Creating Topologically Consistent 3D City Models of LOD+ with Extrusion

Yunfei Shi[1],Biao He [2],

[1]College of Resources & Environment, Linyi University, 276000 Linyi, China,
shiyunfei_001@163.com
[2]Shenzhen Municipal Committee of Urban Planning,Land Resource and Real Estate,
518040 Shenzhen, China, 1688698@qq.com

Abstract. There are many methods to create 3D city model, in which the simplest one is to extrude building footprints to obtain "block-shaped" polyhedra representing buildings. While the resulting 3D city models are not topologically consistent, if the 2D topological relationships between the footprints are not taken into account. As a result, the created models only meet the purpose of visualization. In this paper, we focus on creating topologically consistent 3D city models from the footprints of heterogeneous structure buildings. The generated models are more detailed than LOD1 of CityGML and are called LOD1+, which not only construct topology between buildings but also generate internal structure topology of them. We have implemented the procedure, tested it with real-world datasets, and validated it.

Keywords: 3D city model, extrusion, topological consistent, heterogeneous structure building

1   Introduction

3D city models have been increasingly applied as data source or visualization method in a growing number of fields such as urban planning, architecture, archaeological reconstruction, tourism, civil engineering, mobile telecommunication, energy supply, navigation, etc[1].Different applications require various detail and accuracy of 3D model. For this purpose, 3D city models were divided into five levels of details (LODs) by City Geography Markup Language (CityGML), from LOD0 where only the terrain is stored to LOD4 where buildings have detailed roofs structures, windows, rooms and even pieces of furniture [2].

There are many ways to create 3D city model，in which the simplest one is arguably with extrusion. That is, given a set of footprints representing the buildings or rooms (polygons in the plane), assign them a height and then push them upwards to create polyhedra[3]. Extrusion is indeed easy if the only thing you want to do with your 3D city model is to look at it! [2]. Many commercial software can extrude polygons to obtain buildings, but the resulting models cannot be stored in a topological data structure [3]. The reason is that the created models are not topologically consistent. In a nutshell, that means that extruding results contains

duplicate points, overlapping faces, faces intersecting where there are no points, etc[2]. Fig.1 illustrates a process where two building footprints are being extruded. Extrusion without considering other footprints only guarantee that each resulting polyhedron is valid, but not ensure that a group of footprints will generate a topologically consistent 3D city models.

(a)   (b)   (c)
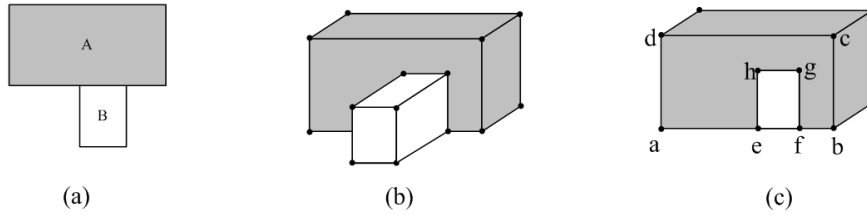
Fig.1. (a)Two polygons in the plane.(b)Two polyhedra obtained by extrusion of the two polygons.(c) To be topologically consistent the front polygon of the polyhedron obtained by the extrusion of A should be modeled with two polygons: efgh and aehgfbcd.

Many studies [2] [3] have been developed on topologically consistent 3D city model by extrusion. Models created by existing studies belong to LOD1 in CityGML. In this level, whole building is represented as a "block-shaped" model with flat roof structures and no internal floors and rooms. However, a building often includes many floors and rooms in the real world. Existing studies only consider the topological construction between buildings, but do not take into account that of internal structure (floors and rooms) of buildings. In fact, topology of internal structure of buildings is very important for many applications, such as indoor navigation [4], safe escaping and so on. In addition, because a building contains multiple floors and the different floor corresponds to different structure (i.e. heterogeneous structure), that make the extruding process need to consider simultaneously vertical and transverse topology. Therefore, topological construction of internal structure of buildings is more complicated than that between buildings.

