# A topological enabled three-dimensional model based on constructive solid geometry and boundary representation

**Huang Ming**[1] · **Du Yanzhu**[2] · **Zhang Jianguang**[1] · **Zhang Yong**[1]

Received: 2 July 2016 / Revised: 26 August 2016 / Accepted: 29 August 2016 / Published online: 8 September 2016

**Abstract** Ordinary triangular mesh model can be constructed from discrete point cloud. However, this kind of model contains large amount of data. It is not only difficult to split, but also lacks topological relation information. We proposed a CSG–BRep (Constructive Solid Geometry—Boundary Representation) topological model to overcome these problems. CSG–BRep model can record topological relationship of 3D model in great detail. We first introduced aspects of the topological model: location, orientation and sub-shape. Then we proposed two algorithms to access topological structure. We also proposed algorithms for performing Boolean operation on CSG–BRep models. Finally, we demonstrated CSG–BRep construction using LIDAR point cloud as a data source. We would show that, compared to ordinary triangular mesh model, CSG–BRep model is composable and can effectively reduce data volume. In addition, CSG–BRep model has detailed topological relation information, allowing further querying and analysis of 3D spatial topological information.

**Keywords** Triangular mesh · LIDAR · CSG–BRep · Topological model · Boolean operations · Spatial topological query

✉ Huang Ming
huangmingbucea@126.com

[1] Key Laboratory for Modern Urban Surveying and Mapping of National Administration of Surveying, Mapping and Geoinformation, Engineering Research Center of Representative Building and Architectural Heritage Database, The Ministry of Education, Beijing University of Civil Engineering and Architecture, No. 1, Zhanlanguan Road, , Xicheng District, Beijing 100044, P. R. China

[2] Atlas Informatics Corporation, No Academic Affiliation, 989 112th AVE NE, APT 1007, Bellevue, Washington 98004, USA

## 1 Introduction

3D spatial and topological relation is critical for GIS data model, spatial query, spatial analysis [1]. In recent years, 3D LIDAR enables fast and convenient point cloud acquisition. Current approach is to construct mesh from point cloud. This approach is good at producing detailed 3D models for visualization. However it has many drawbacks: the data volume is too large to lend be easily managed, the models also lacks topological relation data [2,3]. Therefore there is a need for a new modeling approach to accurately represent object, integrate with mesh models while reducing data volume and maintain spatial and topological relations. The topological information could also facilitate 3D spatial query.

### 1.1 Related works

There are a lot of research on 3D topological model. For example, Lin Li, Zhigang Zhao, Renzhong Guo have proposed 3D topological construction among solid objects [4], Their proposal was to first construct each individual objects then establish relationships among objects. This approach could handle topological relationship between objects. However, the relationship is limited to adjacency or nearness. Their proposal did not provide any practical solution to represent topological relation between intersecting objects.

Zhenfeng Shao, Deren Li, Chengqi Mi proposed 3D reconstruction through objects and images [5], their proposal includes concepts like topological links, internal topology, external topology. Their proposal uses connectivity and saturation to describe 3D object topology and relation among point, line and surfaces. This model can be used with stereoscopic images to reconstruct object. This approach could handle complex model reconstruction. However, the model reconstruction was too coarse to be applicable to detailed objects like historical buildings.

Changbin Wu, Guonian Lv analyzed state-of-art in the field of spatial topological modeling [6],they surveyed current research on representing and analyzing spatial topological information. They pointed out that spatial topological relation visualization had been the focus of GIS research so far [7]. This shows the importance of research on spatial topological representation. They have analyzed classical data models like 9-Intersection Model [8–10], which could only represent relation between simple shapes and could not represent relation between arbitrarily complex objects.

Patrick Erik Bradley, Norbert Paul proposed to use G-maps to represent spatial topological structure [11],This method can effectively represent complex spatial objects. The main shortcoming of G-maps, however, is their extreme verbosity. G-maps have a tendency to becoming extremely complex even in the case of usual 3D volume modelling. In fact, it is too complex for practical use.

Constructive solid geometry (CSG) allows a modeler to create a complex surface or object by using Boolean operators(Union, Intersection, and Difference) to combine simpler objects [12]. CSG has many advantages such as simple, high speed for modeling, no redundant information, recording in detail parameters of solid and so on. For the simple information of CSG, the data structure is unable to store detailed information of the final objects, such as boundary, the vertex information.

Simon Daum proposed boundary-based Brep [13,14] method for analyzing topological information in BIM (Building Information Model this method has the advantage of describing detailed building topological information at the point, line and surface level [15]. However its largest modeling unit is surface instead of objects. This makes representing topological information among many building objects difficult.

Zhuojun Bao, Hans Grabowski proposed converting BRep into precise binary tree to handle spatial information [16]. This method is based on quadtree or octree. It can effectively represent single spatial model's surface, edge and vertex. It also consume less memory than octree [17]. However, it is still limited at representing topological information between models.

However, these three-dimensional topology models are still in their early stages [18–20]. There is no good way to access topological information. There are five major problems in the above literature: (1) they assumed that the model already exists when exploring topological relationship. (2) They only proposed topology models, but there is no research on how to access the topology. (3) Existing topological models could handle adjacency relationship but not intersection relationship. (4) Models based on body and image are constructed by stretching. They lack finer structures and internal topological connection relations. (5) Most topological models have data redundancy and have incomplete topology and geometry information fusion.

### 1.2 Overview and contributions

To address the above problems, this paper designed a system for expressing and accessing the topological elements by combining the advantage of CSG with Brep. The CSG–BRep topology model proposed in this paper has the following advantages:

- Significant reduction in data volume and improved ease of model storage and management
- Clear documentation of model generation process
- Enabling topological connection relationship processing
- Handling both regular and irregular shape
- Handling topological relations between models

This paper is divided into three sections:

In Sect. 2, we proposes a CSG–BRep topology model to describe the topological relation of 3D model.

In Sect. 3 we show algorithms for constructing CSG–BRep model by fitting parameters on point cloud and combine with topology model. We also use Boolean operation to construct complex CSG–BRep model and achieve the unity of topology.

