

Construction of 3D Volumetric Objects for a 3D Cadastral System

Shen Ying,^{*} Renzhong Guo,[†] Lin Li,^{*} Peter Van Oosterom[‡] and Jantien Stoter[‡]

^{*}*School of Resource and Environmental Sciences, Wuhan University*

[†]*Urban Planning, Land and Resources Commission of Shenzhen Municipality*

[‡]*Faculty of Architecture and the Built Environment, Delft University of Technology*

Abstract

This article presents a new method to illustrate the feasibility of 3D topology creation. We base the 3D construction process on testing real cases of implementation of 3D parcels construction in a 3D cadastral system. With the utilization and development of dense urban space, true 3D geometric volume primitives are needed to represent 3D parcels with the adjacency and incidence relationship. We present an effective straightforward approach to identifying and constructing the valid volumetric cadastral object from the given faces, and build the topological relationships among 3D cadastral objects on-the-fly, based on input consisting of loose boundary 3D faces made by surveyors. This is drastically different from most existing methods, which focus on the validation of single volumetric objects after the assumption of the object's creation. Existing methods do not support the needed types of geometry/topology (e.g. non 2-manifold, singularities) and how to create and maintain valid 3D parcels is still a challenge in practice. We will show that the method does not change the faces themselves and faces in a given input are independently specified. Various volumetric objects, including non-manifold 3D cadastral objects (legal spaces), can be constructed correctly by this method, as will be shown from the results.

1 Introduction

In our contemporary social context, the development of land use has subdivided land parcels into three-dimensional (3D) spaces according to certain property rights, especially in metropolitan areas with dense population. This results in 3D parcels (LADM 2012) above or below the land surface. In such circumstances, the local government needs to construct and manage 3D cadastral objects to be able to manage the development of real urban 3D spaces appropriately. This requires the geometric representation of these 3D parcels to be modeled as volume primitives in a DBMS environment. A volumetric primitive (we also use the term “solid” in the article) is a complete representation of a polyhedron able to support the various calculations and analysis related to the 3D cadastral objects, as in Figure 1. The volumetric primitives in 3D space need to be mutually exclusive and they need to exhaustively partition the extent of the domain (i.e. no gaps are allowed). There are many box-shaped 3D parcels or irregularly shaped buildings in reality (Figure 1a), and sometimes even the structure of a single 3D

Address for correspondence: Shen Ying, School of Resource and Environmental Sciences, Wuhan University, Wuhan, China. E-mail: shy@whu.edu.cn

Acknowledgements: The work is supported by the National Natural Science Foundation of China under Grant 41371369 and the Ministry of Economic Affairs, Agriculture and Innovation under Grants 11300 and 11185). The authors thank the associate editor and the referees for their helpful comments and suggestions.

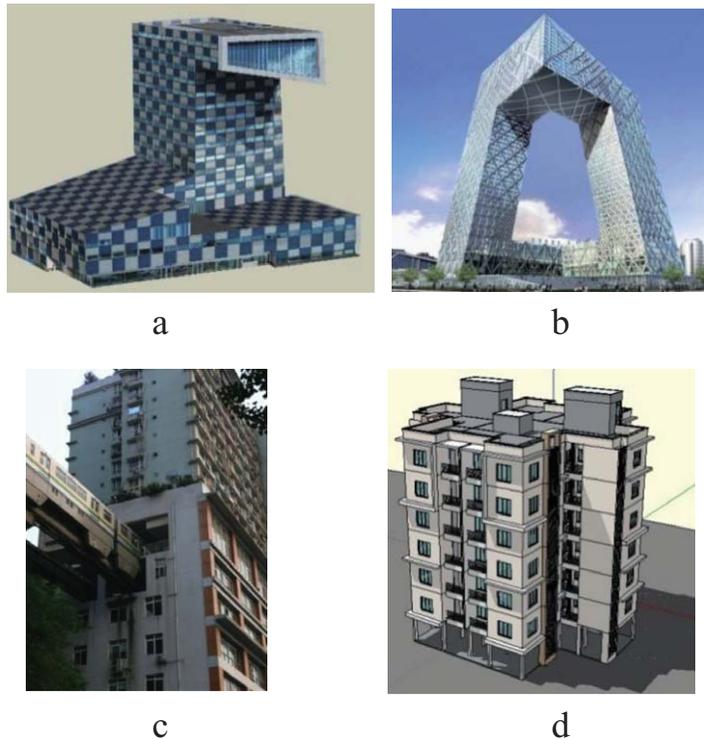


Figure 1 Real entities with box-based polyhedron (building in Rotterdam) (a), 3D hole (CCTV building in Beijing) (b), and through-hole (light railway station within building in Chongqing) (c), and normal condominium (d)

cadastral volumetric object can be quite complex: caves in the interior, through-holes, singularities as illustrated in Figure 1, which can be represented as non-manifold objects (Figures 1b-c). Typically, multiple distinct apartments forming a complicated condominium are always tangent along the shared face/edge (Figure 1d); the focus will shift to the topological relationship among these aggregated cadastral objects. Each apartment as a volume should be recognized and built up from the assembled faces and then correspond to 3D parcels.

Therefore, proper 3D geometries are required for 3D cadastres. Surveying data can be investigated by the surveyors or the engineers, thus the creation and submission of 3D volumetric objects are the key phases in a 3D cadastre system. However, what are acceptable (valid) 3D cadastral object representations and how to create their 3D geometries (even the non-2-manifold geometries) are still challenges (Van Oosterom 2013). The non-manifold 3D representations (self-touching in edge or node; see Figure 2) are not well supported by current GIS, CAD, and DBMS software or by generic ISO standards such as ISO 19107 (Van Oosterom 2013).

For a proper visualization of 3D objects discrete individual faces may be sufficient. However, in a 3D cadastral system, a real 3D geometric volume is required to describe the true 3D characteristics of the 3D cadastral objects, and closed 3D volumetric representations of 3D objects play an essential role (Thompson and Van Oosterom 2011). We need the proper *boundary faces* to define a volume as well as an effective method for determining each

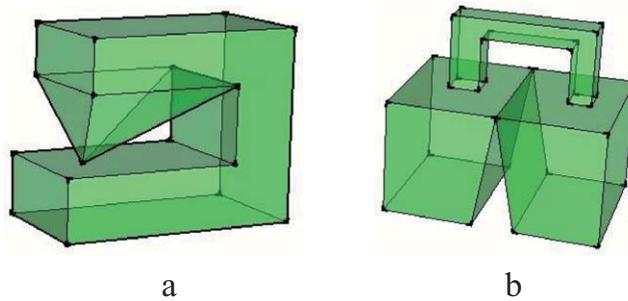


Figure 2 Solids with non-manifold conditions: (a) point non-manifold condition; and (b) edge non-manifold condition

boundary face with its normal vector to delimit the closed interior space of the 3D polyhedron. At least three aspects should be clearly developed in order to manage the 3D parcels correctly: (1) precise geometric models that describe the shapes and geographic locations of various 3D parcels, based on the flat faces; (2) volumetric or solid models that indicate all its boundary faces with orientation to present the corresponding 3D parcel objects; and (3) the topological relationships that encode all the information about the adjacencies among 3D parcels shared common faces/edges to keep the consistency of the objects' geometries and support spatial query and management.

This article describes a detailed method for direct volumetric model construction (representing both geometry and topology), which does not need solid reconstruction nor post-construction validation¹ – a technique almost all current existing methods rely on. This method is more logically rigorous. The input are loose boundary 3D faces acquired by surveyors as the cadastral boundaries with official identifications, stored in a database. This is why polyhedron-based volume in this article is envisaged in 3D cadastres, not tetrahedron-based volume.

The article is organized as follows. After discussing the current state-of-the-art in Section 2, related knowledge, the basic land administration model and our model are stated in Section 3. Details of direct 3D geometric volume construction based on face-sorting and face-connecting are presented in Section 4. The topological processing to achieve the target model, as well as the feasibility and reliability of the methods are also discussed in this section. Test cases are described in Section 5. Finally, the conclusion and outlook are delivered in Section 6.

2 Related Works

As far as the valid volumetric object is concerned, also named solid, there are several definitions that are a little different depending on different starting points. Some reviews of the ISO 19107 Spatial Schema are given (Kazar et al. 2008; Verbree and Si 2008). Observe that the ISO 19107 solids are not sufficient for 3D cadastral applications: the ISO19107 solid is a simple solid whose shell is not allowed to touch (they have to be 2-manifold). The Computational Geometry Algorithms Library (CGAL) defines a polyhedron in a practical way as a 3D solid composed of vertices, edges, faces and an incidence relationship between them (CGAL 2011). The organization beneath is a half-edge data structure, which restricts the class of

representable surfaces to orientable 2-manifolds (Verbree and Si 2008). The definition of Oracle SDO_GEOMETRY based on ISO 19107 is designed for the storage of one single isolated object. Implementation limitations in Oracle include: (1) each SDO_GEOMETRY has its own 3D coordinates, which results in duplicate data in the solid set; (2) faces are simple and do not have holes; and (3) relationships among SDO_GEOMETRY are not built and given.

