A SIMPLIFIED GEOMETRIC AND TOPOLOGICAL MODELING OF 3D BUILDINGS: COMBINATION OF SURFACE-BASED AND SOLID-BASED REPRESENTATIONS

Chokri Koussa, Mathieu Koehl
MAP-PAGE UMR 694 CNRS/MCC – INSA de Strasbourg, Graduate School for sciences and technology,
24 Boulevard de la Victoire, F-67084 Strasbourg, France
chokri.koussa@insa-strasbourg.fr, mathieu.koehl@insa-strasbourg.fr

ABSTRACT

This paper focuses on the field of 3D data modeling within Geographical Information System (GIS). CSG (Constructive Solid Geometry) and B-Rep (Boundary Representation) methods have been combined to define a simplified geometric and topological conceptual 3D model for buildings. As a building is composed of a set of walls, the geometric and topological representation will be based essentially on the concept of wall object. In this work, we consider that the same urban object can be modeled differently depending on the Level Of Detail (LOD) and the further operations on the 3D model. For example, a wall can be modeled as a simple surface, but if more details are required, it can be modeled as a solid having a non negligible thickness. Therefore, we propose a 3D model for urban objects that dynamically change appearance according to the level of detail or the level of zoom. This can bring more realism and flexibility when representing such 3D objects. In addition to data visualization and interrogation, data storage optimization is one of the most important points of interest for us in this project. The proposal consists in the storage of the minimum of data in the database. Then, the complete 3D model is obtained by interpretation of complementary parameters related to the initial dataset, i.e. solid interpretation from a surface and a thickness. The originality of the proposed work consists in an algorithm that allows an automatic passage from surface-based representation to solid-based representation. The conceptual model has been translated into a PostgreSQL database. To create the final 3D model, the data are queried from the database using a Java platform. Some samples are described to illustrate the implementation of the work concepts.

Key words: 3D GIS, modeling, CSG, B-Rep, SDB (Spatial DataBase).

INTRODUCTION

GIS has become an important issue for managing and processing geo-referenced data. It is used by both public and private communities to accomplish different types of applications. 2D GIS software (i.e, ArcGIS, PostGIS, etc.) have known a considerable maturity for both data visualization and analysis. But, the real world objects are tridimensional. Thus, “X” and “Y” are unable to represent faithfully the reality when “Z” is missing. 3D data acquisition, storage and visualization are more and more unproblematic due to the technological progress of both spatial data acquisition tools (Global Positioning System-GPS, Terrestrial Laser Scanner-TLS, etc.) and computer tools (hardware, software, high-speed Internet, etc.). Before their integration in the 3D GIS, spatial data must be organized and structured to optimize their further visualization and querying. Therefore, 3D spatial modeling is one of the most important issues in 3D GIS research [7]. Such data are modeled through different levels of modeling (conceptual, logical and physical levels). The conceptual model consists in the representation of real world objects via a set of concepts and relationships between these concepts. This conceptual modeling is independent from all logical, physical and technological implementations. The particularity of spatial data is that are generally huge and complex to represent. Thus, conceptual modeling is a very important stage in the definition of data structure for 3D GIS. This paper focuses on the field of spatial data modeling for 3D GIS. It takes inspiration from the two principal 3D modeling approaches: B-Rep and CSG. These approaches are combined in order to optimize data storage and visualization. With this purpose, this work introduces a simple conceptual model for the geometry and topology of 3D buildings. It takes inspiration from the existing standards and research work. This paper is organized as follows: It starts by a summary of some 3D modeling approaches. Then, it explains work done in the field of spatial data modeling. After that, a new approach for 3D urban objects modeling is described by some UML (Unified Modeling Language) class diagrams. Next, some results and experimentation are shown to illustrate the implementation of the work concepts. Finally, a conclusion is drawn and our future work is explained.
RELATED WORK

Why Modeling
It is very important to clarify the meaning of the term “model” for data in general and for spatial data in particular. [20] defines a model as “a preliminary representation of something, serving as the plan from which the final, usually larger, object is to be constructed”. It is an abstract representation of a real world object, a phenomenon, an approach, an idea, etc. Such representation precedes generally the physical establishment of the object modeled. In this case, object modeled have to meet the requirements and constraints defined by the model. However, this is not true for spatial data because geographic objects already exist and the model has to meet the characteristics and the specificities of such objects. This can make the modeling task more difficult. [19] defines the spatial data modeling as the “analytical procedures applied with GIS. It is the set of procedures that simulates real-world conditions within a GIS using the spatial relationships of geographic features”. Spatial modeling includes the definition of spatial object geometry, their relationships, their thematic information, etc. Data modeling allows a well comprehension of data structure, data meaning and data use. Data manipulation and querying can so be easier and effective.