In Sect. 4, we show an example of CSG–BRep model construction and show model querying and analysis of 3D spatial topological information.

## 2 CSG–BRep topological model

### 2.1 CSG–BRep model topology elements

Model is composed of basic elements [21]: vertex, edge, face, wire, shell, solid and compsolid as well as compound, showing a progression relationships from simple to complex as shown in Table 1.

We would like our model to contain topology connection relationship [22–24]. CSG method could handle the topology between different models. However, the smallest unit of representation in CSG is a solid and could not represent model internal topological relations. In this paper, we combined CSG and BRep to build topological model. Solid boundary is defined by face set. Each face is defined by its surface and surface boundary. Face boundary is defined by its edge set. An edge is defined by a set of points. As shown in Fig. 1.

In topology, edge is composed of vertex, wire is composed of edge, face is composed of wire, shell or solid is composed of face. A higher level topological object is progressively constructed from a lower level topological object. Objects

**Table 1** Topology of the elements and its significance

| Element | Meaning |
|---|---|
| Vertex | A topological element which is corresponding to the point. It has no dimension |
| Edge | A topological element which is corresponding to the constrained curve. It is one-dimensional |
| Wire | A set of edges (via point) and may not closed |
| Face | A part of the plane in 2D and a part of the surface in 3D |
| Shell | Made of faces, may or may not be closed |
| Solid | Space confined by the shell, three-dimensional |
| Compsolid | Composed of solids connected by surfaces |
| Compound | An object which is composed of different type of elements |

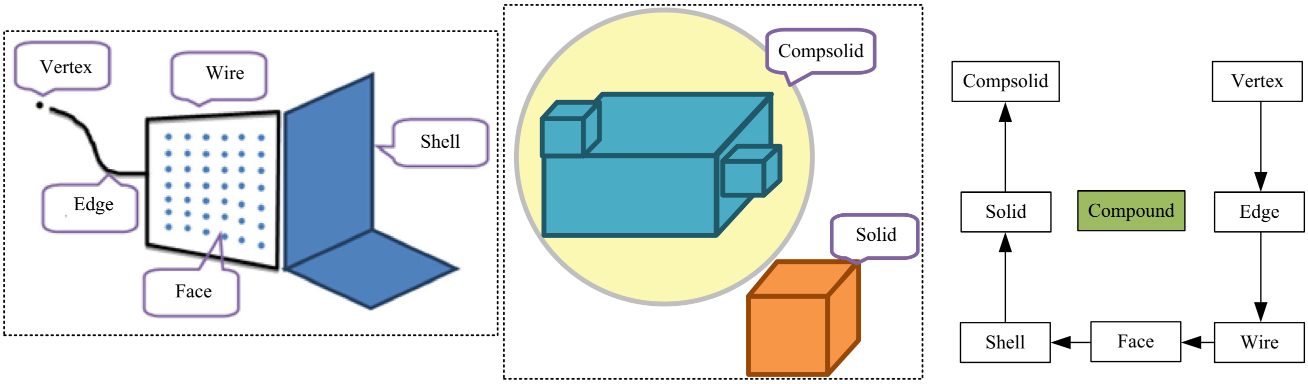


**Fig. 1** The structure of the topological model based on CSG–BRep

from different levels cannot be combined. At the same time, different solid can be formed into compsolid. All topology objects can form compound.

Higher level topology element contains references to lower level elements [25]. All references are from more complex elements to simpler topology element. Meanwhile, simpler topology element may be shared by different, more complex, topology element. Figure 2 shows an example model according to the topology model proposed in this paper.

In our topology model, only three kinds of topological elements are directly associated with geometric objects: vertex, edge, face are associated with Geo_Vertex, Geo_Edge, Geo_Face, respectively. Further, each vertex, edge, and face is respectively assigned a tolerance value for determining the vertex, edge, and face that is generated within the error range. Geometric meaning of a vertex tolerance of T is a ball with radius T centered on the vertex. The ball must contain all the endpoints of curve which is connected with the vertex. Tolerance of the edge is the maximum deviation between three-dimensional curves and any other representations. Its geometric meaning is a tube which radius is T and contains a three-dimensional curve as skeleton. Geometric meaning of a face tolerance is a plate with thickness T. Under normal circumstances, the following criteria must be met: When the edge is located at the face, the vertex is located at the edge, tolerance of face <= tolerance of edge <= tolerance of vertex. As shown in Fig. 3: tolerance of geometric elements which is associated with topological elements.

Geometric basic objects such as point, curve and surface can be further linked to more complex geometric objects. Then topology information and geometric information is directly or indirectly linked together. When performing a variety of operations, topology information can be accessed through geometric information or vice versa.

Each topological element belongs to a shape (CSG_Shape) object. Mid-level topological object is connected with both parent and child topological objects. For a given object, our model allows accessing both parent and child topological relations. Arbitrary topology object contains three variables: topological location CSGLocation, topological orientation CSGOrientation and a subset of the topological shapes CSG_Subshape. As shown in Fig. 4: CSG_Subshape records a set of connected topology objects. CSGLocation records the rotation matrix relative to the world coordinate system. CSGOrientation records orientation of topological objects.

## 2.2 Topological location

In order to track the position of the topological elements, each shape defines a local coordinate system CSGLocation. The