In this study, we follow the definition of a 3D parcel as given by Kazar et al. (2008) and Thompson and Van Oosterom (2011), in which the main rule is that the volumetric object is internally connected, which means that a shell can self-touch, as long as the interior of the solid stays connected. Hence, we can state that a valid volumetric object is a 3D primitive that can be represented by one closed polyhedron, refined by a set of connected faces. The volumetric object satisfies the following characteristics: closeness, interior connection, face-construction and proper orientation. Evidently, the volumetric object here can have through-hole/ring or cavity that allows its boundary faces to touch each other, which is not a 3-manifold in some cases.

The problem with existing methods is that they are indirect, require one to make secondary assumptions, and then need data-driven justifications to determine whether the original hypothesis holds. These methods merely validate or re-verify the isolated or single volumetric object but cannot directly recognize and extract it, or its set, from the set of 3D faces. Some software, such as SketchUp and Oracle, know how to check whether there is a correct solid with the pre-selected faces, but neither provide available functions to convert the faces to a real 3D volumetric object, nor know how to detect and extract the volume surface from the faces, not to mention the assembled volumetric objects. The check of validation on existing solid objects in Oracle is analyzed according to certain conditions related to the simple/complex face, avoiding the procedures of straightforward solid construction (Thompson and Van Oosterom 2011).

These methods of post assumptions validation for finding the solid are limited by the pre-defined 3D “object/solid”, which may or may not really be a valid solid. Why should we restrict the solid validation methods to post assumption validation (for example, see Weiler 1988b and Brugman et al. 2011) where the result depends on the geometry of the presupposed “solid”? Moreover, this verification method is merely aiming at an individual “solid”. If it is not valid what can we do? Modifying the data may result in geometric conflicts with other connected solids.

Given the above limitations, we aim to develop a more direct method of the solid validation process, describing the steps below:

1. We build valid solids at the beginning of object generation to satisfy the validation requirements.
2. If we build a valid solid and the sets of solids directly there is no need to validate its existence afterwards.

In 3D urban environments, 3D land parcels are the main objects we should model and manage. 3D parcels cannot intersect and penetrate mutually, and each of them has precise geometric boundaries to construct the valid volumetric object that is a closed spatial occupation, based on the Boundary-Representation (B-Rep) (Stoter and Zlatanova 2003). More importantly, volumetric models based on B-Rep with 3D topology support the polyhedron/polyhedra with holes, singular shapes or non-manifold configurations (Lee and Lee 2001), which are critical to represent 3D cadastral objects. Many topological data structures have been proposed to handle different situations, including manifold or non-manifold (Brisson 1993; Françon and Bertrand 2000; Masuda 1993; Weiler 1988a, b). Since the 3D parcels are associ-

ated with each other or sets, and are not isolated 3D objects, we wish to build a 3D environment corresponding to 3D volumetric objects and the sets of them with topological relationships compatible with 2D parcels.

3 3D Topology-Based Cadastral Model

A full topological model for the 3D cadastre, land planning and management is needed for the following reasons: (1) to utilize the surveying boundaries to generate the 3D cadastral objects (we use the term “volumetric model” geometrically and topologically); (2) to represent the 3D volumetric objects with high quality, and consistent topology without intersection; and (3) for the rapid topological queries necessary for real-time user interaction and management.

A comprehensive land administration model is essential to build the cadastral management system. The LADM (Land Administration Domain Model) provides a conceptual description for a land administration system, including a 3D topology spatial profile (Thompson and Van Oosterom 2011). Here we only focus on the 3D space of a spatial unit based on a 3D topological spatial representation and for other details refer to ISO 19152 (Thompson and Van Oosterom 2011). In terms of the 3D topology representation, a 3D boundary face has positive/negative information included in the association with a 3D spatial unit to indicate the orientation of the face. However the volume is not indicated. How to construct the 3D spatial unit and establish the topological relationships is the main topic of this article.

According to the requirements of the LADM, considering topological information alone is not sufficient to describe a 3D object. Geometrical information must also be associated with each topological primitive. So our model needs to describe and store the geometric and topological relationships of 3D cadastral units, as well as the entities, as Figure 3 shows. Two feature objects as 3D cadastral objects, i.e. 3D land parcel or 3D legal spatial building unit, are defined, and represented geometrically by a 3D volume based on boundary representation.

This data model is oriented towards the application and storage of a 3D cadastral system. Especially, we extend the geometric-topological model in LADM, which is based on ISO 19107, and redesign the model to support non-manifold 3D objects to represent realistic 3D cadastral objects. The crucial intermediate step is how to construct the valid volumetric model (details in Section 4), which needs additional topological relationships and recognitions, such as the *loop* and *hole* (terms we will define later on). It is noteworthy that no hole or loop information is needed in the topological model because a hole or loop in 3D space is not consistent with one in 2D space. The normal vector of the face comes into effect only when the face is used to construct the volumetric model in 3D space.

As Figure 3 shows, the real geometric coordinates are only recorded in *Node*, and other topological primitives (*Edge*, *Face*, and *Volume*) will use the references to describe the relationships. An *edge* meets other edges only at *node*. A *face* is a polygon enclosed by the *edges*, and the face may have other inner rings, but information about the hole/ring is not known at the beginning. The topological primitives are hierarchically interrelated, and a (n-1)-dimensional geometric primitive is used as the boundary of n-dimensional geometric primitives (Pigot 1991). This principle of hierarchical relationship of topological primitives can ensure the consistency and reduce the redundancy of spatial data. As far as the valid volume is concerned, and assuming that the basic topological relationships among node, edge, face had been built, the key points can focus on how to:

1. Organize the faces to construct and generate the *Volume*;

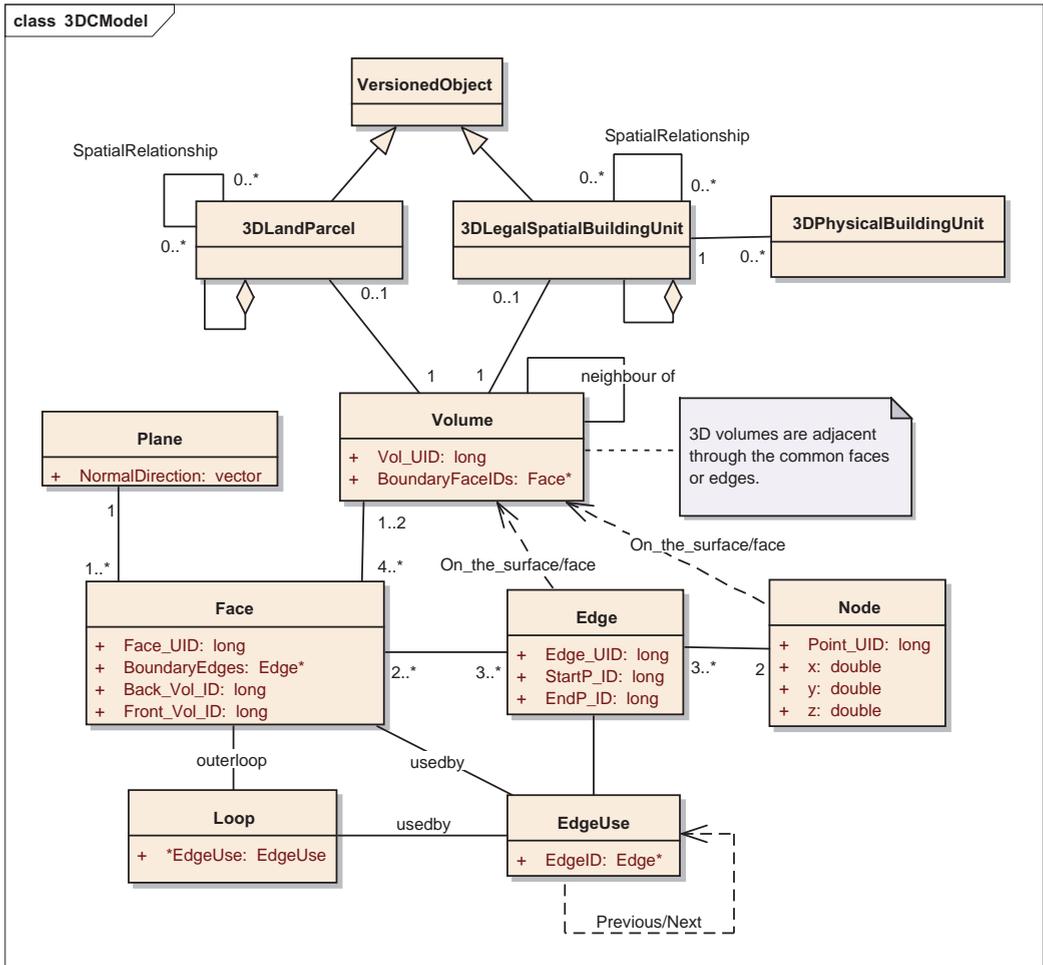


Figure 3 Data model in the prototype system

2. Handle the topological relationship between *Volume* and *Face* with the corresponding normal vector; and
3. Handle the relationships among the *volumes*.

