Automatic Generation of 3-D Building Models by Straight Skeleton

Kenichi Sugihara
Gifu Keizai University

1. Introduction

Automatic generation of 3-D building models as shown in Figure 1 is important in several fields, such as urban planning and gaming industries. However, enormous time and labour has to be consumed to create these 3-D models, using 3-D 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, a GIS and CG integrated system for automatically generating 3-D building models has been proposed, based on building polygons (building footprints) on a digital map shown in Figure 1 left [Sugihara 2006]. Figure 1 left also shows most building polygons’ edges meet at right angles (orthogonal polygon). An 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 the digital map, however, not all building polygons are orthogonal. In this paper, the new system is proposed for placing parts of a building properly, in either orthogonal or non-orthogonal polygons. This proposed system will place building parts, such as doors and windows, fences, shop façades along the inner contour which is setback from the original building footprint by straight skeleton computation defined by a continuous shrinking process [Aichholzer et al. 1995].

Since 3-D urban models are important information infrastructure that can be utilized in several fields, the researches on creations of 3-D urban models are in full swing. Various types of technologies, ranging from computer vision, computer graphics, photogrammetry, and procedural modeling, have been proposed and developed for creating 3-D urban models.

Müller et al. [2006] 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 also 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. However, the algorithm by the straight skeleton is not shown and there is no digital map description in their paper. My approach deals with digital maps and clarifies the algorithm by straight skeleton.

2. Inner Polygon by Straight Skelton

Parts of a building, such as doors and windows and setbacked façade, are placed on the inner polygon setbacked by a fixed distance from the footprint or the original polygon of the building. This inner polygon receded by a fixed distance is computed by the straight skeleton [Aichholzer et al. 1995] which is defined by a shrinking process in which each edge of the polygon moves inwards parallel to themselves at a constant speed as shown in Figure 2. Lengths of edges of the polygon might decrease or increase in this process. The edges incident to two reflex vertices will grow in length as shown by ‘ed3’ in Figure 2. A reflex vertex is a vertex whose internal angle is greater than 180 degrees. 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. Each vertex of the polygon moves along the angular bisector of its incident edges. This situation continues as long as the boundary does not change topologically. 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.

After either type of event, we are left with a new, or two new, polygons which are shrunk recursively if they have non-zero area. Shrinking procedure is uniquely determined by the distance between the two edges of before & after shrinking procedure. The distance is referred to as d_{shri}, shown by e_{d_{shri}} in Figure 2. e_{d_{shri}} is the d_{shri} when an ‘edge event’ happens in the shrinking process.

Figure 1 Building polygons on a digital map (Left) and automatically generated 3D building model (Right)
The edge event $d_{shri}$ is calculated as follows:

$$d_{shri} = \frac{L_i}{\cot(0.5 \times \theta_i) + \cot(0.5 \times \theta_{i+1})}$$  \hspace{1cm} (1)

Where $L_i$ is the length of $ed_i$, $(edge)$, and $\theta_i$ & $\theta_{i+1}$ are internal angles of vertices incident to $ed_i$.

Edge event will happen when $0.5 \times \theta_i + 0.5 \times \theta_{i+1} < 180$ degrees, i.e., the sum of the internal angles of two vertices incident to an edge is less than 360 degrees. After calculating $d_{shri}$ for all edges and finding the shortest of them, when $d_{shri}$ reaches the shortest found, an edge event happens and edges disappear for the first time in the process. In Figure 2, when $d_{shri}$ is $e_d$, ed4 and ed5 disappear. The edge (ed3) incident to two reflex vertices grows in length in shrinking process.

The straight skeleton, $S(P)$, is defined as the union of the pieces of angular bisector traced out by polygon vertices during the shrinking process. Figure 4 left shows non-orthogonal building polygons overlapped by satellite image. In Figure 4 middle, inner building polygons are generated by the straight skeleton, and along these inner polygons, windows and setbacked façade, fences are placed and thus 3-D building model are automatically generated as shown in the right of Figure 4.

3. Conclusion

In this paper, the algorithm by straight skeleton is clarified for generating inner polygons along which parts of the building are placed. The methodology is applicable for orthogonal and non-orthogonal building polygons. Thus, the proposed integrated system succeeds in automatically generating general shaped 3-D building models with roofs and parts of a building, such as doors, windows, fences and setbacked façade, are placed along the inner polygons, based on general shaped building polygons.

References


Figure 2 Inner polygons shrunk by a distance of $d_{shri}$ by straight skeleton computation

Figure 4 Automatically generated 3D building model from non-orthogonal polygon