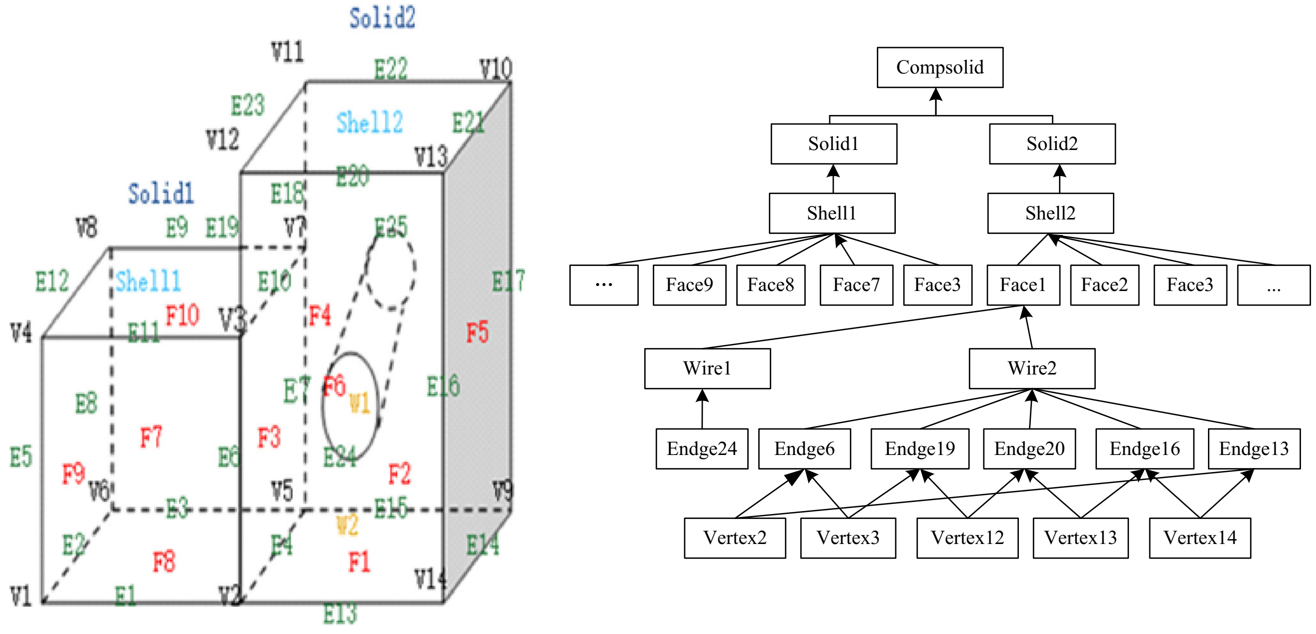


**Fig. 2** Description of the complex model

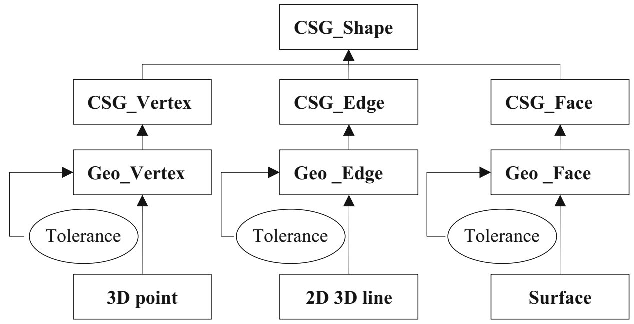


**Fig. 3** Geometry elements which is directly related to topology elements

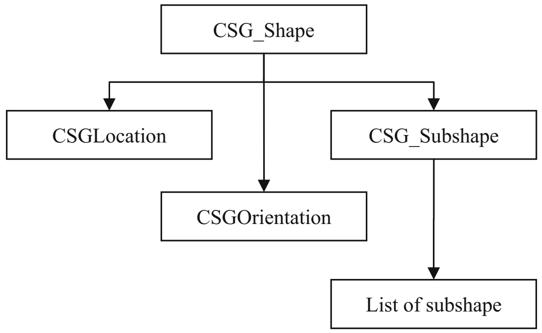


**Fig. 4** Three member variables of shape

local coordinate system can be represented by the following two ways: Three mutually perpendicular vector which is represented by the right hand rule,A transformation relative to the world coordinate system. CSGLocation records the rotation matrix relative to the world coordinate system. Let Q be the matrix stored in CSGLocation, then Q must be a 3 * 4 matrix, and the following two conditions must be met:

$$1)\ \mathrm{Q}_2 = \begin{pmatrix} q11\ q12\ q13 \\ q21\ q22\ q23 \\ q31\ q32\ q33 \end{pmatrix}, \mathrm{d} = |\mathrm{Q}_2|, \mathrm{d} \neq 0 \tag{1}$$

$$2)\ \mathrm{Q}_3 = \frac{Q2}{\sqrt[3]{\mathrm{d}}}, \mathrm{Q}_3^{\mathrm{T}} = \mathrm{Q}_3^{-1} \tag{2}$$

Matrix Q is a linear transformation matrix, it can convert a point (x, y, z) into another point (u, v, w) by matrix multiplication:

$$\begin{pmatrix} \mathrm{u} \\ \mathrm{v} \\ \mathrm{w} \end{pmatrix} = \mathrm{Q}^{*} \left(\mathrm{x\ y\ z\ 1}\right)^{\mathrm{T}}$$

$$= \begin{pmatrix} q11^{*}x + q12^{*}y + q13^{*}z + q14 \\ q21^{*}x + q22^{*}y + q23^{*}z + q24 \\ 931^{*}x + q32^{*}y + q33^{*}z + q34 \end{pmatrix} \tag{3}$$

Q could also be a combination of basic transformation matrices. Composite transformation matrix can be obtained by combinations of these basic transformation matrices.

The location of the shape will affect its child shape. For example, an edge do translation transformation along the vector (x, y, z), at this time transformation matrix which is relative to the world coordinate system of the edge stored in CSGLocation is:

**Table 2** Four-definition of the interior according to CSGOrientation

| CSGOrientation | Definition of the interior |
|---|---|
| CSG_FORWARD | The interior is the default region |
| CSG_REVERSED | The interior is the region complementary to the default |
| CSG_INTERNAL | The interior includes both regions |
| CSG_EXTERNAL | The interior includes neither region |

$$\begin{pmatrix} 1\ 0\ 0\ \mathrm{x} \\ 0\ 1\ 0\ \mathrm{y} \\ 0\ 0\ 1\ \mathrm{z} \end{pmatrix} \tag{4}$$

When traversing a subset of the edge, CSGLocation of vertex is still stored in the variable matrix (4). If translational and rotational transform transformation (along the X-axis PI / 2) are applied on edge, the same composites transformation matrix which is relative to the world coordinate system stored in CSGLocation of edge and its child vertex is:

$$\begin{pmatrix} 1\ 0\ 0\ \mathrm{x} \\ 0\ 1\ 0\ \mathrm{y} \\ 0\ 0\ 1\ \mathrm{z} \end{pmatrix} * \begin{pmatrix} 1\ 0\ 0 & 0 \\ 0\ 0\ -1 & 0 \\ 0\ 1\ 0 & 0 \end{pmatrix} \tag{5}$$