A valid volume is made up of and closed by at least four faces with their normal directions. *Class Plane* is designed to emphasize the face’s normal direction, which means that every face used in the body is only a half-plane face.

A 3D *volume* is a 3D primitive to describe the volumetric object and is basically incident to *faces*, the lower dimensional 2D geometric primitive. We define the volumetric model as a seamless 3D space with interior orientation, and commonly its shells which, closed and made up of the faces, together completely separate the interior and exterior of the volume; volumes cannot intersect and penetrate mutually. The basic requirements and initial conditions of the volumes will be elaborated in Section 4.1.1.

An important condition of *Face* is that its normal direction points outward or inward to the *volume*, which is essential for volume construction. The face’s normal direction determines

the interior orientation of the 3D volume, and *Class Face* is an oriented facet or patch with one outer loop, and zero or more inner loops. In general, the term face denotes a simple flat face that is used to define a part of the boundary.

It is noteworthy that whether the inner loop is a *hole* depends completely on the orientation of the prospective volume. For example, there are two incident volumes touching each other at the middle laminar faces, as Figure 4a shows, and the faces they share or intersect include three faces, f_1 , f_2 and f_3 (see Figure 4b). Topologically, in the 2D view, there is a face f_1 with two inner hole faces f_2 and f_3 . However, in the 3D primitive view, the situation is different. When two 3D volumes are constructed, the system should maintain consistent topological relationships between two volumes. In terms of the above primitive volume, there should be only one face f_1 delimited by one outer loop oriented upward and two inner loops that indicate the heterogeneous holes, f_2 and f_3 , as Figure 4c shows. For the lower volume, there should be a union of the three homogeneous faces f_1 , f_2 and f_3 , all with a downward face normal direction, as described in Figure 4d.

The *hole* concept in 2D is quite different from that in 3D. On the one hand, 2D hole is based on 2D face while 3D hole is based on 3D volume. For example, f_2 can be a hole in f_1 in 2D (Figure 4b) and two 3D holes exist in the top volume in Figure 4a. On the other hand, “half-plane” or side of the face should be taken into consideration when a 2D hole is involved in a 3D hole, and in the 3D environment, a 2D hole is only oriented to one 3D volume in one side, not both sides.

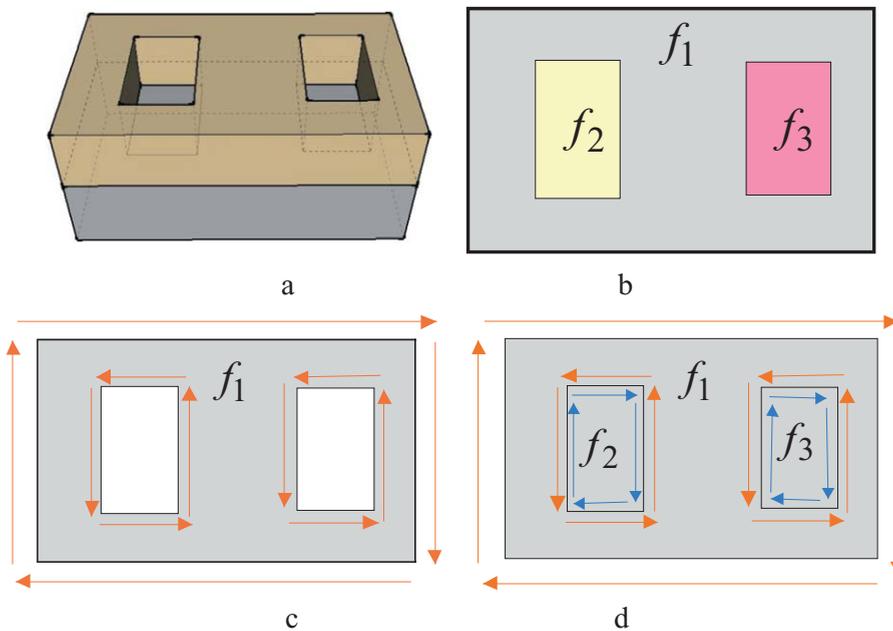


Figure 4 Face with inner holes and its different roles in different volumes: (a) two incident volumes; (b) common faces; (c) topological organization for the upper volume; and (d) topological organization for the lower volume

4 Solid Construction Based on Face-Sorting and Face-Connecting

4.1 Solid Construction

In this section we specify in details how to construct the solids from a set of input candidate faces. We construct the solids according to the following steps: (1) the input faces incident to a given edge are sorted around that edge, as well as the node related to the edges; (2) the next-neighbor face around that edge in the order based on the dihedral angles is determined and connected; (3) the result is stored in the topological data structure; and (4) solids that are deemed valid according to our definition are extracted by navigation in the data structure.

4.1.1 Initial conditions and constraints

Here we suppose that the precise 3D data must correctly describe the discrete faces. This also can be guaranteed by the surveyors in 3D cadastre. The input loose faces must respect certain criteria for the algorithm to construct solids and properly terminate. Basic geometric and numeric pre-processing is thus required first to deal with the raw data. For example, certain tolerances and thresholds are set to snap the points, and pre-processing is carried out to ensure there are no self-intersection and no dangling nodes and edges before solid construction. Furthermore, initial discrete sets of surfaces in 3D space should be divided and segmented entirely. Van Oosterom et al. (2004) described the definition and validity of the planar polygon. Chen et al. (2008) and Chandra et al. (2008) emphasized the simplicity and used the direction of the polygon when they discussed the constraints of the polyhedron construction, during which the certainty of the polygon direction would determine the solid. In contrast to these researchers, our approach does not use the orientation of the polygon as input, but the orientation is computed during the solid construction (potentially two solids for every polygon). Here, the basic constraints and conditions in the input step before constructing the valid solids are as follows:

1. Each polygon is closed (same starting and ending point).
2. Lines and polygons must maximally intersect, and full subdivision of surfaces in planar faces is necessary.
3. Each face is a flat polygon, and curved faces should be divided into several flat faces.
4. No isolated, hanging or dangling point and edge is allowed.
5. From a geometric point of view, faces may have inner face(s)/ 2D hole, but whether there exists a 3D hole depends on how the solid is constructed (explained in Section 3).

The aim of the initial conditions is to keep the incoming 2D faces ‘clean’. Figure 5 depicts the isolated edges interior (d in Figure 5a) or exterior (a in Figure 5a) of the polygon, and a hanging/dangling edge or line-edge in the boundary of the polygon edge (b and c in Figure 5a), as well as a redundant edge (e in Figure 5b). They should all be deleted. There should be two edges f and g incident to two faces f_3 and f_4 in Figure 5c, respectively. In other words, each edge should be related to the boundary of a certain polygon, but its orientation is not vital because of the recalculation of the face’s normal direction in 3D space during solid construction. Moreover, since the concept of the hole will be reconstructed in 3D space, the interior holes should be represented correctly in the geometry.

The above scenarios list the normal constraints for 2D input, which is different in 3D space. Hanging and dangling edges, or a redundant edge, may be used by other faces with different plane direction in 3D space. From the view of full subdivision and segmentation, the

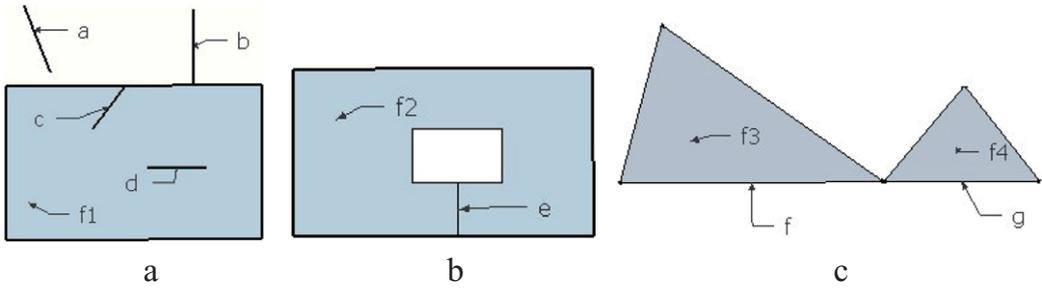


Figure 5 Hanging and dangling edges (a), redundant edge (b) and line segmentation (c)

