Automatic Generation of 3D Building Models by Building Polygon Rectification

Kenichi SUGIHARA, Xinxin ZHOU, Zhen-jiang SHEN

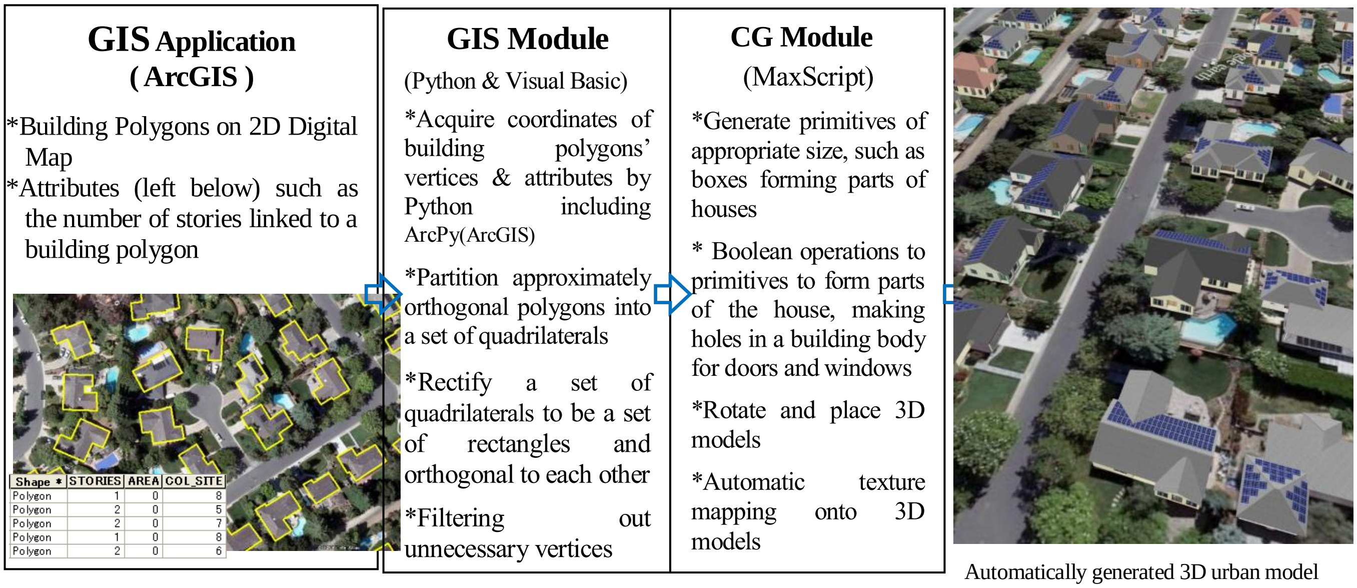
Abstract: A 3D urban model is important in urban planning. Urban planners may draw the maps for sustainable development. 3D urban models based on these maps are quite effective in understanding what if this alternative plan is realized. But, enormous time and labour has to be consumed to create these 3D models. In order to automate laborious steps, a GIS and CG integrated system is proposed for automatically generating 3D building models, based on building polygons on a digital map. In the digital map, not all building polygons are precisely orthogonal. However, creating 3D building models are expected to be orthogonal. When placing a set of boxes as building bodies for creating the buildings, there may be gaps or overlaps between these boxes if building polygons are not precisely orthogonal. In this paper, the new methodology is proposed for rectifying the shape of building polygons and generating 3D building models without any gap and overlap.

Keywords: Automatic generation, 3D urban model, Computer Graphics, 3D building model, Digital map, Urban planning, Building polygon rectification

1. Introduction

A 3D urban model, such as the one shown in Figure 1 right, is important in urban planning and gaming industries. Urban planners may draw the maps for sustainable development. 3D urban models based on these maps are quite effective in understanding what if this alternative plan is realized. However, enormous time and labour has to be consumed to create these 3D models, using 3D modeling software such as 3ds Max or SketchUp. In order to automate laborious steps, a GIS and CG integrated system was proposed for automatically generating 3D building models, based on building polygons or building footprints on a digital map shown in Figure 1 left [Sugihara 2006,2009].

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 an orthogonal polygon, 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 [Sugihara 2012]. In the digital map, however, not all building polygons are precisely orthogonal. When placing a set of boxes as building bodies for forming the buildings, there may be gaps or overlaps between these boxes if building polygons are not strictly orthogonal. In this paper, the new methodology is proposed for rectifying the shape of building polygons and constructing 3D building models without any gap and overlap.