### 2.3 Topological orientation

Topological orientation (CSGOrientation) is extremely important and is closely related to boundary$^{14}$. This paper uses parameters notation to describe curves and surfaces because parametric representation is simple, easy to calculate and numerically stable. There are three kinds of shape need to use the topological orientation: curve which is constrained by vertex,surface which is constrained by edge,space which is constrained by face. In each case the topological form used as the boundary of a geometric domain of a higher dimension defines two local regions of which one is arbitrarily considered as the default region. For a curve limited by a vertex the default region is the set of points with parameters greater than the vertex. That is to say it is the part of the curve after the vertex following the natural direction along the curve. For a surface limited by an edge the default region is on the left of the edge following its natural direction. For a space limited by a face the default region is found on the negative side of the normal to the surface.

Based on this default region, the CSGOrientation allows definition of region to be kept, which is called the interior. There are four CSGOrientation defining the interior. As shown in Table 2.

The notion of CSGOrientation is a very general one. In the CSG–BRep model, it can be used in any context where regions or boundaries appear.

In topology, CSG_FORWARD and CSG_REVERSED are most widely used. CSGorientation of shape has an impact on its child. The vertex orientation has no direct geometric meaning. If a vertex's (CSG_Vertex) orientation is CSG_FORWARD, it must match the small side of the curve which is represented by parameter value. On the other hand, if a vertex's (CSG_Vertex) orientation is CSG_REVERSED, it must match the large side of the curve which is represented by parameter value. If an edge's (CSG_Edge) orientation is CSG_FORWARD, it means the logical direction of the edge is the same as the natural direction of parametric curves. If an edge's (CSG_Edge) orientation is CSG_REVERSED, it means the logical direction of the edge is opposite of the natural direction of parametric curves. Similarly, if a face's (CSG_Face) orientation is CSG_FORWARD, it means the direction of face is the same as the normal direction of surface. If a face's (CSG_Face) orientation is CSG_REVERSED, it means the direction of face is opposite of the normal direction of surface. CSGOrientation plays a vital role in shape visualization and shape Boolean operations.

### 2.4 Subset shape

For a shape, its subset shape (CSG_Subshape) contains a list of reference to all shapes that are related to the said shape. When a simple reference is insufficient (An edge can be used by more than one face of a solid), two pieces of information are added - topological orientation and a topological location reference. This is necessary for two topology elements of the same level to correctly sharing the same lower topology element.

## 3 Access algorithms of topology element based on topology model

### 3.1 Construction of detailed voxel model from point cloud data

We presents following algorithm for building voxel model from point cloud data. Firstly, we use improved RANSAC algorithm to fit parameters for simple point cloud data [26–28]. For complex point cloud data, we first perform manual segmentation then fit parameters to determine CSG voxel size. CSG voxel is constructed from basic topology elements: first, geometry and topology data are fused together according to CSG–BRep model, then voxel model are created from simple element like vertex to more complex elements. A simple voxel model include box, cylinder, cone, sphere, torus, or closed free-form surfaces. As shown in Fig. 5, cylindrical and spherical and cone and revolution (vase)surface are

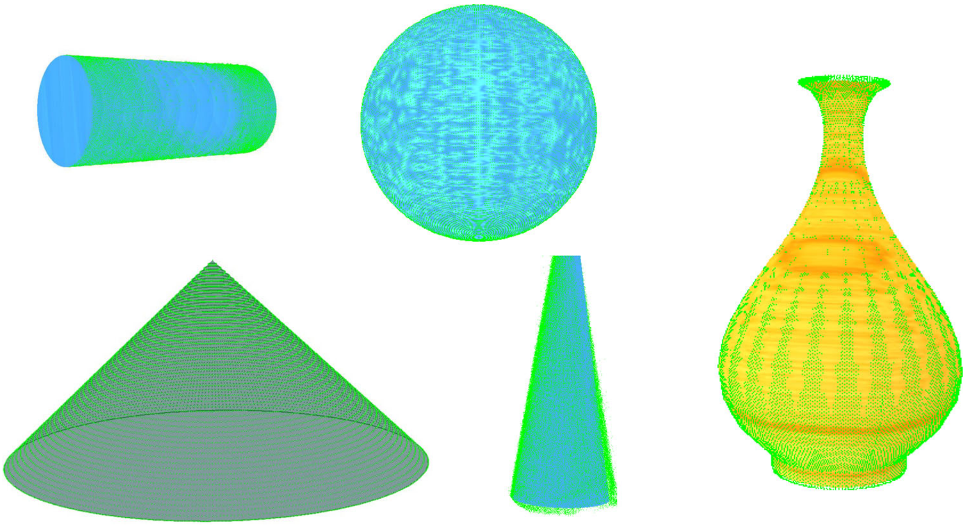


**Fig. 5** Fit parameters based on point cloud and build CSG voxel

fitted to discrete point cloud, then CSG voxel cylinders and spheres is generated.

## 3.2 Algorithm of accessing topology

There are two kinds of algorithms for accessing topology: for a given topological shape object, we can either access topological structures that are subset of the shape or access topological structures that are superset of the shape.

### *3.2.1 Access topological structures that are subset of a shape*

A CSG–BRep topological shape object contains a variables CSG_Subshape. It contains a list of topological objects that are subset of the containing object. This allows direct traversal of topology elements. If a topological element is referenced multiple times, then that element could be traversed multiple times. For example, an edge in box is usually shared by two faces. To ensure a shared topology element is only accessed once during a traversal, hash map is used to remove duplicated reference.

Pseudo-code implementation:

```
void MapCsgChildShape(const CSG_Shape& S,    const    CSG_ShapeEnum    T,
    CSG_MapOfCsgShape& M)
{
    CsgExp_Explorer Ex(S, T);
    while (Ex.More()) {
        M.Add(Ex.Current());
        Ex.Next();
    }
}
```

In the above example: S is the shape to be traversed. T is the enumeration type for the kind of shape we want, in this case T is ‘edge’. CsgExp_Exporer is a helper class to iterate through all subset shapes of S of the type T. M is a hash map that only allow one reference of a given element to be added.