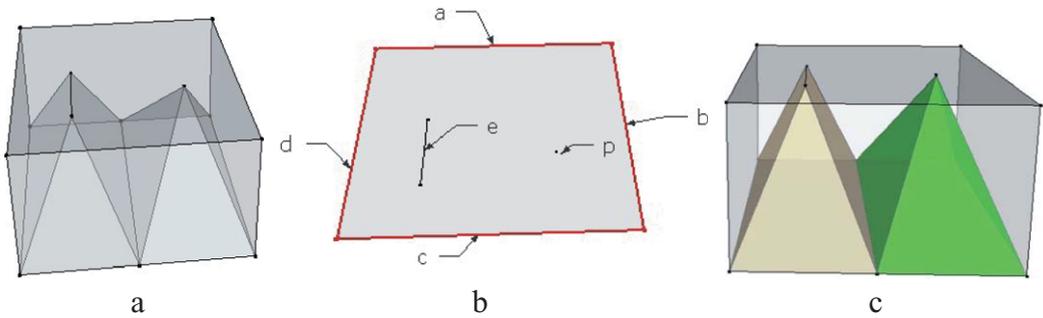


Figure 6 Maximal subdivision and segmentation in 3D space: (a) input faces in 3D space; (b) topological edges for the top face; and (c) three constructed solids

shape in Figure 6, should result in five edges (a, b, c, d, e) and one point (p). When we construct the top face, four edges (a, b, c, d) are used as the boundary, and the location relationship that edge e and point p are ON the face which is potentially known, as it can be calculated from the geometric location (Figure 6b). But for the top face, there is no topological information about edge e and point p . At the same time, edge e and point p are used in other faces with different plane directions. The resulting solids should include three solids (Figure 6c). The data model in Figure 3 supports these situations. From the locational point of view, edge e and point p bounding two below solids locate the same plane related to the top face bounding above solid (Figure 6c). From the topological perspective, edge e and point p have no relationships with above solid.

It is worth noticing that all the above constraints and conditions apply not only to 2D space but also to 3D space. If two faces intersect with each other in 3D space, they should be subdivided and partitioned at the common edge, or even point (as in Figure 7). This means the faces in 3D space can share an edge, but cannot intersect with each other. Maximal subdivision and segmentation of polygons should be carried out in 3D space. For example, two polygons ABC and $BEFG$ in Figure 7 intersect and a new edge BD should be created to segment the two polygons into four polygons: ABD, BCD, BED and $BDFG$. Otherwise, an unclear subdivision may lead to improper and illogical topological results. The two cuboids in Figure 7b can be constructed individually, but they “share” an intersection volume part, which conflicts with the concept of mutual exclusion and space subdivision. They should be deemed

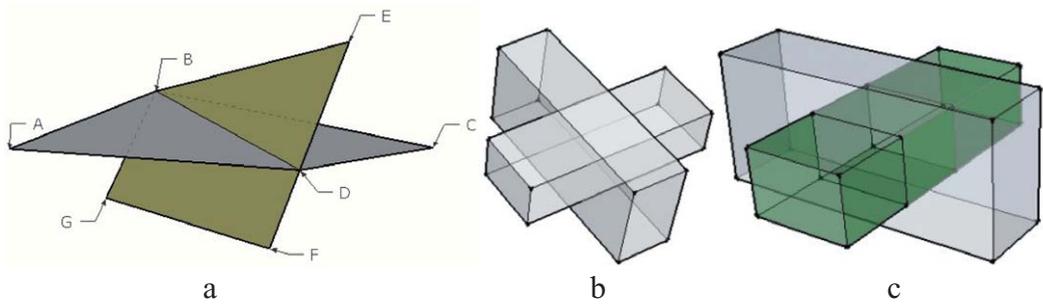


Figure 7 Maximal subdivision of polygons into flat faces

to intersect, and if the input faces are correctly segmented, this situation can be solved, as Figure 7c shows. In a sense, the basic condition refers to the maximal subdivision of flat faces. In short, all inputs are only simple faces.

Although a complete geometric description contains all the information about the shape, it is better to have both geometric and topological information in the initial conditions to facilitate the construction of solids. Moreover, research has proven that the automatic construction of 2D topological relationships based on incidence among points, edges and faces are already realized (Patel et al. 2005; Rock and Wozny 1992; Yan et al. 2000). Existing tools and software, such as Esri's ArcGIS and Trimble's SketchUp, can be used to establish 2D topological relationships. Therefore we assume that the topological relationships among node, edge, and face in 2D have already been built. Many data structures, such as the half-plane, wire-edge and radial face, can be used to describe the given faces (Thompson and Van Oosterom 2011; Weiler 1988a).

Besides the maximal face subdivision, another important view concerns the certainty of the given faces. It is well known that wire frame models based on the given vertices and edges do not yield unique solids. In 3D environments, we cannot build the faces dependent on the edges because, when a new edge is added, we do not know where the new face is or how we can create it exactly.

Before the detailed automatic construction procedures of valid solids, all faces should be certainly determined. That means that new faces are not allowed to be created, and the existing data are not allowed to be modified. In this article, solid only refers to the *minimum solid*. Moreover, this designation of minimum solid in 3D space is necessary to guarantee the uniqueness of the constructed object; otherwise there may be different results.

Within the above specifications of the constraints and conditions of basic geometries and topologies, the focus is on the process of constructing the valid solid. The next subsection elaborates the procedures of face-sorting based on shared edge and face-connecting to construct the valid solids. This is different from Euler's operators (Masuda 1993; Boguslawski and Gold 2010) that are used to generate a single solid without focusing on geometric-topological consistency. As a consequence they cannot build the polyhedrons with holes or 3D connected non-manifolds, and they do not provide the conditions for selecting the next geometric primitive if there is more than one.

4.1.2 Sorting faces around the shared edge

The topological relationship between the faces can be built based on the shared incident edge. The degree of the edge is defined according to the number of the faces incident to this edge,

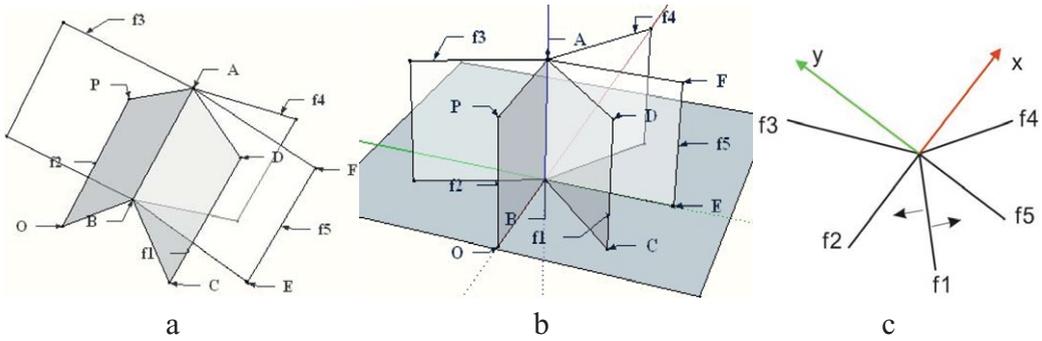


Figure 8 Face-sorting based on the shared edge

which potentially expresses the number of the faces along one edge side. Degree 1 indicates that the related face is hanging or dangling in 3D space, and Degree 2 means the two related faces may connect with each other at the “corner” of this common edge. The problem is how to deal with the faces radiating from the edge with a degree larger than 2. Face-sorting based on the shared edge is an effective way for handling this situation. Face-sorting around the shared edge is the foundation of the automatic construction of solids because it provides the ordering information about the faces needed to select and connect the successive faces.

In fact, each face incident to an edge exists in the half-plane and is delimited by the shared edge as Figure 8a shows. assuming we use a plane perpendicular to the common edge to cut the sheaf of faces (Figure 8b), the faces radiating from this common edge in 3D space become the radial-line beam in a 2D plane sharing one node (Figure 8c). We can sort the radial-lines according to the angles between them in a clockwise (or counter-clockwise) direction. The sequence in Figure 8c is $\{f1, f2, f3, f4, f5\}$ in a clockwise direction. Actually, the angles and the location relationships of the radial edges can be calculated by the dihedral angles of the bundle of faces. With the dihedral angles and the sorted sequence, for each face there exist only two candidate faces that may be used during the construction of a valid solid. For $f1$ in Figure 8c, the successive face on the left is $f2$ and on the right is $f5$. Furthermore, the face $f5$ is the farthest left face. This side view is based on the transformed radial-line beam for the faces in 3D space. This idea can be further realized and confirmed by the incidents and topological relationships among the faces in the bundle.

4.1.3 Determining and connecting the next-neighbor face around the shared edge

The crucial step of solid construction is how to determine the next connected face. The dihedral sorting sequence shown in Figure 8 does not find the successive neighbor face. The successive neighbor face should be incident with the current face, and connectivity between two neighbored faces is based on the topological incidence.