In this paper, we focus on creating more detailed topologically consistent 3D city models than LOD1 and are called LOD1+, which not only construct topology between buildings but also generate internal structure topology of them. Our main contribution is to create topologically consistent 3D City models of LOD1+ level from heterogeneous structure buildings. The remainder of this paper is organized as follows. Section 2 provides the basis of research. Section 3 gives our extruding approach for heterogeneous structure buildings. Section 4 gives ours experiments. Finally, section 5 offers conclusions of our work.

2 The Basis for research

Topology is one of the mechanisms to describe relationships between spatial objects. Thus, it is the basis for many spatial operations. Let S be a set of spatial objects in $\mathbb{R}^3$, the three dimensional Euclidean space. Spatial objects in S are formed by four geometric primitives: (i) ρ^0 is a point(0 dimensionality); (ii) ρ^1 is a straight line

segment (1-dimensionality), which is referred to simply as a line in the following; (ⅲ) ρ^2 is a polygon (2- dimensionality); (ⅳ) ρ^3 is a polyhedron (3- dimensionality).

Topological consistency: Let S be a set of spatial objects in $\mathbb{R}^3$, we say that S is topologically consistent if the following rules are valid [3]:

(1) every line segment ρ^1 in S is formed by two points ρ^0 also in S;

(2) the intersection of two lines ρ^1_1 and ρ^1_2, denoted $\rho^1_1 \cap \rho^1_2$, is either empty or is a point in S;

(3) the intersection of two polygons ρ^2_1 and ρ^2_2, denoted $\rho^2_1 \cap \rho^2_2$, is either empty or a finite set of primitives in S (points or lines).

(4) the intersection of two polyhedrons ρ^3_1 and ρ^3_2, denoted $\rho^3_1 \cap \rho^3_2$, is either empty or a finite set of primitives in S (points or lines or polygons).

The related storage structures of topological data include half-edge, winged-edge [5], Quad-edge [6] and so on. From the view of manifold, they belong to manifold topology. In addition, many researchers also propose non-manifold data structures, such as Face-edge [7], Radial edge, X-maps and so on. The paper uses half-edge as data structure to storage the data with extrusion.

3 Our extruding approach

This section presents our algorithm to create topologically consistent 3D city model. The algorithm is based on the idea that the floor plans taken as the input are a set of topologically consistent polygons (in 2D). We use existing GIS tools (e.g. arcgis) to create topologically consistent 2D dataset as input and output topologically consistent 3D models of buildings and a set of faces embedded in 3D. In created models, a building (polyhedra) is enclosed by its bounding faces and simply has a reference to these faces, and faces (or edges or nodes) shared by adjacent polyhedra (or faces or edges) are not duplicated.

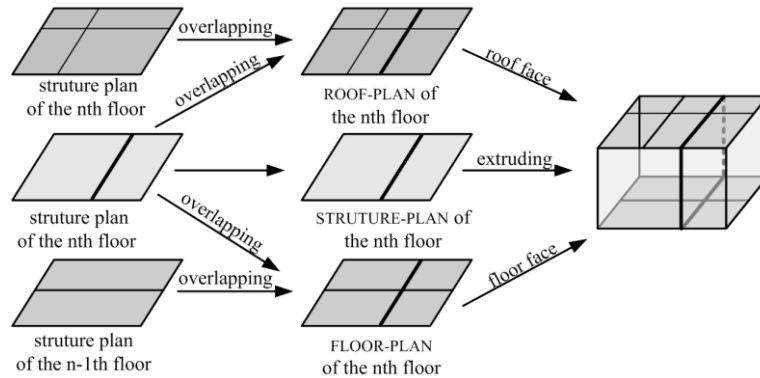### 3.1 Generation of ROOF-PLAN and FLOOR-PLAN

According to the above analysis, the generation of floor faces or roof faces is influenced by the structure of adjacent floors. Therefore, extruding process needs to consider synthetically the relationship between the extruded floor and its adjacent floors. For this reason, we divide floor plans into three types: STRUCTURE-PLAN, ROOF-PLAN and FLOOR-PLAN.

STRUCTURE-PLAN is used to create vertical walls (polygons) and determines the unit structure of the whole floor. If the extruded floor is the *nth*, STRUCTURE-PLAN will be the original floor plan.