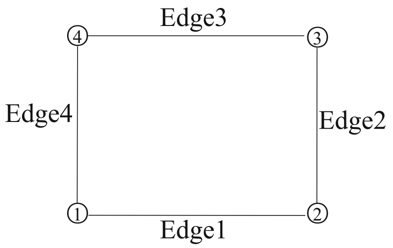


**Fig. 6** Find the edge associated with the vertex on the square

### *3.2.2 Access topological structure of a superset of the shape*

In topological data structure of CSG–BRep presented in this paper, references are from higher level shapes to lower level shapes (e.g. from CSG_Shell or CSG_Solid to CSG_Vertex). Therefore there is no direct access to topological elements that are superset of a given topological element. Therefore, we propose another algorithm. We will use a square as an example, as shown in Fig. 6.

To find edges associated with a given vertex on the square, we first use the following algorithm to build an index from vertex to edges:

**Table 3** Find the edge associated with the vertex on the square

| VertexID | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| LeftEdge | Edge4 | Edge1 | Edge2 | Edge3 |
| RightEdge | Edge1 | Edge2 | Edge3 | Edge4 |

a. Traverse the square. Find all edges which are located on the square.
b. Traverse all vertices which are located on the edges found in step a.
c. Populate an index from vertex to edge during the traversal in step a and b.

Pseudo-code implementation for creating the index:

```
void    MapCsgFatherShape (
        const CSG_Shape& S,
        const CSG_ShapeEnum TS,
        const CSG_ShapeEnum TA,
        CSG_IndexOfCsgShape & Index)
{
      CsgExp_Explorer exa(S, TA);
      while (exa.More()) {
            // traverse ancestor shapes
            const CSG_Shape& ancestor = exa.Current();
            CsgExp_Explorer exs(ancestor, TS);
            while (exs.More()) {
                  // traverse children shapes
                  const CSG_Shape& child = exs.Current();
                  Index.Add(child, ancestor);
                  exs.Next();
            }
            exa.Next();
      }
}
```

S is the shape to be traversed, TS is the enumeration type for vertex, TA is the enumeration type for edge. CsgExp_Exporer is a helper class to iterate through all subset shapes of S of a given type T. Index is a data structure that allow only once instance of ‘child’ as key, but allow accumulating references of ‘ancestor’.

Using this algorithm we can build an index as shown in Table 3 for the example square. Then we can look up the edges for any given vertex on the square.

## 4 Unify topological relationships between models using Boolean operators

We could build and access topological relations of a single model using algorithms presented in previous sections. However, we still need to handle topological relations between models. Here we propose the use of Boolean operators to unify topological relationships between models [29,30]. This allows algorithm for accessing topology to handle relationships between models.

### 4.1 The geometry parametric equation of the typical voxel

Intersection operations is the core operations of Boolean operations. Example operations include the intersection between one line and another line, the intersection between a line and a surface, as well as the intersection between two surfaces and so on. The actual intersection computation uses the geometric information associated with a topology element, such as parametric equation of curves and surfaces. Table 4 shows parametric equations of some typical basic voxel.

In Table 4, a line is defined by a 3D point P and a 3D vector D. The line passes through the point P and is parallel to vector D. A plane is defined by a 3D point P and pair wise orthogonal 3D vectors Du and Dv which defines a normal N. The plane passes through the point P, and has a normal N. A cylinder is defined by a 3D point P, pairwise orthogonal 3D vectors Dv, DX and DY and a non-negative real number r. The cylinder axis passes through the point P and is parallel to vector Dv. The cylinder has the radius r. The cone data consist of a 3D point P, pairwise orthogonal 3D directions DZ, DX and DY. The cone axis passes through the point P and has the direction DZ. The plane which passes through the point P and is parallel to directions DX and DY is the cone referenced plane. The cone section by the plane is a circle with the radius r. A sphere is defined by a 3D point P, pairwise orthogonal 3D directions DZ, DX and DY and a non-negative real number r. The sphere has the center at P, radius r. A torus is defined by a 3D point P, pairwise orthogonal 3D directions DZ, DX and DY and non-negative real numbers r1 and r2. The torus axis passes through the point P and has the direction DZ. r1 is the distance from the torus circle center to the axis. The torus circle has the radius r2. A linear extrusion surface data consist of a 3D direction Dv. The linear extrusion surface has the direction Dv and base curve C. A revolution surface is defined by a 3D point P, a 3D direction D and a 3D curve. The surface axis passes through the point P and has the direction D. 3D curve and the axis are coplanar.

### 4.2 Boolean operation algorithm

The Boolean operations proposed here is based on intersection algorithm of parametric curve and surface. When basic voxel are represented by parametric equation, we propose following methods perform Boolean operations between voxels.

a. Calculate the intersection between two voxels. Assume the two voxels to be intersected to be A and B. First, traverse all edges (parametric curve) of A. Meanwhile,

**Table 4** Several parametric equations of typical basic voxel

| Typical elements | Parametric equations |
|---|---|
| Line | $C(u) = P + u \cdot D, u \in (-\infty, \infty)$ |
| Plane | $S(u,v) = P + u \cdot Du + v \cdot Dv, (u,v) \in (-\infty,\infty) \times (-\infty \quad \infty)$ |
| Cylinder | $S(u,v) = P + r \cdot (\cos(u) \cdot Dx + \sin(u) \cdot Dy) + v \cdot Dv, (u,v) \in [0, 2 \cdot \pi) \times (-\infty,\infty)$ |
| Cone | $S(u,v) = P + (r + v \cdot \sin(\phi)) \cdot (\cos(u) \cdot DX + \sin(u) \cdot DY) + v \cdot \cos(\phi) \cdot DZ, (u,v) \in [0, 2 \cdot \pi) \times (-\infty,\infty)$ |
| Sphere torus | $S(u, v) = P + r \cdot \cos(v) \cdot (\cos(u) \cdot D_x + \sin(u) \cdot D_y) + r \cdot \sin(v) \cdot D_Z, (u, v) \in [0, 2 \cdot \pi) \times [-\pi/2, \pi/2]$ |
| Surface of linear extrusion | $S(u, v) = P + (r_1 + r_2 \cdot \cos(v)) \cdot (\cos(u) \cdot D_x + \sin(u) \cdot D_y) + r_2 \cdot \sin(v) \cdot D_Z, (u, v) \in [0, 2 \cdot \pi) \times [0, 2 \cdot \pi)$ |
| Surface of revolution | $S(u, v) = C(u) + v \cdot D_v, (u, v) \in \text{domain}(C) \times (-\infty, \infty)$ |
| | $S(u, v) = P + V_D(v) + \cos(u) \cdot (V (v) - V_D(v)) + \sin(u) \cdot [D,V (v)], (u, v) \in [0, 2 \cdot \pi) \times \text{domain}(C)$ |