As mentioned above, the normal direction of the discrete face has no meaning without reference to the solid. Only when we construct valid solids, the normal direction of a face can be involved. If we attempt to construct the solid based on two candidate faces created by the face-sorting sequence, the complexity of the solid construction will be raised to 2^n (n is the number of shared edges) since there are two different choices for the next face at every shared edge. However, the topological incident between two neighbored faces can reduce this complexity to a single choice of the successive face during the solid construction.

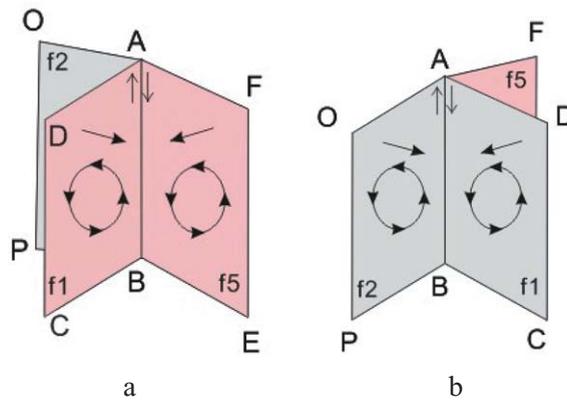


Figure 9 Determination of the next-neighbor face with the transitivity of the normal direction

Similar to the search and construction of planar polygons, the two opposite sides of the face should be distinguished and, in terms of half plane, the face's normal direction is used to determine the next-neighbor face. A next-neighbor face is defined as the face with the minimal angle to the denoted face around the common edge. At the beginning of construction, a random face and its normal direction are chosen as starting points. The designated face and its normal direction will define the direction of the face-sorting sequence and the direction of the candidate neighbor faces. This determines the unique candidate for a potentially minimum valid solid if it exists. Once we have selected the face $f1$ with its one normal direction oriented right as Figure 9a depicts, the boundary of face $f1$ should be organized counter-clockwise. The oriented boundaries of face $f1$ are organized as $BADCB$ and the shared boundary edge AB has the direction from B to A . Because of the topological incidence, the direction of the boundary edge AB in the next neighbor face $f5$ should be from A to B . At the same time, according to the face-sorting sequence and the given normal direction of $f1$, face $f5$ is selected as the next face. So the next-neighbor face we get during solid-construction should be $f5$ and the boundary edges should be organized counter clockwise as $ABEFA$, as Figure 9a shows. Following the same principle, if we select another normal direction of $f1$ (organized as $ABCDA$) as the starting point the next face will be $f2$ with clockwise boundary edges (Figure 9b). This transitivity of normal direction and the next-neighbor plays a vital role in solid construction.

4.1.4 Construction of the solids

In this article, we focus on the *minimum solid* based on the above face-sorting sequence and the connectivity with the next-neighbor face. Although the initial faces with certain normal direction and connectivity criterion are specified, there are still different ways to construct the valid solids that are unique. Every face has several boundary edges, and each edge may connect with other potential faces. There are two ways to search and connect the successive faces: depth-first search (DFS) and breadth-first search (BFS), which will create the same unique solid. Both methods make it possible to progress from one face to its neighbor face to get the topological relationship. The difference between them is that, during the construction process, DFS search will form a strip zone while BFS search will form a surrounding shape.

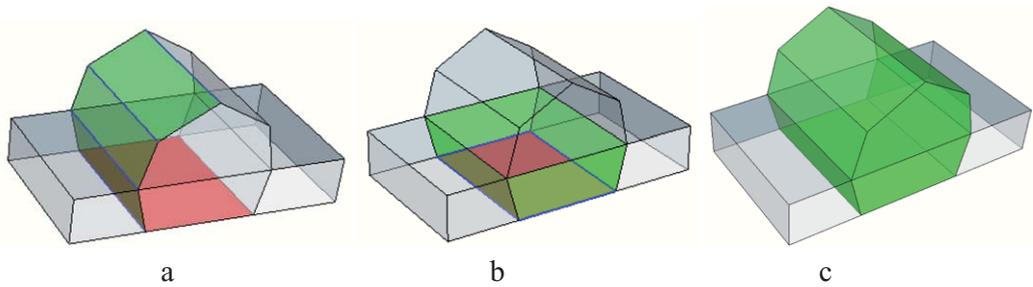


Figure 10 Different shapes during the process: (a) the strip zone based on DFS; (b) the surrounding shape based on BFS; and (c) resulting solid

Figure 10a illustrates the shapes of a strip zone based on DFS. Starting with the mid-bottom face, in each iteration only one edge is chosen as the encircling edge (blue color). To connect the next face, after three iterations the green faces may be one possible strip zone during construction based on DFS. As for BFS, each edge (blue color) of the starting mid-bottom face will be searched at the first iteration, and the face sequence is formed surrounding the starting mid-bottom face circularly (Figure 10b). But neither the DFS nor BFS method can affect the uniqueness of the result with the same start face, as the constructed solid is exactly the same (Figure 10c).

Taking all the given faces in 3D space into consideration, we should detect the flag and eliminate the faces that will not be involved during the solid construction. Thus, before constructing the solid, the degree of each edge is calculated to judge whether the incident face is a hanging or dangling one (as the flowchart in Figure 11 shows). If true, this face will be excluded later. The result of the elimination process of the faces is that the remaining faces will be involved in solids via face-sorting and face-connecting. In such cases, the tool developed with this algorithm can highlight them and warn the user to correct the spatial data in 3D space when necessary.

With all the sorted faces around shared edges, the flowchart of constructing valid solids based on BSF can be described as shown in Figure 11. For any individual face f_i and its ONE side, create a face array representing the resulting solid to store the face's ID and direction. Then flag face f_i traverses each edge of face f_i ; check whether it is visited. If not, flag this edge and check the edge degree dg to determine the next face f_{i+1} according to the given face f_i and its direction n_{il} . If $dg > 2$, then determine the next-neighbor face f_{i+1} based on face-sorting; if $dg = 2$, then select another face. In addition, we can store the edge's ID if we want to record the relationship between edge and solid explicitly. The method will shift to the face f_{i+1} until all the faces with dual directions are tested.

It can be inferred that the stop condition of the constructing workflow is that all the faces are tested and participate directly in the solid construction. Because of the detecting and removing of the hanging and dangling faces, the approach can identify separate isolated solids and solids set with the edge's face degree in step 3. Edges with degree 3 or more indicate that they are a solid-coherent set and the solids are adjacent, i.e. sharing this edge or face. Face-connecting in this context is based on the face-sorting.

It should be noted that during the construction of valid solids, the step of searching and determining the next-neighbor face can be transited to the next face with half-plane, and the approach dominantly records the direction of the faces and the interior of the solid. The topological relationships between solid and face are stored, as well as the shared edge. It is easy to