ROOF-PLAN is used to create roof face of the extruded floor. If the extruded floor is *n*, ROOF-PLAN will be obtained by overlapping the *nth* and the *(n+1)th* STRUCTURE-PLAN (As shown in Fig.2).If building only has *n* floors, ROOF-PLAN is equivalent to STRUCTURE-PLAN.

FLOOR-PLAN is used to create floor face of the extruded floor. If the extruded floor is *n*, ROOF-PLAN will be obtained by overlapping the *nth* and the *(n-1)th* STRUCTURE-PLAN (As shown in Fig.2). If *n* is equal to 1, FLOOR-PLAN is equivalent to STRUCTURE-PLAN.

From these three definitions, we can find that ROOF-PLAN of the *nth* floor is equivalent to FLOOR-PLAN of the *(n+1)th* floor and FLOOR-PLAN of the *nth* floor is equivalent to of the *(n-1)th* ROOF-PLAN floor. For a building with n floors, it needs *n* STRUCTURE-PLAN and *n+1* ROOF-PLAN or FLOOR-PLAN.
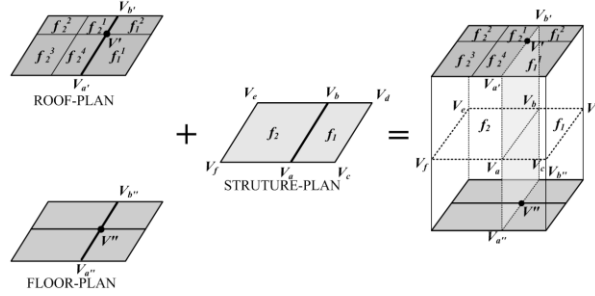

Fig.1. Generation of ROOF-PLAN and FLOOR-PLAN

The overlap operation can use existing tools (e.g. arcgis) to implement. In order to ensure overlapped plans are topologically consistent, clean operation is used to reconstruct topology. In this way, we create topologically consistent ROOF-PLAN and FLOOR-PLAN for every floor of a building.

3.2 Topological construction of vertical direction

After the generation of ROOF-PLAN and FLOOR-PLAN, the next step is to create vertical walls and body data. ROOF-PLAN and FLOOR-PLAN only determine the structure of roof face and floor face of extruded floor, but the whole floor structure is determined by STRUCTURE-PLAN. It's noteworthy that STRUCTURE-PLAN is only used as reference to create vertical walls but itself is not the topological primitive of created model, which is different from the usage of ROOF-PLAN and FLOOR-PLAN.

Let's use a simple sample to illustrate. As shown in Fig.3, vertical shaded face (wall) is created by extruding the line segment *ab*. But *ab* is not a part of the created face. The edge set of the created face is composed by the following line segments: $\{V_{a''}V'',\ V''V_{b''},\ V_{b''}V_{b'},\ V_{b'}V',\ V'V_{a'},\ V_{a'}V_{a''}\}$. It can be seen that each vertex in STRUCTURE-PLAN creates a vertical edge, and each edge and each polygon in STRUCTURE-PLAN creates a vertical face and a polyhedron separately.

Fig. 2. Generating process of vertical wall

Generation of vertical elements. *Creating vertical edges.* Taking out a vertex from the vertices set of STRUCTURE-PLAN, it seeks the vertex corresponding to counterpart in ROOF-PLAN and FLOOR-PLAN respectively and forms a vertical edge. For instance, vertex V_a corresponds to vertex $V_{a'}$ in FLOOR-PLAN and $V_{a''}$ in FLOOR-PLAN respectively (Fig.3).Connecting $V_{a'}$ and $V_{a''}$, a vertical edge is created. In order to facilitate following work, the direction of edge is from the vertex in FLOOR-PLAN to the vertex in ROOF-PLAN. Iterating the process, we can obtain the vertical edge set.

*Creating vertical faces (walls).*For each edge of STRUCTURE-PLAN, corresponding edge set is sought in ROOF-PLAN and FLOOR-PLAN. An edge in STRUCTURE-PLAN may correspond to multiple edges in ROOF-PLAN and FLOOR-PLAN. For example, the corresponding edges of edge V_aV_b are $V_{a'}V'$ and $V'V_{b'}$ in ROOF-PLAN (fig.3).The found corresponding edges (crosswise edges) and the created vertical edges generate vertical faces in common. As shown in fig.3, the vertical face created by edge V_aV_b is $V_{a''}V''V_{b''}V_{b''}V'V_{a'}V_{a''}$, which is formed by edge set $\{V_{a''}V'', V''V_{b''}, V_{b''}V_{b'}, V_{b'}V',, V'V_{a'}, V_{a'}V_{a''}\}$. Here, we need to sort the edge set and make them arrangement in clockwise or counterclockwise order.