traverse all faces (parametric surface) of B. Calculate all the points of intersection between edge of A and face of B. In this case, all the edges of A are divided by the points of intersection. Similarly, calculate all the points of intersection between edge of B and face of A.

b. Classify the edge which has been divided. After the step a, the edges of A are divided into several segments. A segment can falls into one of the following three cases: 1: the segment is inside B, 2: the segment is outside of B, 3: the segments of A and B overlap each other. Similarly, perform the same classification to edges in B with respect to A.

c. Calculate a new line of intersection. Firstly we determine effective intersection points of A. If an intersection point in step a is on the boundary of A and within or on the boundary of B, then the intersection point is an effective intersection point. Similarly, we also determine effective intersection points of B with respect to A. After obtaining all effective points, we reconnect these effective points to get new lines of intersection.

d. Remove or keep original boundaries of A and B. We use the classification result of step b and the new line of intersection of step c to complete the Boolean operation. When intersection algorithm is applied to A and B, keep A's edges that are inside or on boundary of B and remove A's edges that are outside of B. Similarly, perform the same action to B's edges with respect to A. When union or difference algorithm is applied to A and B, keep A's edges that are outside of B and remove A's edges that are inside of B.

e. Use the new boundaries to reconstruct a topology model shape.

## 5 Experiments and results

We conducted an experiment to model wooden structure of Supreme Gate which is in Beijing. We used Visual Studio 2010 and C++ for developing software used in this experiment. We used point cloud data collected by three-dimensional laser radar as data source. We first used improved RANSAC algorithm to fit basic voxel parameters, then build topology model using Boolean operations. The model is shown in Fig. 7. For comparison, we also use Geomagic Studio12 to generate a triangular mesh model of the same structure, as shown in Fig. 8.