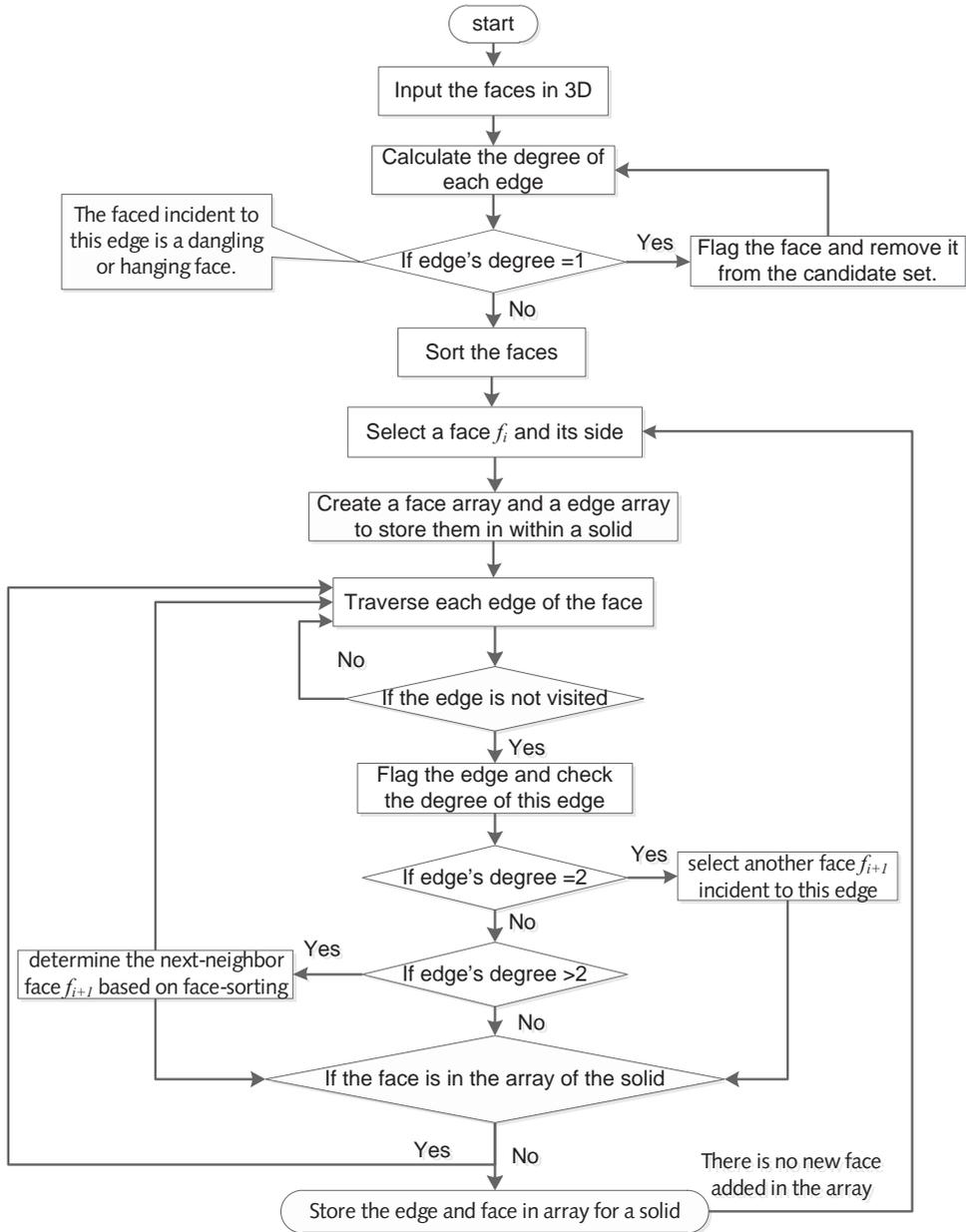


Figure 11 Flowchart of solid construction

find the neighbor solids with the common face/edge and the topological relationships among solids. So the contents and relationships in the data model in Section 3 are implemented and filled in, which can keep the consistency of 3D solids and reduce the redundancy of spatial data effectively.

4.2 Discussion

The 3D volumetric objects created by the above method support both normal 3D solids and non-manifold solids (shapes with self-touching or hole) along with topological relationships that are already valid. Also the 3D volumetric objects in relation to the outer complementary space (named by Maximal Minimal Solid) can be generated, as discussed in the following subsections.

4.2.1 Validation

The approach presented in this article ensures volumetric objects (polyhedral shapes) that satisfy the valid solid characteristics: face-based construction, closeness and uniqueness. First of all, it is evident that a solid is only made up of simple faces. Considering the side duality of geometric faces, the approach ends when all the input faces have been involved in solid construction. This means that every topological edge and face are used twice or even several times during the procedure of solid construction. This assures that there is a path from one face to another, and manifests from a topological view, that the constructed solids are valid and closed. Furthermore, after the processing of solid construction, there are no other faces between two neighbor faces around the shared edge; the unique next face is connected; and, the transitivity of the unique face-connecting process results in the unique polyhedron. In addition, the full subdivision and intersection of the input faces drive the searching process which turns around the common edge, and ensures that no face or other primitive is contained within the solid. This means that there is an interior path from one point on the face to another point on any other face without intersection with any other primitive within this solid.

4.2.2 Non-manifold volumetric objects or singularity shapes can be represented efficiently

Things get even more complicated for a coherent set of 3D solids sharing topological primitives (together forming a space partition) and the various singularities that may be the result. The constructive principle, that a 2D geometric primitive (face) is used as the boundary of a 3D geometric primitive and the hierarchical interrelations between face and solid, supports the singularity of the cross-dimensional/inter-dimensional geometric primitive (edge or point) in a more consistent and elegant way. On the one hand, the algorithm breaks the principles of manifold and simplicity. The non-3-manifold polyhedron in Figure 2, touching itself with a node (Figure 2a) and an edge (Figure 2b), is still a valid solid with an interior-connective 3D space supported by the model in Section 3 and Figure 3. This is important to model some realistic cadastral objects (see Figure 1). On the other hand, the procedures of solid construction that search and detect the faces correctly support and keep the consistent relationships among the resulting coherent solids. Each volume in Figure 12 is created correctly and at the same time, the relationships between them, sharing a node (Figures 12a-b), an edge (Figures 12c-d) and a face (Figures 12e-f), are built and recorded precisely; even the node of one solid is on the surface of another solid (Figure 12e). We should clearly be aware that the shared geometric primitives (point, edge and face) may not only be single features. They can be associated and combined together, for example, two solids sharing two edges (Figure 12g) and two solids sharing both one point, one edge and one face (Figure 12h).

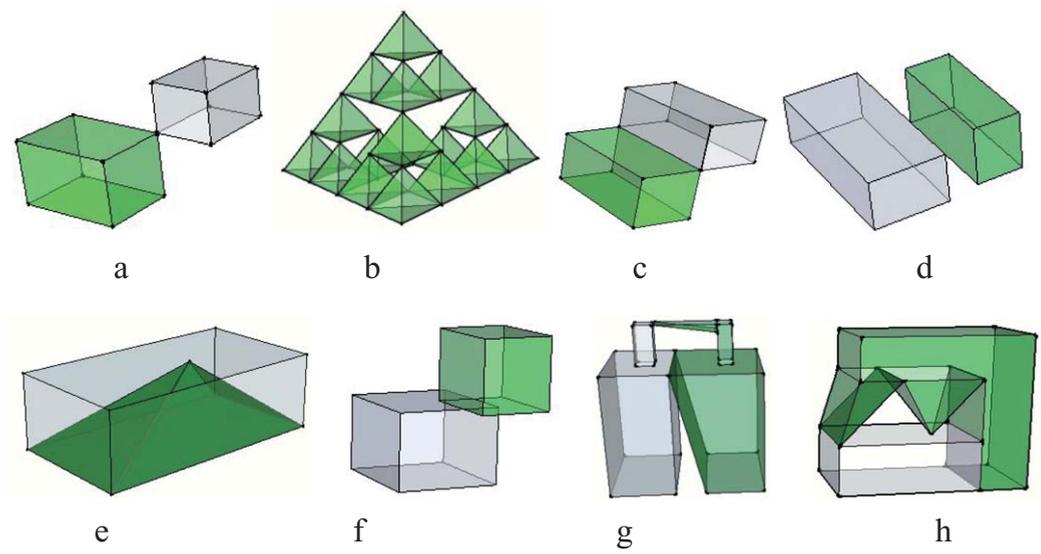


Figure 12 Solids sharing the node, edge or face

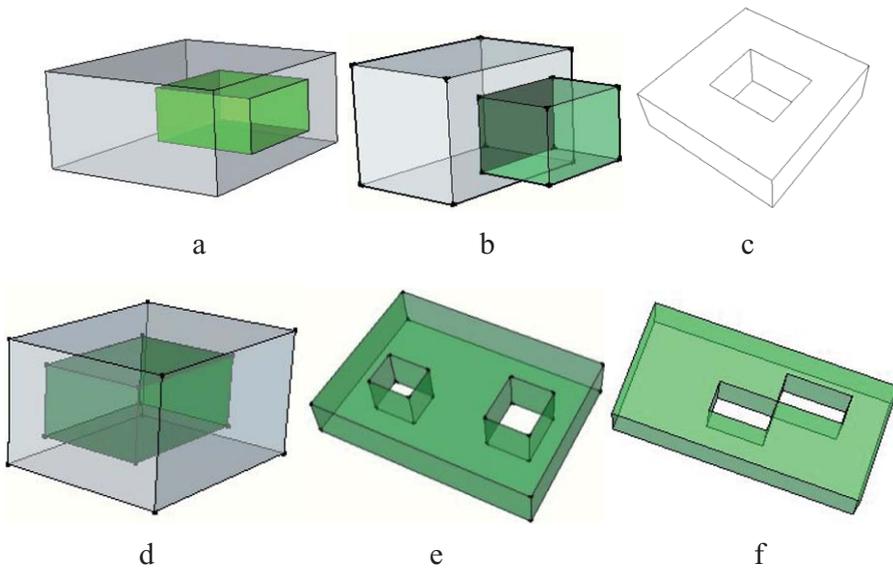


Figure 13 Solids with through-hole/holes: dent solid (a) and cave solid (d), through holes (c, e, f)

4.2.3 The approach supports solids that are composed of faces with 2D holes

In the case of the solids shown in Figure 13, the orientations of faces are detected and two solids are generated with a scenario similar to that shown in Figure 4: in Figure 13a the smaller solid is excluded by the outer bigger one and the side face is excluded by the outer solid; while the face in Figure 13b is used for both solids with different normal directions. The

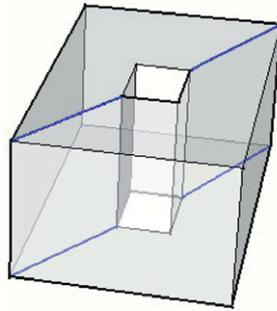


Figure 14 Existing method of validation: re-verification by adding a fictive edge (adapted from Horna et al. 2009; Kothuri et al. 2007)

solid with a 3D through-hole or torus (Figure 13c) can be constructed reasonably because the approach can determine the orientations of the faces and connects them automatically to form the corresponding shapes. However, for the situation in Figure 13, two solids (outer solid and cave inside) can be created separately, but it needs further geometric computation to determine the inclusion/contain relationship between them, and their heterogeneity needs the user's definition. For the above various singularities, the algorithm can successfully recognize and construct the corresponding solid respectively, which go beyond the afterwards validity check in the existing literature (Arens et al. 2005; Kazar et al. 2008; Van Oosterom et al. 2004). At the same time, in many existing validation check methods like in Oracle and CGAL (Horna et al. 2009; Kazar et al. 2008; Kothuri et al. 2007), for solids composed of faces with holes or rings, each holed face is processed in its 2D plan and a fictitious edge is added if necessary according to the definition of a 2-manifold (Figure 14). This leads to the following questions: (1) by which edges is the new face created? and (2) how does the face relate to this edge? If the faces are uncertain, the potential solids are uncertain. This edge-adding operation modifies the source data which is not complying with the consistency of the data.