3D Modeling Approaches

• B-Rep (Boundary Representation)
 Boundary Representations (also known as B-Rep) resemble the naïve representations in which the object is described in terms of its surface boundaries: vertices, edges and faces [6]. This method represents shapes by their limits. B-Rep distinguishes the geometry from the topology. Geometry describes the exact shape and position of each of the edges, faces and vertices. For example, the geometry of a vertex is defined by its position in space generally given by its coordinates (x, y, z). However, topology describes the relationships (connectivity) between faces, edges and vertices. B-Rep is used in Computer Aided Design (CAD). B-Rep has many advantages: i) B-Rep is adapted to construct complex objects. ii) It describes very well the objects surfaces. iii) It is possible to do complex operation on B-Rep objects, i.e. calculating the volume of a solid. iv) The same B-Rep model can contain either geometric and interconnectivity constraints, etc. However, B-Rep has some disadvantages: i) The construction of complex solids is long and tedious. ii) B-Rep is not suitable for the representation of spherical and cylindrical objects. iii) B-Rep representation generates a large volume of data which can decrease the performance of their interrogation, etc. Euler operators are used to verify the validity of B-Rep solids topology.

The formula used is: $F - E + V - L = 2 (B - G)$ where $F$, $E$, $V$, $L$, $B$, and $G$ are respectively the number of Faces, Edges, Vertices, faces’ inner Loop, Bodies, and Genus.

• CSG (Constructive Solid Geometry)
 CSG is an acronym for Constructive Solid Geometry. To understand it, once again, it's best to dissect the term. Constructive, meaning used to put together something more complex, solid, meaning an obstruction (in this case), and geometry, which means a mathematically-defined shape. Basically, constructive solid geometry is geometry used to construct a larger solid piece of geometry. There are also “CSG operations”, which are mathematical algorithms used on CSG pieces (commonly called brushes) to cause brushes to interact with each other and the world [18]. A solid can be defined as a set of primitives shaped in volumetric instances such as cubes, cylinders, cones, spheres, etc. CSG models are constructed through three Boolean operations: the intersection (\( \cap \)) , union (\( \cup \)) and subtraction (\( - \)) of solid objects, some of which may be CSG objects themselves. The union of two objects results in an object that encloses the space occupied by the two given objects. Intersection results in an object that encloses the space where the two given objects overlap. Difference is an order dependent operator; it results in the first given object minus the space where the second intersected the first [8]. CSG solids have to be translated and oriented to be well localized in space. CSG is also used in CAD. It has many advantages: i) Objects constructed by this technique have perfect boundaries whatever the LOD. ii) Solid construction via CSG primitives and Boolean operations is more intuitive than specifying all solid boundaries using B-Rep representation.iii) CSG creates very realistic rendering (ray casting). However CSG has some limits: i) There is no formal description of solid boundaries. ii) We can easily reach complex construction operations which could decrease the performance of calculation, etc.

• Hybrid approach (CSG-Brep)
 The principle is the combination of B-Rep and CSG advantages and the limitation of their disadvantages. Advantages of such method can be: i) A solid can be made by both a CSG tree and B-Rep primitives. ii) It allows a rapid visualization of solids. iii) It can contain both geometric and topological constraints. iv) It describes very well the solid boundaries, etc. The limits of such approach are: i) Data can be very huge. ii) It is necessary to maintain the coherence between the B-Rep and CSG concepts, etc.

ASPRS 2009 Annual Conference
Baltimore, Maryland ♦ March 9-13, 2009
• Parametric approach

The principle of the parametric modeling is to define an entity by a set of parameters (numbers and features) which can be easily modified. Such parameters can be the length, the angle, the diameter, etc. of geometric objects. Parameter allows defining the entity behavior and the relationships between its components. This method is used in CAD tools, i.e. MCAD (Mechanical Computer Aided Design). Such tools use generally a tree to store the different operations done on the model. These operations can be the extrusion, scan, smoothing, etc. This method has some advantages: i) It is possible to attribute some thematic information to the geometric model. ii) The model can be semantically well described, etc. However, parametric modeling has also some limits: i) Geometry and features have to be defined manually by the designer so it can be a tedious task. ii) A solid is described by many parameters which can change, etc.