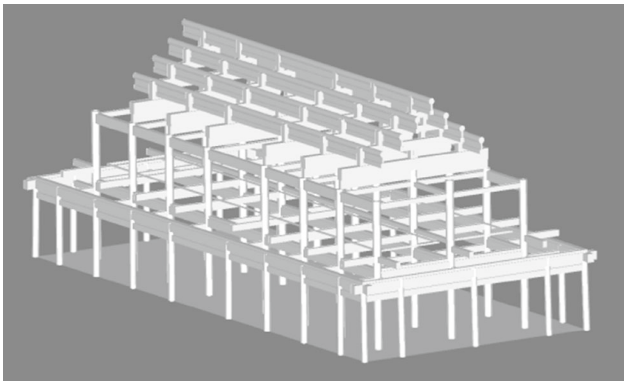


**Fig. 7** CSG–Brep model of wooden structure

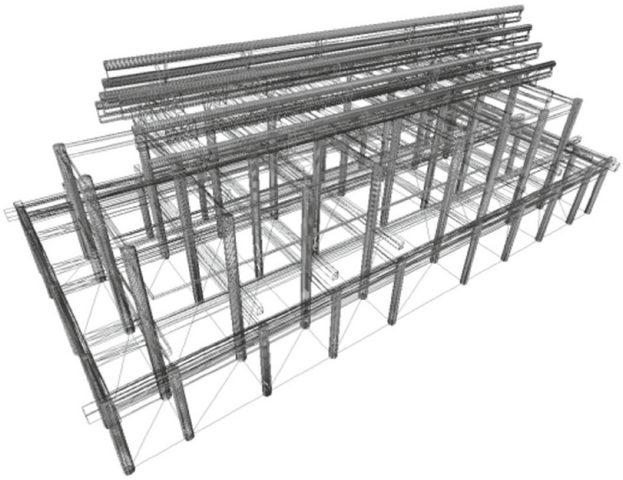


**Fig. 8** Triangular mesh model of wooden structure

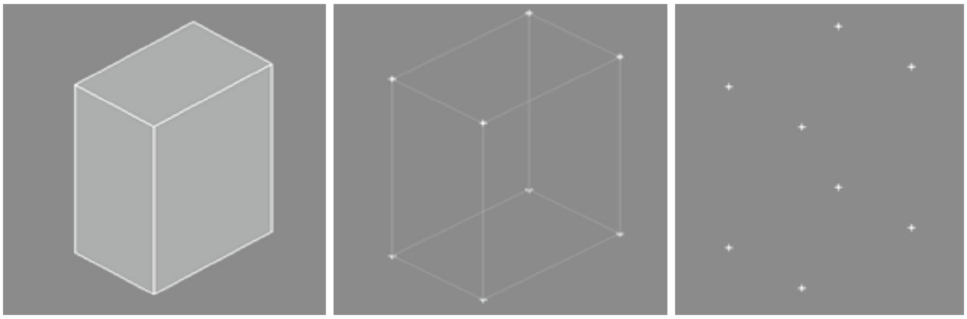


**Fig. 9** Access child edge and point

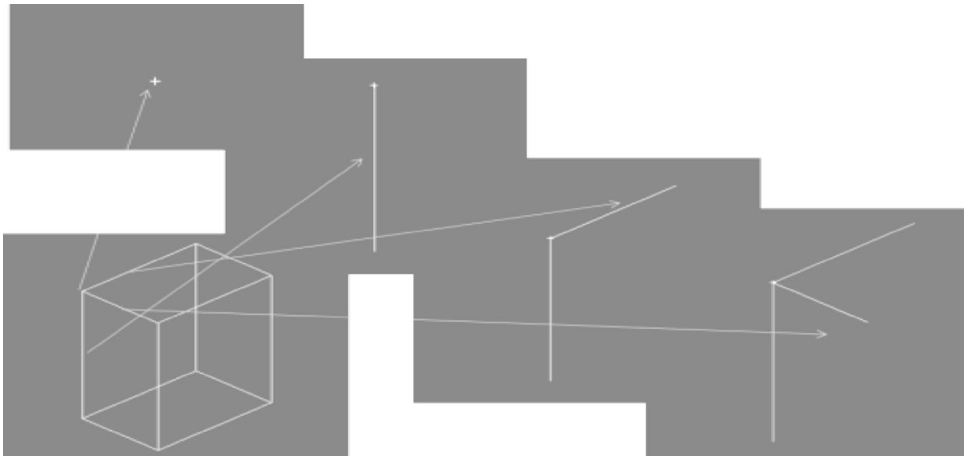


**Fig. 10** Access father edge

We can query and analyze the CSG–BRep wood structure model through algorithms presented in access algorithms of topology element based on topology model. We found that it consist of 81 cylinders and 148 boxes. The model could be split and it has detailed topological relations. The model's memory footprint is only 336KB. In comparison, the triangular mesh model of wooden structure consist of a total of 3991 points and 7475 triangles. This model could not be easily split nor contain topological relations. Its memory footprint is 1986KB.

This comparison demonstrates that CSG–BRep model is effective in reducing data size and produce models that can be split to different components. Most importantly, the CSG–BRep model contains detailed topological relations and enables query and analysis of 3D spatial topological information.

The ability to query a CSG–BRep model enables us to extract information previous difficult to extract. For example, we can use algorithm presented in 0 to extract a basic box which represent a wooden pillar in the large wooden structure. As shown in Fig. 9, we could query many important information about this box by accessing it is subset elements. We could extract the physical dimensions of this pillar by access edges information in this box. Figure 10 shows that we can query connected edges and surfaces of if we a given a vertex. Figure 11, shows that we can also query parent face the box if given a vertex.

This ability to query both subset and superset elements allows us to flexibly extract information contained in models generated from point cloud data.

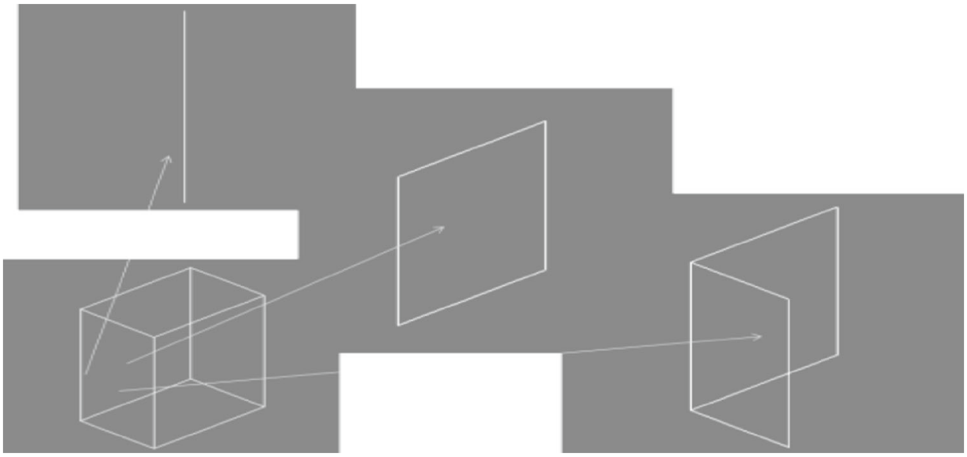
## 6 Conclusion

Firstly, we have described CSG–BRep topological model and its uses of topological location, topological orientation, subset element list when defining a model. We have also listed two algorithms for accessing detailed elements in a detailed CSG–BRep model. These two algorithms also enables topological query for 3D spatial objects.

Secondly, we have presented operations on CSG–BRep models to convert topological relationship between models into topological relationship inside a model. This approach describes spatial objects using basic geometric objects, and in turn converts spatial object intersection into intersection between parametric representations of basic geometric objects.

Lastly, we have demonstrated constructing a CSG–BRep model from real world point cloud data. We have shown that CSG–BRep model has several advantages of being composable, having smaller data size, containing complete topological information and having queryable internal elements than triangular mesh model. We have shown that CSG–BRep model provides a solution for modeling and accessing detailed topological relation.

**Fig. 11** Access father face



**Acknowledgements** This paper was supported by plan projects National Administration of Surveying,Mapping and Geoinformation of China (Grant No. 2013CH-15) and by the National Natural Science Foundation of China (Grant No. 41301429). Project also supported by the Special Scientific Research Fund of Surveying Public Welfare Profession of China (Grant No. 201512009), and by science and technology planning project of Beijing Municipal Education Commission (Grant No. 2016 subprojects49) and by Beijing Natural Science Foundation (No. 8142014).

