full swing. Various types of technologies, ranging from computer vision, computer graphics, photogrammetry, and remote sensing, have been proposed and developed for creating 3D urban models.

Procedural modeling is an effective technique to create 3D models from sets of rules such as L-systems, fractals, and generative modeling language (Parish et al. 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 a straight skeleton computation defined by a continuous shrinking process (Aichholzer et al. 1995).

The straight skeleton is the set of lines traced out by the moving vertices in this shrinking process and can be used as the set of ridge lines of a building roof (Aichholzer et al. 1996). However, the roofs created by the straight skeleton are limited to hipped roofs or gable roofs with their ridges parallel to long edges of the rectangle into which a building polygon is partitioned. There are many roofs whose ridges are perpendicular to a long edge of the rectangle, and these roofs cannot be created by the straight skeleton. Since the straight skeleton treats a building polygon as a whole, it forms a seamless roof so that it cannot place roofs independently on partitioned polygons.

To create the various shapes of 3D roofs, we proposed the system that has an option to select partitioning scheme; ‘separation prioritizing’ or ‘shorter DL (dividing line) prioritizing’, which is decided by attribute data manually stored beforehand. The proposed system also tries to select a suitable DL for partitioning or a separation, depending on the RL expression of a polygon, the length of DLs and the edges of a polygon. But, the system was not able to create 3D building models with roofs from non-orthogonal building polygons. In this paper, we propose the system for creating 3D building models with general shaped roofs by straight skeleton.

More recently, image-based capturing and rendering techniques, together with procedural modeling approaches, have been developed that allow buildings to be quickly generated and rendered realistically at interactive rates. Bekins et al. (2005) 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. (2007) extend the technique to inverse procedural modeling 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. (2010) 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 interactive modeling, 3D building model with plausible detailed façade can be achieved. However, the limitation of these modeling is the large amount of user interaction involved (Nianjuan et al. 2009). When creating 3D urban model for urban planning or facilitating public involvement, 3D urban models should cover lots of citizens' and stakeholders' buildings involved. This means that it will take an enormous time and labour to model 3D urban model with hundreds or thousands of building. Thus, the 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 stakeholder can recognize as their future house or real-world buildings.

3. Proposed system for automatic building generation

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

The GIS module 'pre-processes' building polygons on the digital map. 'pre-process' includes the functions as follows; **(1)** filtering out an unnecessary vertex whose internal angle is almost 180 degrees **(2)** partitioning orthogonal building polygons into sets of rectangles **(3)** generating inside contours by straight skeleton computation for positioning doors, windows, fences and shop façades which are setback from the original building polygon **(4)** forming the straight skeleton for the general shaped roof **(5)** exporting the coordinates of polygons' vertices and attributes of buildings.

The attributes of buildings consist of the number of storeys, the image code of 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).

The CG module receives the pre-processed data that the GIS module exports, generating 3D building models. CG module has been developed using Maxscript that controls 3D CG software (3ds MAX, Autodesk Inc). In case of modeling a building with roofs, the CG module follows these steps:
(1) generation of primitives of appropriate size, such as boxes, prisms or polyhedra that will form the various parts of the house **(2)** Boolean operations are applied to these primitives to form the shapes of parts of the house, for examples, making holes in a building body for doors and windows, making trapezoidal roof boards for a hipped roof and a temple roof **(3)** rotation of parts of the house **(4)** positioning of parts of the house **(5)** texture mapping onto these parts according to the attribute received.

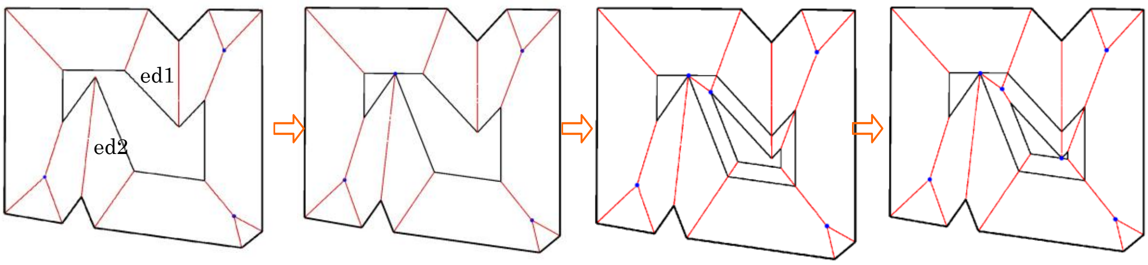
4. Straight skeleton computation for roof generation

As shown Fig.2 (f), Aichholzer et al.(1995,1996) 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. In this paper, 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'.