Creating polyhedra. For each face (polygon) in STRUCTURE-PLAN, corresponding face set is sought in ROOF-PLAN and FLOOR-PLAN. Similarly, a face in STRUCTURE-PLAN may correspond to multiple faces in ROOF-PLAN and FLOOR-PLAN. The corresponding faces of face f_2 are f_2^1, f_2^2, f_2^3 and f_2^4. The found corresponding face (horizontal face) set and the created walls compose polyhedra in common.

Algorithm of seeking corresponding relationships. The above three steps all have a core issue: seeking corresponding relationship, namely, finding vertices, edges and faces in STRUCTURE-PLAN corresponding counterparts in ROOF-PLAN and FLOOR-PLAN separately. The seeking process of three corresponding relationship is following.

Seeking corresponding vertices. Due to the model obtained by extrusion, the extruded vertex should have same x and y coordinates in vertical direction. When we seek the corresponding vertices of the vertex A (x, y, z) in STRUCTURE-PLAN, z coordinate can be ignored and seek the nearest vertex to A (x, y). Theoretically, the coordinates obtained nearest vertex should equal to A. However, owing to storage error of computer, the nearest vertex is not equal to A and has a little offset. Therefore, the nearest vertex is the seeking corresponding object exactly.

Seeking corresponding edges. Based on the method of seeking corresponding vertices, we can obtain the corresponding vertices in ROOF-PLAN and FLOOR-PLAN of vertices of each edge in STRUCTURE-PLAN. As shown in fig.3, corresponding vertices in ROOF-PLAN of vertices of edge V_aV_b are $V_{a'}$ and $V_{b'}$. After confirming corresponding vertices, it needs to check whether $V_{a'}$ and $V_{b'}$ can make up an edge. If they can, the edge $V_{a'}V_{b'}$ is corresponding edge. Otherwise, we need to ignore z coordinate and find the vertex which distancing $V_{a'}V_{b'}$ less than given threshold in ROOF-PLAN. If the vertex is found, we think that the vertex is on the line. For example, Vertex **V'** is on the $V_{a'}V_{b'}$ in fig.3. By the way, corresponding edge set can be found.

Seeking corresponding faces. With the method of seeking corresponding edges, we can acquire the corresponding edges (in ROOF-PLAN or FLOOR-PLAN) of all edges of each face in STRUCTURE-PLAN. The obtained corresponding edges draw up a range in ROOF-PLAN and FLOOR-PLAN. If the corresponding edges just surround a face, the surrounded face is corresponding face. Otherwise, we need to judge all faces of ROOF-PLAN (or FLOOR-PLAN) and pick out those faces which drop in the tagged range of the corresponding edges. These faces compose the corresponding face set.

   With above algorithm, topological consistent 3D models of every floor are created. The next step is to build the topology between floors. Because the extrusion of every floor is independent, the common elements (nodes, edges and faces in ROOF-PLAN or FLOOR-PLAN) of adjacent floors are reconstructed twice. Moreover, the duplicated common elements are same in structure. Thus, the topological construction of adjacent floors can be implemented by deleting the duplicated nodes, edges and faces and adding references to the reserved common topological primitives for adjacent polyhedra in different floors. By this way, the created 3D buildings models include a set of complete topological relationships, in which common faces are shared by adjacent polyhedrons, and common edges are shared by adjacent faces and the common nodes are shared by adjacent edges.