## References

1. Chen, J., Zhao, R.L.: Spatial relationsin GIS: a survey on its key issues and research progress. Acta Geodaetica Cartogr. Sin. **28**(2), 95–102 (1999)
2. Choudhary, A., Kandemir, M., No, J., et al.: Data management for large—scale scientific computations in high performance distributed systems. Cluster Comput. **3**(1), 45–60 (2000)
3. Wang, Y., Liu, Z., Liao, H., et al.: Improving the performance of GIS polygon overlay computation with MapReduce for spatial big data processin. Cluster Comput. **18**(2), 1–10 (2015)
4. Lin, L.I., Zhigang, Z.H.A.O., Renzhong, G.U.O., Biao, H.E.: Research on 3D topology construction of space objects. Geomat. Inf. Sci. Wuhan Univ. **37**(06), 719–723 (2012)
5. Zhenfeng, S.H.A.O., Deren, L.I., Qimin, C.: 3D topological reconstruction of complicated buildings based on stereo image pair. Editor. Board Geomat. Inf. Sci. Wuhan Univ. **29**(11), 999–1003 (2004)
6. Changbin, W.U., Guonian, L.V.: Research on several problems of spatial topological relations. J. Geo Inf. Sci. **12**(4), 524–531 (2010)
7. Zhao, L., Chen, L., Ranjan, R., et al.: Geographical information system parallelization for spatial big data processing: a review. Cluster Comput. 1–14 (2015)
8. Zhanlong, C.H.E.N., Qiqi, F.E.N.G., Xincai, W.U.: Representation model of topological relations between complex planar objects. Acta Geod. Cartogr. Sin. **44**(4), 438–444 (2015)
9. Li, S.J.: A complete classification of topological relations using the 9-intersection method. Int. J. Geogr. Inf. Sci. **20**(6), 589–610 (2006)
10. Du, S.H., Guo, L., Wang, Q., Qin, Q.M.: Efficiently computing and deriving topological relation matrices between complex regions with broad boundaries. ISPRS J. Photogramm. Remote Sens. **63**(6), 593–609 (2008)
11. Bradley, P.E., Paul, N.: Comparing G-maps with other topological data structures. Springer J. GeoInformatica **18**(3), 595–620 (2014)
12. , James, F.D.: Constructive Solid Geometry, Computer Graphics: Principles and Practice, pp. 557–558. Addison-Wesley (1996)
13. Kim, B.C., Mun, D.: Feature-based simplification of boundary representation models using sequential iterative volume decomposition. Elsevier J. Comput. Graph. **38**(38), 97–107 (2014)
14. Levine, J.A., Jadhav, S., Bhatia, H., Pascucci, V., Bremer, P.T.: A quantized boundary representation of 2D flows. Wiley J. Comput. Graph. Forum **31**(3pt1), 945–954 (2012)
15. Daum, S., Borrmann, A.: Processing of topological BIM queries using boundary representation based methods. Elsevier J. Adv. Eng. Inf. **28**(4), 272–286 (2014)
16. Bao, Z., Grabowski, H.: Converting boundary representations to exact bintrees. Elsevier J. Comput. Ind. **37**(1), 55–66 (1998)
17. An, N., et al.: Storing spatial data on a network of workstations. Cluster Computing. **2**(4), 259–270 (1999)
18. Liu, K., Shi, W.: Extended model of topological relations between spatial objects in geographic information systems. Int. J. Appl. Earth Obs. Geoinf., **9**(3):264–275 (2007)
19. Nonaka, H., Ohsawa, Y.: A method of caching topology on implicit topology model. J-STAGE J. Theory Appl. GIS **11**(1):9–17 (2003)
20. Nian, R., He, B., Lendasse, A.: 3D object recognition based on a geometrical topology model and extreme learning machine. Springer J. Neural Comput. Appl. **22**(3), 427–433 (2013)
21. He, B., Li, L., Guo, R., Shi, Y.: 3D topological reconstruction of heterogeneous buildings considering exterior topology. Geomat. Inf. Sci. Wuhan Univ. **26**(5), 579–583 (2011)
22. Zhang, H., Yu, J., Liu, Y., Du, P.: Establishment of 3D topology for laneway based on spatial reasoning. Geogr. Geo-Inf. Sci. **25**(4), 53–55 (2009)
23. Boess, V., Ammermann, C., Niederwestberg, D., Denkena, B.: Contact zone analysis based on multidexel workpiece model and detailed tool geometry representation. Elsevier J. Procedia CIRP **4**, 41–45 (2012)
24. Lauren, L., Beghini, A.P., Rodrigo E., Menezes, I.F.M., Celes, W., Paulino, G.H.: An object-oriented framework for finite element analysis based on a compact topological data structure. Elsevier J. Adv. Eng. Softw. **68**, 40–48 (2014)
25. van Oosterom, P.: Variable-scale topological data structures suitable for progressive data transfer: the GAP-face tree and GAP-edge forest. J. Cartogr. Geogr. Inf. Sci. **32**(4), 331–346 (2005)
26. Dung, L.-R., Huang, C.-M., Yin-Yi, W.: Implementation of RANSAC algorithm for feature-based image registration. J. Comput. Commun. **1**(6), 46–50 (2013)
27. Yue, Z., Christian, C., Hervé, L.: Automatic needle detection and tracking in 3D ultrasound using an ROI-based RANSAC and Kalman method. J. Ultrason. imaging **35**(4), 283–306 (2013)

28. Schnabel, R., Wahl, R., Klein, R.: Efficient RANSAC for point-cloud shape detection. Comp. Graph. Forum **26**(2), 214–226 (2007)
29. Martínez, F., Ogayar, C., Jiménez, J.R., Rueda, A.J.: A simple algorithm for Boolean operations on polygons. Elsevier J. Adv. Eng. Softw. **64**, 11–19 (2013)
30. Pereira, A.M.B., Arruda, M.C., de Miranda, A.C.O., Lira, W.W.M., Martha, L.F.: Boolean operations on multi-region solids for mesh generation. Springer J. Eng. Comput. **28**(3), 225–239 (2012)

**Huang Ming** is an associate professor of civil engineering and architecture at Beijing University. He received the Ph.D. degree in geophysics from China University of Mining Technology, Beijing in 2009. His research interests include 3DGIS applications, 3D modeling, 3D visualization and 3D TLiDAR point cloud data processing.

**Du Yanzhu** graduated from National University of Singapore with First Class Honours in 2009. He continued to pursue his interest in artificial intelligence in Masters' program at Stanford University. After earning his master's degree, he joined a crowdsourcing startup CloudCrowd in San Francisco where he gained first-hand experience at merging human intelligence and machine intelligence. He currently works as Software Architect in a productivity software startup Ivy Softworks in Seattle.

**Zhang Jianguang** is a master's student in the Department of Surveying and Mapping Engineering, Beijing University of Civil Engineering and Architecture, China. His research interests include laser point cloud interactive visualization, point cloud data processing.

**Zhang Yong** is a master's student in the Department of Surveying and Mapping Engineering, Beijing University of Civil Engineering and Architecture, China. His research interests include 3D GIS, CSG.