GIS Application (ArcGIS)

*Building Polygons on 2D Digital Map
*Attributes (left below) such as the number of stories linked to a building polygon

Shape *	STORIES	AREA	COL_SITE
Polygon	1	0	8
Polygon	2	0	5
Polygon	2	0	7
Polygon	1	0	8
Polygon	2	0	6

GIS Module

(Python & Visual Basic)

*Acquire coordinates of building polygons' vertices & attributes by Python including ArcPy(ArcGIS)

*Partition approximately orthogonal polygons into a set of quadrilaterals

*Rectify a set of quadrilaterals to be a set of rectangles and orthogonal to each other

*Filtering out unnecessary vertices

CG Module

(MaxScript)

*Generate primitives of appropriate size, such as boxes forming parts of houses

* Boolean operations to primitives to form parts of the house, making holes in a building body for doors and windows

*Rotate and place 3D models

*Automatic texture mapping onto 3D models

Automatically generated 3D urban model

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

Kenichi Sugihara : Gifu Keizai University, Ogaki city, Japan
Phone: +81-584-77-3511 E-mail: sugihara@gifu-keizai.ac.jp

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 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 1995]. However, there is no digital map description and 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 modeling approaches, have been developed that allow buildings to be quickly generated and rendered realistically at interactive rates. Bekins et al. [Daniel 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.

Vanega et al. [Carlos 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.

By these interactive modeling, 3D building models with plausible detailed façade can be achieved. However, the limitation of these modeling is the large amount of user interaction involved [Nianjuan 2009]. When creating 3D urban models 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 a 3D urban model with hundreds 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 residence or real-world buildings.

3. Pipeline of automatic generation

As shown in Figure 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 shown in Figure 1 left below, consist of the number of storeys, the image code of roof, wall and the type of roof (gable roof, hipped roof, gambrel roof, mansard roof, temple roof and so forth). The maps are then preprocessed at the GIS module, and the CG module finally generates the 3D urban model.

As a GIS module, a Python program including ArcPy(ArcGIS) acquires coordinates of building polygons' vertices and attributes.
Preprocessing at 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 approximately orthogonal polygons into a set of quadrilaterals. (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) Rectify a set of quadrilaterals to be a set of rectangles and orthogonal to each other. (5) Export the coordinates of polygons' vertices, the length, width and height of the partitioned rectangle, and attributes of buildings.

The CG module receives the pre-processed data that the GIS module exports, generating 3D building models. In GIS module, the system measures the length and inclination of the edges of the partitioned rectangle. The CG module generates a box of the length and width, measured in GIS module.

In case of modeling 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 examples, 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 inclination of the partitioned rectangle. (4) Place parts of the house. (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 building higher than 3 stories.

CG module has been developed using Maxscript that controls 3D CG software (3ds MAX, Autodesk Inc).

4. Functionality of GIS Module

4.1 Polygon Expression & Partitioning Scheme

At map production companies, technicians are drawing building polygons manually with digitizers, depending on aerial photos or satellite imagery as shown in Figure 1&2. This aerial photo and digital map also show that most building polygons are approximately orthogonal polygons. An approximately orthogonal polygon can be replaced by a combination of rectangles. When the vertices of a polygon are numbered in clockwise order and edges of a polygon are followed clockwise, an edge turns to the right or to the left by 90 degrees. It is possible to assume that an orthogonal polygon can be expressed as a set of its edges' turning direction; an edge turning to the 'Right' or to the 'Left'.

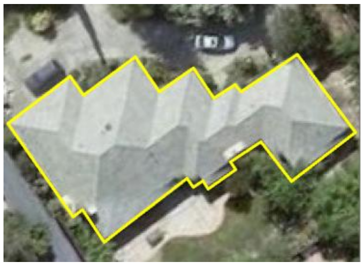


Figure 2: Building polygon on satellite image, expressed as RRLRLRLRLRRRLLRRLRLRLR

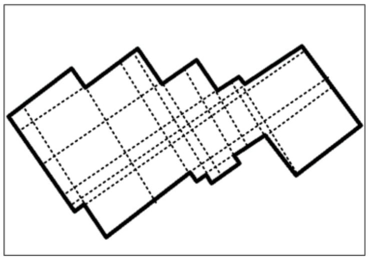


Figure 3: Two DLs from each L vertex, 9 L vertices, so 18 DLs can be drawn from L vertices

A useful polygon expression (RL expression: edges' Right & Left turns expression) is proposed for specifying the shape pattern of an orthogonal polygon [Sugihara 2005]. For example, an orthogonal polygon with 22 vertices shown in Figure 2 is expressed as a set of its edges' turning direction; RRLRLRLRLRRRLLRRLRLRLR where R and L mean a change of an edge's direction to the right and to the left, respectively.