According to Aichholzer et al.(1995,1996), in shrinking process as shown in Fig.2, each vertex of the polygon moves along the angular bisector of its incident edges. This situation continues until the boundary change topologically. There are two possible types of changes:
(1) **Edge event**: *An edge shrinks to zero, making its neighboring 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.

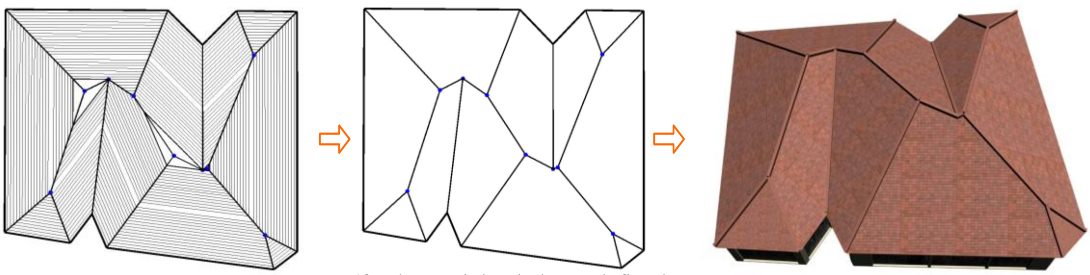


(a) Shrinking polygon just before a split event

(b) Split event happens and the polygon is split into two polygons.

(c) Shrinking polygon split just before another split event

(d) Split event happens and the polygon is split into two triangles.



(e) Polygons shrinking at a constant interval: nodes by an edge event: nodes by a split event: nodes by a collapse of a triangle to a point.

(f) The straight skeleton defined as the union of the pieces of angular bisectors traced out by polygon vertices during the shrinking process.

(g) A roof model automatically generated: each roof board is based on an 'interior monotone polygon' partitioned by straight skeleton.

Fig.2. Shrinking process and a straight skeleton, a roof model automatically gener-

For example, as shown by 'ed1' and 'ed2' in Fig.2 (a), the edges incident to one reflex vertex 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. During the shrinking process, when the edge is shrunk to a point, we call this event an edge event.

Shrinking procedure is uniquely determined by the distance $\mathbf{d}_{shri}$ between the two edges of before & after shrinking procedure. The distance $\mathbf{e\_d}_{shri}$ is the $\mathbf{d}_{shri}$ when an 'edge event' happens in the shrinking process. $\mathbf{e\_d}_{shri}$ for edge (ed_i) is calculated as follows:

$$e\_\mathbf{d}_{shri} = L_i / (\cot(0.5 * \theta_i) + \cot(0.5 * \theta_{i+1}))$$

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

Fig.2 from (a) to (d) show a shrinking process for a non-orthogonal concave polygon: the polygon just before a split event: the polygon being split into two polygons after the split event happens. Fig.2 (e) 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. Fig.2 (f) shows the straight skeleton defined as the pieces of angular bisectors traced out by polygon vertices during the shrinking process. Fig. 2(g) 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'.