4.2.4 Maximal minimum solid

Generally, there are two minimum solids according to the two-sided direction of the same face within one solid set. With respect to the outer space, the "minimum solid" is the Maximal Minimum Solid (MMS) because this solid is also the minimum solid of the complementary space with a similar concept. But the MMS contain all the coherent aggregated minimum solids and the counterpart of MMS is the outer shell of the solid set; and it is also valid and can be constructed. The normal direction of each face of the MMS is oriented to the complementary space in 3D space. An efficient way to test or identify the MMS is related to the maximal or minimal outer boundary faces and works via the following steps. First, select the outer face and test one of its normal directions; if its face normal vector intersects with other faces, then move to another normal direction. With the outer faces and its outer normal direction, the algorithm can construct the minimum solid with the outer orientation. Remarkably, the MMS is also unique in the 3D environment based on the given faces: and as for the single isolated solid, the minimum solid and the MMS have the same geometric polygons, just with different normal directions. In practical applications, for each isolated solid or separate solid set, we should first exclude the MMS to avoid repetition.

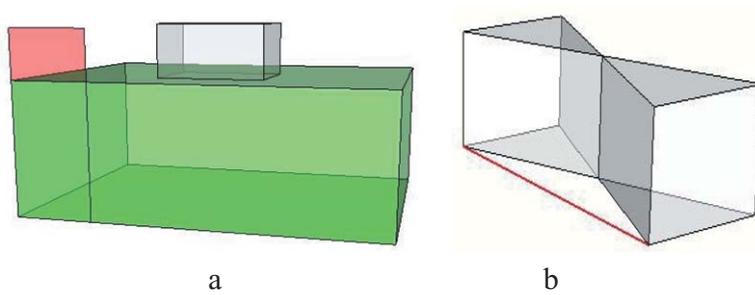


Figure 15 Data checking and warnings of the errors of dangling face (a) and isolated edge (b)

4.2.5 Open and mixed data

In addition, consider that the 3D cadastral system must be compatible with the 2D situation because there are 2D land parcels and 3D parcels that exist simultaneously. In a practical system, we implement the functions to support the open and mixed-dimensional spatial data, to detect the hanging and dangling lines or faces with convenient tools, like the red face in Figure 15a and the red line in Figure 15b. This tool is also used for the pre-processing in Section 4.1.

5 Test Cases

We implement the algorithm and build up the prototype system based on SketchUp (Version 8.0), Ruby API (Version 1.8.6) and Oracle 10g (not Oracle Spatial) to demonstrate its feasibility. Initial input is 3D data (collection of faces) created by other tools or software, such as a survey on extrusion (Ledoux and Meijers 2011; Ying et al. 2011). They can be organized based on the topological model in SketchUp. Each object in SketchUp is made up of edges and faces, and SketchUp Ruby APIs provide a basic topological model between the class *Edge* and *Face*. An automatic construction of the valid cadastral volumetric objects is realized and spatial relationships stored into the database, with the help of Ruby programs based on SketchUp.

Here, in order to assess the overall approach, we use valid solid for 3D cadastral volumetric objects and their sets, which is a true 3D GIS implementation. Each 3D cadastral volumetric object is represented by a valid 3D solid with a certain closed 3D space, whether it is on the Earth's surface, above it or underground. Moreover, cadastral boundary points and lines are measured by the surveyors, specified by the legal system and recorded according to certain specifications and rules, which satisfy the constraints and conditions that have been stated in Section 4.1.1. Based on the approach, we implement a practical 3D cadastral system to manage 3D cadastral objects, especially property units, which are normally in aggregated and associated states, and the non-manifold buildings. We need to represent the 3D cadastral object of the real world, whether they are a disjoint volumetric object or its set. Figure 16a shows the China Central TV building with an irregular shape, non-vertical faces and a twisted ring. Our prototype system builds the irregular 3D volumetric object (blue in Figure 16a) reasonably, with 21 flat faces, compatible with the 2D land parcel (brown face in Figure 16a) in different colors.

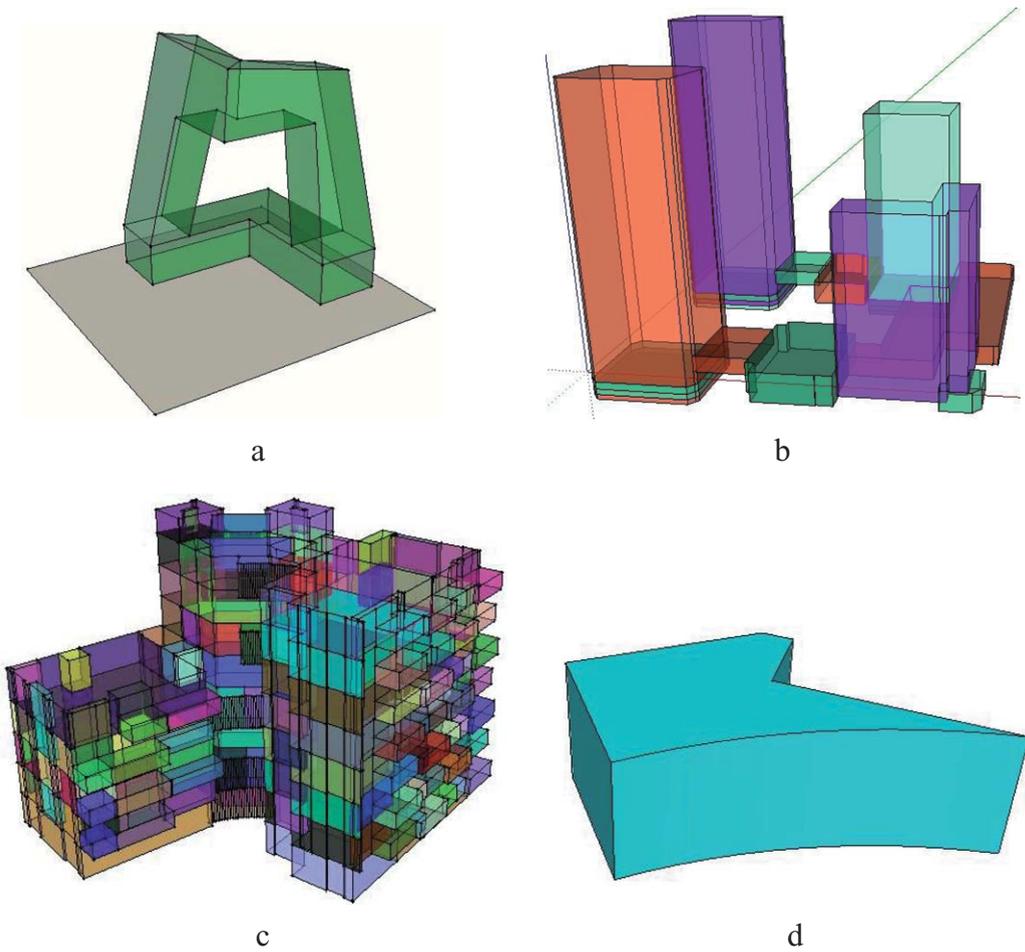


Figure 16 Construction of the volumetric object and its set: (a) CCTV building (an irregular polyhedron with a 2D parcel); (b) each volumetric object in podium building; (c) property unit set of one residential building; and (d) a volumetric object with curve surface