The more vertices a polygon has, the more partitioning scheme a polygon has, since the interior angle of a 'L' vertex is 270 degrees and two DLs (dividing lines) can be drawn from a L vertex. In the polygon shown in Figure 3, there are 9 L vertices, so 18 DLs can be drawn from each L vertex as shown in dotted lines. In our proposal, [Sugihara 2012] among many possible DLs, the DL that satisfies the following conditions is selected for partitioning.

(1) *A DL that cuts off 'one rectangle'.*

(2) *A DL whose length is shorter than the width of a 'main roof' that a 'branch roof' is supposed to extend to, where a 'branch roof' is a roof that is cut off by a DL and extends to a main roof.*

How the system is finding 'branches' is as follows. The system counts the number of consecutive 'R' vertices (=n_R) between 'L' vertices. If n_R is two or more, then it can be a branch. One or two DLs can be drawn from 'L' vertex in a clockwise or counter-clockwise direction, depending on the length of the adjacent edges of 'L' vertex. How the DL is drawn (clockwise or counter-clockwise), that is, 'dividing pattern' is used for reconstructing a rectified polygon and saved at the divided rectangle.

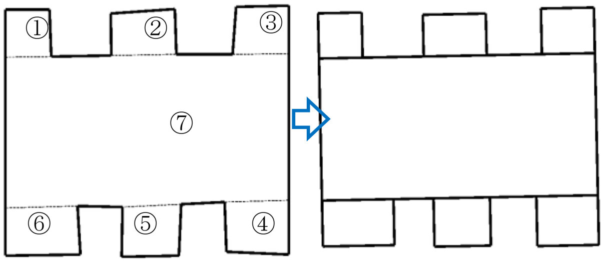
4.2 Active branch rectangle

Figure 4 shows the process of polygon partition and rectification of polygon shape, generation of 3D model. In Figure 4 (b), partitioned rectangles are numbered according to the order of partitioning. Rect (rectangle) ①& ③ are the branch rects that will extend to their main roof as shown in Figure 4 (d).

In Figure 4 (b), at the time when rect ① is divided, it is impossible for rect ① to know which rect is adjacent to and which edge of the rect is adjacent to. It is not until rect ② is divided that rect ① knows which rect is adjacent to. After polygon partitioning, rect ① will start to search for an adjacent rect. Rect ③ also has to look for 'one' adjacent rect. Although rect ③ has two adjacent rects, when rect ② is divided, it is possible for rect ③'s edge to save the information about the adjacent rect and adjacent edge, dividing pattern.

Rect ①&③ are defined as 'active rect' that will search for an adjacent rect and rectify its shape according to the orthogonality. After rectification, divided precise rectangles become orthogonal to each other as shown in Figure 4(c).

Figure 5 shows how 'branch rects' are cut off: three dividing patterns. In Figure 5(a), rect ①&④ are divided by Backward Dividing Line (BDL) which is in the opposite direction in terms of polygon vertices numbering. In Figure 5(a), rect ③&⑥ are divided by Forward Dividing Line (FDL) which is in the clockwise direction, same as the polygon vertices numbering. Rect ②&⑤'s dividing lines come from both directions.



(a) Polygon partitioning (b) Shape rectification

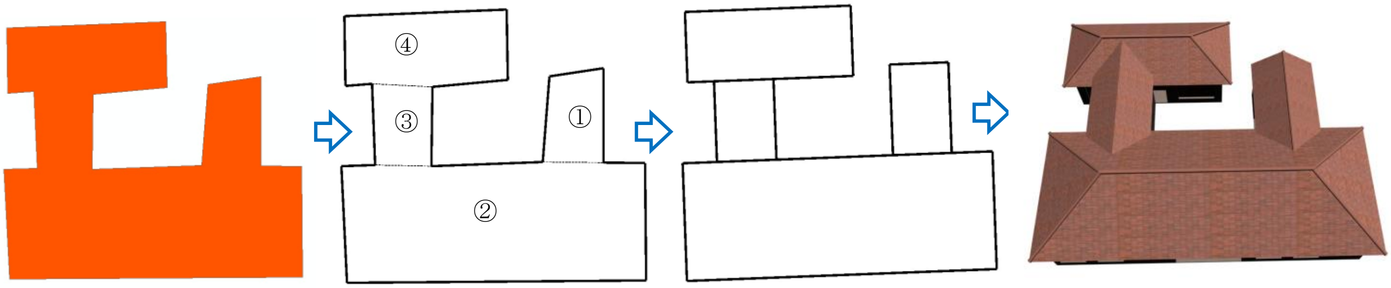
Figure 5: How branch rects are cut off: 3 cutting pattern: FDL(③&⑥) and BDL(①&④) , F&B DL(②&⑤)

This dividing pattern is saved at the branch rect, and is used for shape rectification. While these divided branch rects are looking for an adjacent rect and will extend to the adjacent rect, divided rect ② in Figure 4 (b) is not looking for a rect but absorbing an adjacent branch rect. The divided rect whose n_R is more than three can be an absorbing rect. After rect ① in Figure 4 (b) being cut off, rect ② has four consecutive 'R' vertices (=n_R).