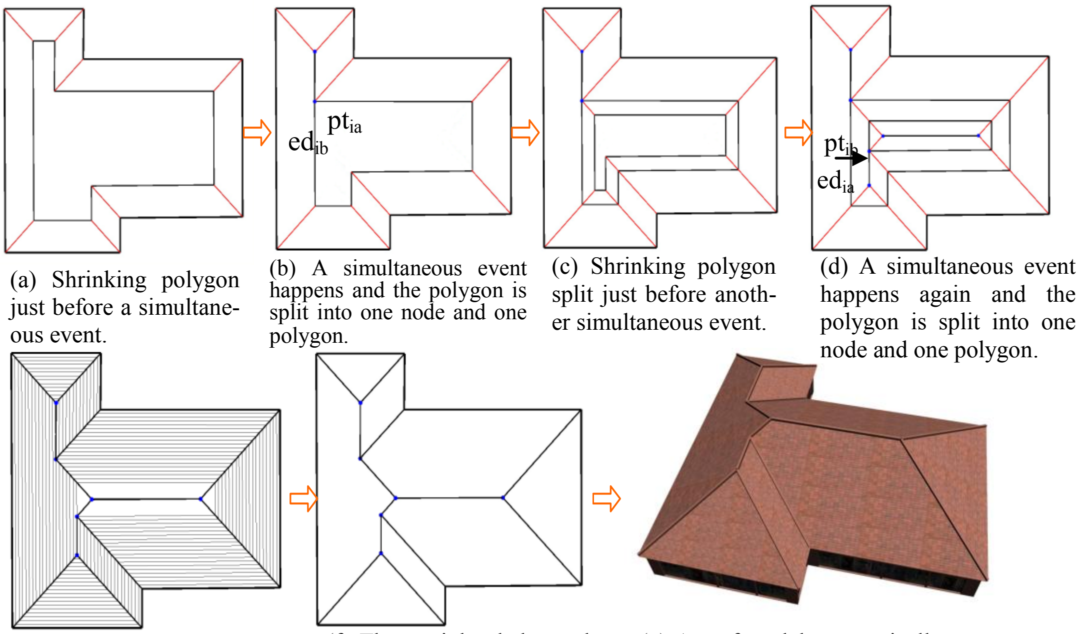
After calculating $\mathbf{e\_d}_{shri}$ for all edges and finding the shortest of them, the shrinking process may proceed until $\mathbf{d}_{shri}$ reaches the shortest $\mathbf{e\_d}_{shri}$ found. When any type of event happens and the polygon changes topologically, we are left with one or more new polygons which are shrunk recursively if they have non-zero area. At that moment, the system recalculates the length of each edge and internal angles of each vertex in order to find the shortest $\mathbf{d}_{shri}$ for next events.

For some polygons such as shown in Fig.3, the event different from above two events will happen. In our research, we propose to add the third event: 'the simultaneous one' in which above two events happen simultaneously. This event happens at an orthogonal part of the polygon as shown in Fig.3, 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.

Fig.3 shows a shrinking process of an orthogonal polygon. In the process, as shown Fig.3 (b) & (d), 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} & pt_{ib} are the vertices next to two vertices coherent by the edge event, and ed_{ia} & ed_{ib} are the edges adjacent to these two coherent vertices.

Aichholzer et al.(1996) demonstrated three edge events let a triangle collapse to a point in the last stage of each split polygon as shown in Fig.2 (e). In this paper, we propose 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 have the same and the shortest $\mathbf{e\_d}_{shri}$.

Since a line segment does not have area, it is not shrunk anymore. The central area of an orthogonal polygon in Fig.3 (e) shows a line of convergence to which the shrinking polygon (rectangle) is converged.



(a) Shrinking polygon just before a simultaneous event.

(b) A simultaneous event happens and the polygon is split into one node and one polygon.

(c) Shrinking polygon split just before another simultaneous event.

(d) A simultaneous event happens again and the polygon is split into one node and one polygon.

(e) Polygons shrinking at a constant interval: nodes by an edge event & a simultaneous event: nodes by a collapse of a rectangle to a line

(f) The straight skeleton defined as the union of the pieces of angular bisectors traced out by polygon vertices during the shrinking pro-

(g) A roof model automatically generated: each roof board is based on an 'interior monotone polygon' partitioned by straight skeleton

Fig.3. Shrinking process and a straight skeleton for simultaneous

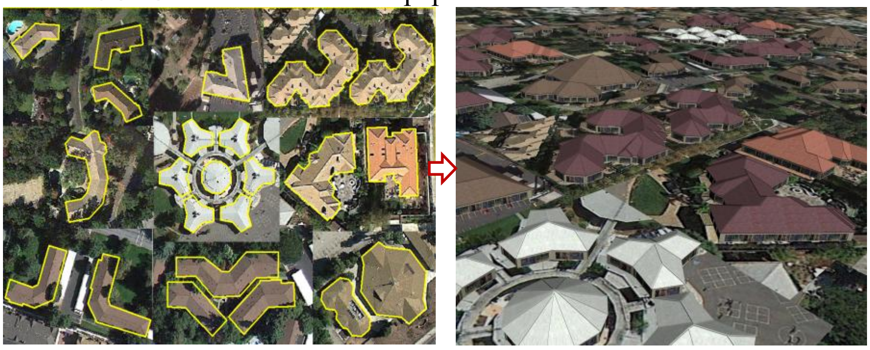


Fig. 4. Non-orthogonal building footprints and 3D building models automatically generated by straight skeleton computation

5. Application

Here are the examples of 3D building models automatically generated by the integrated system. Fig.4 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.(1996) exclude degeneracies caused by special shapes of polygon, e.g., 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 at the same time. Fig.4 center shows the 3D dodecagon building model automatically generated based on the degeneracy of 12 edges being shrunk to only one node.