The challenges and advantages of our algorithm are to correctly search and build each 3D volumetric object respectively from/for its coherent set. A complex podium building, including the podium buildings, four skyscrapers, three underground floors, and three over-bridge arcades (Figure 16b) is represented by 245 faces, and the corresponding set of volumetric objects composed of 20 solids are examined and approved at the first phase during urban planning. Figure 16c manifests one residential building with its apartments. Each volume of the property unit/apartment with the sharing boundary edges and faces is confirmed and represented by one valid volumetric object with different colors according to their different orientations and their faces' normal direction in the solid set environment. Based on our algorithm, there are a total of 252 valid solids derived by 2,976 faces, among which 1,164 faces are used to represent 12 curve "surfaces". For example, 26 flat faces depict one "curve surface" in a solid as Figure 16d shows. Of course, all the cadastral semantics and attributes can be associated with these 3D cadastral objects represented by the valid volumetric objects, which is not

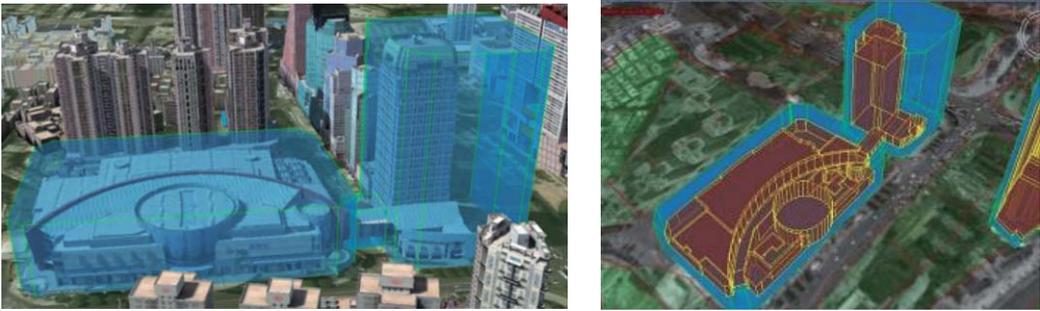


Figure 17 3D legal spatial building space and 3D physical building space

discussed in this article. The developed algorithms are applied to support various polyhedrons to represent the 3D land parcels in urban planning and management in Shenzhen, China (Van Oosterom 2013). For example, a 3D physical building space, owned by a property developer, consists of one shopping center, one office premises and a cross-over-street arch; and this 3D physical building space is kept within the bounds of the 3D legal spatial building space (3D cadastral space) shown in Figure 17.

6 Conclusions

In this article, we presented a 3D topology, which can be implemented in cadastral applications. The approach in this article can efficiently recognize and construct solids and represent the topological relationships of valid volumetric solids in 3D space in a straightforward way, whether there is an isolated solid, or a multi-solid/solid set. Moreover, the internal structure and spatial relationships of a complex volume set are preserved by maintaining each volume under valid control. In practical applications these manage and register the 3D parcels. This is a completely different approach from performing a validity check afterwards. The input faces themselves are stable and they are independently specified. This direct 3D volume construction conforms to normal sequential data flow and business logic to provide valid 3D volumetric objects for 3D cadastral systems without the need for a post production validity check. The algorithm is capable of supporting various 3D shapes and non-manifold volumetric objects with holes or caves, and causes no problems with regard to the topological consistency. Real 3D volumetric objects are constructed first with the input faces, storing the references in the 3D topological model. At present, the attention in the context of 3D cadastre is concentrated on the geometries of the 3D cadastral objects to model the representation of topological relationships. The main findings in this article can be utilized in city models with certain semantics to support volumetric object modeling.

Note

- 1 Post-construction validation: current mainstream methods require one to assume that the shapes (solids) already exist in the 3D object and then design a test to see if this existence assumption holds. In the method we present, this assumption step is no longer required as a necessary research process.

References

- Arens C, Stoter J, and Van Oosterom P 2005 Modelling 3D spatial objects in a geo-DBMS using a 3D primitive. *Computers and Geosciences* 31: 165–77
- Boguslawski P and Gold C 2010 Euler operators and navigation of multi-shell building models. In Neutens T and De Maeyer P (eds) *Developments in 3D Geo-information Sciences*. Berlin, Springer Lecture Notes in Geoinformation and Cartography: 1–16
- Brisson E 1993 Representing geometric structures in d-dimensions: Topology and order. *Discrete and Computational Geometry* 9: 387–426
- Brugman B, Tijssen T, and Van Oosterom P 2011 Validating a 3D topological structure of a 3D space partition. In Geertman S, Reinhardt W, and Toppen F (eds) *Advancing Geoinformation Science for a Changing World*. Berlin, Springer-Verlag Lecture Notes in Geoinformation and Cartography: 359–78
- CGAL 2011 Computational Geometry Algorithms Library. WWW document, <http://www.cgal.org>
- Chandra N S and Govardhan A 2008 Design and implementation of polyhedron as a primitive to represent 3D spatial objects. *Journal of Theoretical and Applied Information Technology* 4: 212–18
- Chen T K, Abdul-Rahman A, and Zlatanova S 2008 3D solids and their management in DBMS. In Van Oosterom P, Zlatanova S, Penninga F, and Fendel E (eds) *Advances in 3D GeoInformation Systems*. Berlin, Springer-Verlag: 279–311
- Françon J and Bertrand Y 2000 Topological 3D-manifolds: A statistical study of the cells. *Theoretical Computer Science* 234: 233–54
- Horna S, Meneveaux D, Damiand G, and Bertrand Y 2009 Consistency constraints and 3D building reconstruction. *Computer-Aided Design* 41: 13–27
- Kazar B M, Kothuri R, Van Oosterom P, and Ravada S 2008 On valid and invalid three-dimensional geometries. In Van Oosterom P, Zlatanova S, Penninga F, and Fendel E (eds) *Advances in 3D GeoInformation Systems*. Berlin, Springer: 19–46
- Kothuri R V, Godfrind A, and Beinat E 2007 *Pro Oracle Spatial for Oracle Database 11g*. New York, Springer-Verlag
- LADM 2012 ISO/TC 211 Geographic information – Land Administration Domain Model (LADM), ISO 19152. WWW document, <http://www.gdmc.nl/oosterom/>
- Ledoux H and Meijers M 2011 Topologically consistent 3D city models obtained by extrusion. *International Journal of Geographical Information Science* 25: 557–74
- Lee S H and Lee K 2001 Partial entity structure compact non-manifold boundary representation based on partial topological entities. *Journal of Computing and Information Science in Engineering* 1: 356–65
- Masuda H 1993 Topological operators and Boolean operations for complex-based non-manifold geometric models. *Computer-Aided Design* 25: 119–29
- Patel P S, Marcum D L, and Remotigue M G 2005 Stitching and filling: Creating conformal faceted geometry. In *Proceedings of Fourteenth International Meshing Roundtable*, San Diego, California: 239–56
- Pigot S 1991 Topological models for 3D spatial information systems. In *Proceedings of Autocarto 10*, Baltimore, Maryland: 368–92
- Rock S J and Wozny M J 1992 Generating topological information from a “bucket of facets”. In *Proceedings of Solid Freeform Fabrication Symposium*, Austin, Texas: 251–9
- Stoter J E and Zlatanova S 2003 3D GIS: Where are we standing? In *Proceedings of the ISPRS Joint Workshop on Spatial, Temporal and Multi-dimensional Data Modelling and Analysis*, Québec City, Québec
- Thompson R and Van Oosterom P 2011 Modelling and validation of 3D cadastral objects. In Krek A, Rumor M, Zlatanova S, and Fendel E (eds) *Urban and Regional Data Management*. London, CRC Press: 7–23
- Van Oosterom P 2013 Research and development in 3D cadastres. *Computers, Environment and Urban Systems* 40: 1–6
- Van Oosterom P, Quak W, and Tijssen T 2004 About invalid, valid and clean polygons. In Fisher P F (ed) *Developments in Spatial Data Handling*. New York, Springer-Verlag: 1–16
- Verbree E and Si H 2008 Validation and storage of polyhedra through constrained Delaunay tetrahedralization. In Cova T J, Miller H J, Beard K, Frank A U, and Goodchild M F (eds) *GIScience 2008: Proceedings of the Fifth International Conference*. Berlin, Springer-Verlag: 354–69
- Weiler K 1988a Boundary graph operators for non-manifold geometric modeling representations. In Wozny M J, McLaughlin H W, and Encarnacao J L (eds) *Geometric Modeling for CAD Applications*. Amsterdam, The Netherlands, North-Holland Publishers: 37–66

- Weiler K 1988b The radial edge structure: A topological representation for non-manifold geometric boundary modeling. In Wozny M J, McLaughlin H W, and Encarnacao J L (eds) *Geometric Modeling for CAD Applications*. Amsterdam, The Netherlands, North-Holland Publishers: 3–36
- Yan H, Yang W, Chen Q, and Liang T 2000 A fast algorithm of topological polygon auto-construction based on azimuth calculation. *Journal of Image and Graphics* 5A: 563–7 (in Chinese)
- Ying S, Li L, and Guo R 2011 Building 3D cadastral system based on 2D surveying plans with SketchUp. *Geospatial Information Science* 14(2): 129–36