4.3 Shape rectification

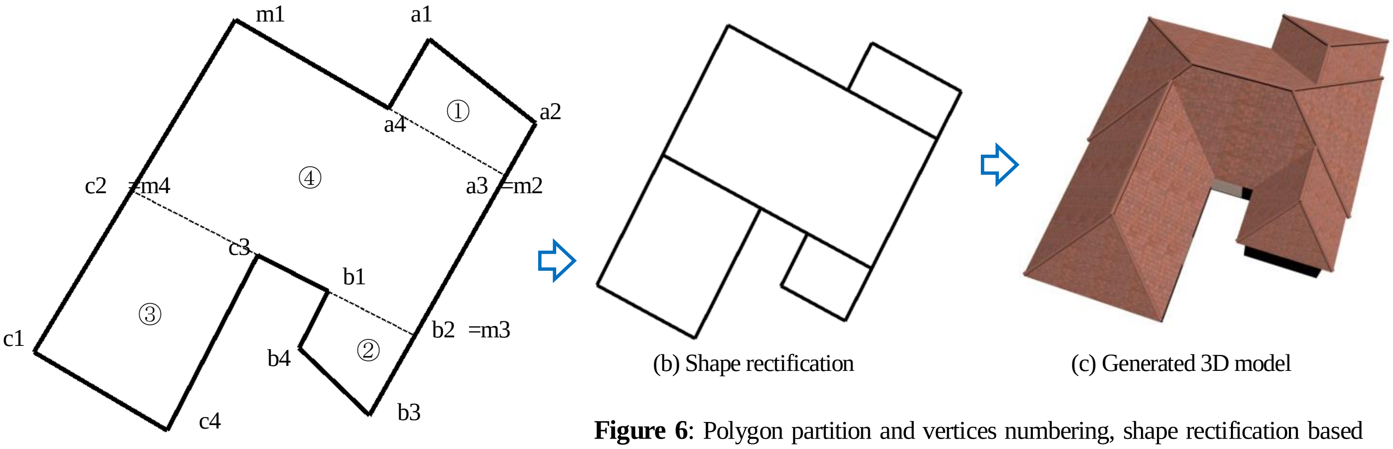
Specifically, the rectification procedure is implemented to the polygon shown in Figure 6 as follows. Before polygon partitioning, all edge length and edge inclination of the polygon are calculated, and the length of all edges are sum up according to the snapped angle of all edge inclination. Then, the angle for a longest sum up edge length can be adopted as the 'main angle' of the polygon, which will be used as the inclination of all partitioned rectangles. Figure 6 shows the process of polygon partition and shape rectification, automatic 3D modeling. Vertices of a partitioned polygon, i.e., a quadrilateral is numbered clockwise with the start point of a longest edge facing right as pt1 (a1, b1,..) or with the start point of a longest edge facing left as pt3 (a3, b3,..) as shown in Figure 6(a).

When a quadrilateral (quad) is cut off, the dividing pattern and which edge of the quad is cut off (active edge) is saved at the quad. During the searching stage, an active quad will search for an adjacent quad by locating which quad the checking point on the active edge contains, and then checking on which edge of the adjacent quad the checking point is. In case of quad ① in Figure 6(a), edge a3a4 will be an active edge, and search for an the adjacent quad. After searching and having found out the adjacent quad is quad ④ and the adjacent edge is m1m2, the mutual vertex is a3(=m2), which the rectification procedure uses as a 'standard position' for rectification. The rectified position of the vertices of quad ① are calculated as follows.



(a) Original building polygon (b) Polygon partitioning (c) Shape rectification (d) Generated 3D model

Figure 4: Process of polygon partition and shape rectification, generation of 3D model



(a) Polygon partitioning

(b) Shape rectification

(c) Generated 3D model

Figure 6: Polygon partition and vertices numbering, shape rectification based on the generatrix, generation of 3D model

a1.x=m2.x + w_S*cosθ- w_L*sinθ
a1.y=m2.y + w_S*sinθ+ w_L*cosθ
a2.x=m2.x +w_S*cosθ: a2.y=m2.y + w_S*sinθ
a4.x=m2.x - w_L*sinθ: a4.y=m2.y + w_L*cosθ

where θ is the main angle and w_S is the average length of two short sides of the rectangle, and w_L is the average length of two long sides of the rectangle.