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 city layout for the future or the ancient city models, we cannot take photos of the future of the city, planning road, or the cities of ancient times. 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. Similarly excavation and investigation companies also submit digital maps as the result of excavation. Usually there are several plans for restored architectural heritage in the estimation process. There are several restoration maps estimated and proposed by several archaeological researchers. 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.

Fig.5 shows the digital map: the city layout proposed by an urban designer and a 3D urban model automatically generated. In the 3D model, terrace houses sharing green spaces in large courtyard are laid out for viewing Nagoya castle. Fig.5 also shows buildings are increasing in height away from castle for each building's top floor to see Nagoya castle.

6. Conclusion

A For everyone, a 3D urban model is quite effective in understanding what if this alternative plan is realized, what image of the town were or what has been built. Traditionally, urban planners design the city layout for the future by drawing building polygons on a digital map. Depending on the building polygons, the integrated system automatically generates a 3D urban model so instantly that it meets the urgent demand to realize another alternative urban planning.

If given digital maps with attributes being inputted, as shown in 'Application' section, the system automatically generates two hundreds 3D building models within less than 30 minutes. In either orthogonal or non-orthogonal polygons, we propose the new system for automatically gener-

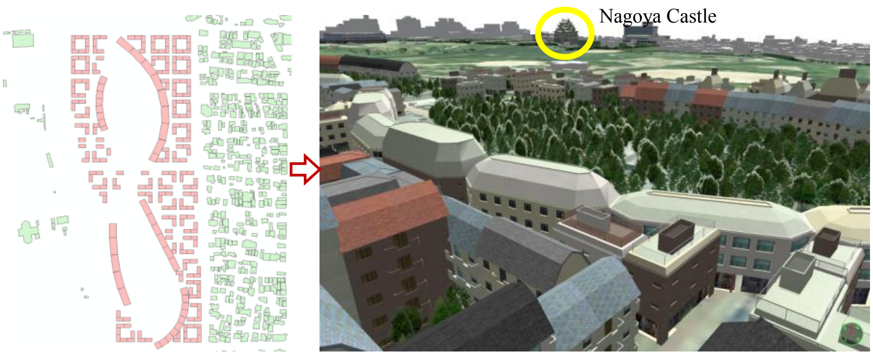


Fig.5. Proposed layout for the town and an automatically generated 3D urban model: terrace houses sharing green spaces in large courtyard, laid out for viewing Nagoya castle

ating 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 event in addition to two events and, at the end of the shrinking process, some rectangles are converged to 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 outlines 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

Aliaga G. Daniel, Rosen A. Paul ,and Bekins R. Daniel (2007). “Style Grammars for interactive Visualization of Architecture”, Visualization and Computer Graphics, IEEE Transactions on Volume:13, 786 - 797

Bekins R. Daniel ,and Aliaga G. Daniel (2005). “Build-by-number: rearranging the real world to visualize novel architectural spaces”, Visualization, 2005. VIS 05. IEEE, 143 – 150.

Nianjuan, Jiang, Ping, Tan, and Loong-Fah, Cheong (2009). “Symmetric architecture modeling with a single image”, ACM Transactions on Graphics - TOG, vol. 28, no. 5

O. Aichholzer, F. Aurenhammer ,and D. Alberts, B. Gärtner (1995). “A novel type of skeleton for polygons”, Journal of Universal Computer Science, 1 (12): 752–761.

O. Aichholzer ,and F. Aurenhammer (1996). “Straight skeletons for general polygonal figures in the plane”, Proc. 2nd Ann. Int. Conf. Computing and Combinatorics (COCOON '96): 117–126.

Pascal Mueller, Peter Wonka, Simon Haegler, Andreas Ulmer, Luc Van Gool. (2006). “Procedural modeling of buildings”, ACM Transactions on Graphics 25, 3, 614–623.

Vanegas A. Carlos, Aliaga G. Daniel ,and Beneš Bedřich (2010). “Building reconstruction using Manhattan-world grammars”, Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on: 358 - 365

Yoav I. H. Parish, and Pascal Müller (2001). “Procedural modeling of cities”, Proceedings of ACM SIGGRAPH 2001, ACM Press, E. Fiume, Ed., New York, 301–308.