4   Experiments

With the proposed algorithm, we establish an experiment system which takes Visual C++ as development tools, and use Open Scence Graph (OSG) as graphic interface and choose a small residential area as test data. The data include floor plans and elevation tables of every building. Floor plans provide the structure information of

every floor and elevation tables provide the height information of every floor. Fig.4 gives a process of a sample building, which only has two floors. (a) is the structure plan of the first floor; (b) is the structure of the second floor; (c) is the result of extrusion.
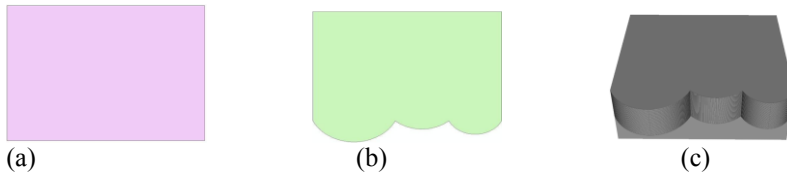

| (a) | (b) | (c) |

Fig. 3. (a) Structure plan of first floor; (b) Structure plan of second floor;(c)Model by extrusion

Fig.5 is the modeling process of complicated building with two floors. (a) and (b) are the structure plans of the first floor and the second floor separately. (c) is the extruding result.
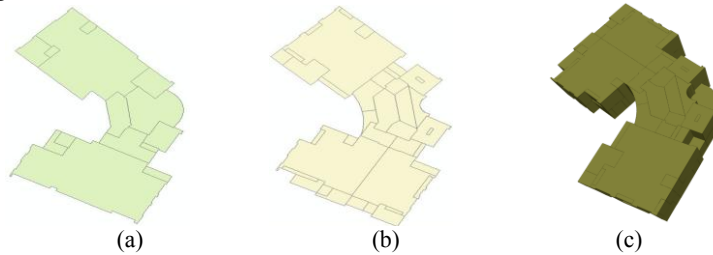

| (a) | (b) | (c) |

Fig. 4. (a)Structure plan of first floor; (b)Structure plan of second floor;(c) Model by extrusion

Fig.6(a) is a sample of whole building with extrusion; (b) is the model of a small residential area with extrusion.
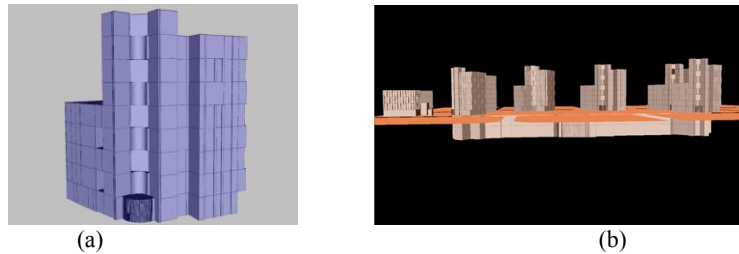

| (a) | (b) |

Fig. 5. (a) Building with extrusion; (b) Small residential area with extrusion

5   Conclusion

The paper analyzes the application demands of 3D city model and proposes a method for constructing topological consistent 3D city model. With the method, we can quickly generate 3D data of large area buildings, avoid the re-survey of buildings and

save the cost of data collection. The shortage of the method is that it can't handle the building with slope walls, which is our further goal.

References

1. Tack,F., Buyuksalih,G.,Goossens,R.: 3D building reconstruction based on given ground plan information and surface models extracted from spaceborne imagery. ISPRS Journal of Photogrammetry and Remote Sensing. 67, 52-64(2011)
2. Ledoux, H., Meijers, M.: Extruding building footprints to create topologically consistent 3D city models. In: A. Krek, M. Rumor, S. Zlatanova, and E. Fendel, editors, Urban and Regional Data Management, UDMS Annual 2009.39-48(2009).
3. Ledoux, H., Meijers, M.:Topologically consistent 3D city models obtained by extrusion. International Journal of Geographical Information Science. 25:557-574(2011).
4. Lee, J., Zlatanova, S.: A 3D data model and topological analyses for emergency response in urban areas. In: Zlatanova, Li (eds.), Geospatial In-formation Technology for Emergency Response, Taylor & Francis (2008)
5.Baumgart, B.G.: A polyhedron representation for computer vision, National Computer conference. Anaheim, CA, AFIPS, 589–596 (1975)
6. Guibas,L., Stolfi,J.: Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams, ACM Transactions on Graphics, 4,74-123(1985)
7.Dobkin, D.P., Laszlo, M.J.:Primitives for the manipulation of three-dimensional subdivisions. Algorithmica. 4, 3-32(1989)