5. Conclusions

Here are the examples of rectification of approximately orthogonal building polygons and an automatically generated 3D urban model in Figure 7 & 8. In the digital map, not all building polygons are precisely orthogonal. However, creating 3D building models are expected to be orthogonal. When placing a set of boxes as building bodies for creating the buildings, there may be gaps or overlaps between these boxes if building polygons are not precisely orthogonal. In this paper, the new methodology is proposed for rectifying the shape of building polygons and generating 3D building models without any gap and overlap.

References

Carlos, V. A., Daniel, A. G., andr Bedřich, B. (2010). "Building reconstruction using Manhattan-world grammars", Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on: 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., and Daniel, B. R. (2007). "Style Grammars for interactive Visualization of Architecture", Visualization and Computer Graphics, IEEE Transactions on Volume:13, 786 – 797

Müller, P., Wonka, P., Haegler, S., Ulmer, A., and Van Gool, L.(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, vol. 28, no. 5

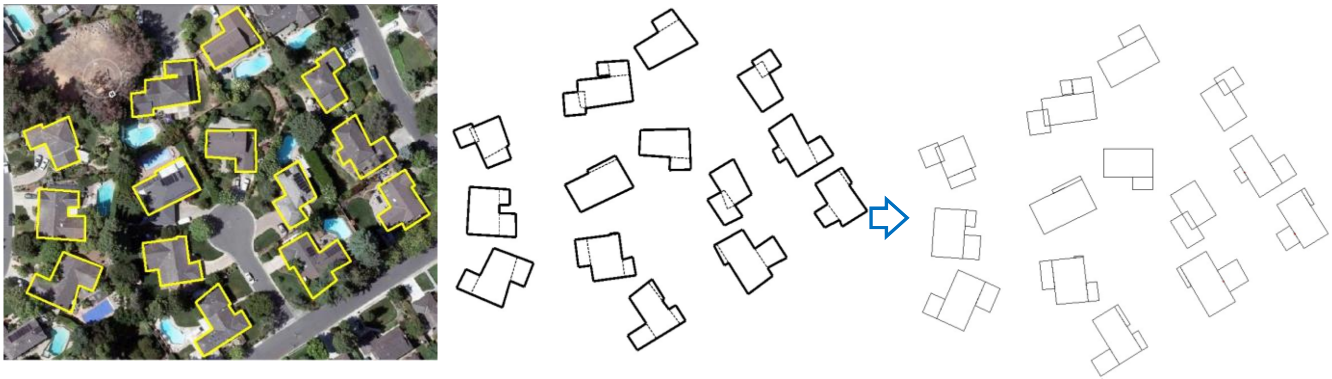
O. Aichholzer, F. Aurenhammer ,and D. Alberts, B. Gärtner (1995). "A novel type of skeleton for polygons", Journal of Universal Computer Science, 1 (12): 752–761.

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 Kenichi (2006). "Generalized Building Polygon Partitioning for Automatic Generation of 3D Building Models", ACM SIGGRAPH 2006, Posters Session Virtual & Augmented & Mixed Reality & Environments

Kenichi SUGIHARA(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



(a) Building polygons on satellite image

(b) Polygon partitioning

(c) Shape rectification

Figure 7: Application of the System: Polygon partition and shape rectification, generation of 3D urban model

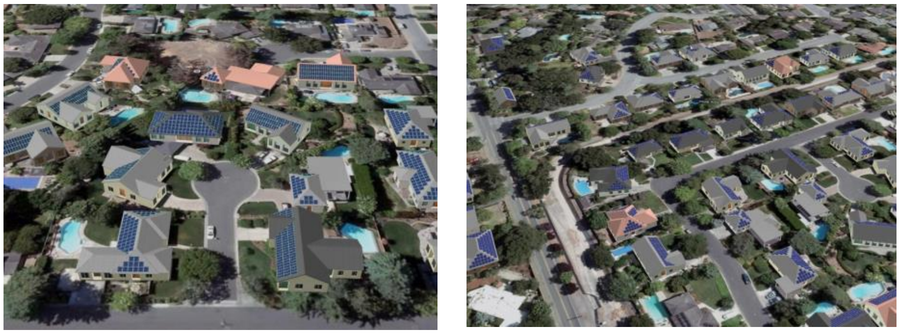


Figure 8: Automatically generated 3D urban model by polygon rectification