Spatial Data Modeling

Modeling spatial data is a very important stage in the creation of a GIS. It allows representing the reality via a set of concepts related with semantic relationships. A model allows designer to early discover the anomalies that can appear, e.g. redundancy, deletion and update anomalies, etc. So, he can correct the model before exploring data. The comprehension of data structures and their relationships facilitates their management, manipulation and interrogation. A 3D city model can include the urban objects and the space in which they exist. In addition to geographic information, spatial data can also be described by attribute information [9]. Spatial objects can be simple (e.g. point, line, surface) or composed of a collection of objects (e.g. composed surface or composed solid). In the literature, many spatial data models have been proposed. The 3D FDS (Formal Data Structure) is based on four objects: Point, line, surface and body and four primitives: Node, arc, edge and face [10]. In this model, the topology depends on geometry. ISO takes its inspiration with this model but recommends a total independence between the topological model and the geometric model [14]. Another approach was introduced by [13] called TEN (TEtrahedral Network). TEN has four primitives: tetrahedron, triangle, arc and node. A body is composed of tetrahedrons, a surface of triangles, a line of arcs and a point of nodes [17]. Since the space is completely subdivided into tetrahedrons, the interiors of objects (e.g. buildings), as well as the open space, are also decomposed into tetrahedrons. Such subdivision is rather inconvenient for 3D man-made objects [17]. UDM (Urban Data Model) represents the geometry of a body or a surface by planar convex faces [3]. Figure 1 shows its conceptual schema.

Figure 1. Urban Data Model, Coors, 2002.

The GSDM (Global Spatial Data Model) is a collection of existing mathematical concepts and procedures which can be used to manage spatial data both locally and globally. It consists of a functional model which describes the geometrical relationships and a stochastic model which describes the probabilistic characteristics -statistical qualities- of spatial data. The functional part of the model includes equations of geometrical geodesy and rules of vector algebra (solid geometry) as related to various coordinate systems [2].

Many other data models were proposed: SDM (Simplified Data Model) [16], and some Object-Oriented models e.g. 3D TIN-based OO model [1], OO-model of de la Losa and Cervelle [4], the Solid Object Management System (SOMAS) [12], etc.

International organizations such as OGC and ISO contribute in the definition of standards for the interoperability between heterogeneous applications working on spatial data. The ISO191**, defined by the TC211 (Technical committee 211), is a set of standards for geographic information. It includes the methods, the tools and the services for the acquisition, the analyses, the access, the presentation and the exchange of geographic information. In particular, ISO19107 defines a conceptual schema for the spatial features of geographic entities through geometric and topological objects. It defines four principle primitives: GM_POINT, GM_CURVE, GM_SURFACE and GM_SOLID. These primitives are represented in Figure 2.
The ISO19136 defines the encoding of ISO19107 in GML3. GML3 is a modeling language for geographic information. It is an OGC specification. It supports spatial and non spatial properties and implements the concepts of ISO19100. GML3 contains a set of primitives which include features, geometry, coordinates, reference system, etc. GML distinguishes between geometry and feature (a feature represents physical entities, i.e. buildings, bridges, etc.).

CityGML is a recent open model for the representation of 3D city objects. It is intended to become an open standard for the storage and the exchange of virtual 3D city models. It defines the properties of many topographic objects such as geometric, topological, semantic and appearance properties [11].

CityGML defines a geometric, topological and semantic model. It is based on GML3 specifications. The geometry model of GML 3 consists of primitives, which may be combined to form complex, composite geometries or aggregates. For each dimension, there is a geometrical primitive: a zero-dimensional object is a Point, a one-dimensional a _Curve, a two-dimensional a _Surface, and a three-dimensional a _Solid [11]. CityGML is applied to various domains e.g. urban planning, disaster management, tourism, etc. Thus, it introduces a set of models such as Appearance Model, Digital Terrain Model, Transportation Model, Vegetation Model, etc. Many projects implement the OGC and ISO standards, i.e. Geoserver, Degree, RedSpider, GeoOxygene, Oracle spatial, etc. The commercial tool Oracle Spatial handles spatial data via a set of primitives, i.e. point, linestring, polygon, solid, collection (multipoint, multicurve...), ring, composite_solid, composite_surface, etc.

**PROPOSAL**

**Spatial Data Acquisition**

Spatial data acquisition tools i.e. Global Positioning System (GPS), Terrestrial Laser Scanner (TLS), etc. have demonstrated a high level of maturity. It is possible, nowadays, to get detailed information about buildings, roads, maps, etc. Such tools use different formats of files (i.e. dxf, shp, dwg, dae, etc.) to import and export spatial data. File formats are designed to optimize data exchange. But, they contain some anomalies (i.e. redundancy, lack of topology). They are not suitable, with their current format, to be integrated in GIS. They must be cleaned, restrctured and enriched by topological and thematic data. With this purpose, this work consists in the test of different file formats, their restrcturation, their integration in a PostGIS/PostgreSQL database and their interrogation for the visualization and analysis. Source data concern some building of the Montpellier city (France). The main steps of the project are described in the Figure 3.

Testing different file formats is based on several criteria, i.e. the number of dimensions (2D or 3D), the size of data, the topological and thematic data interpreted from these files, etc. To carry out different tests, PostgreSQL and PostGIS are installed on Windows and Linux platforms. **dxf2postgis** existent tools generate 2D data, so it is transformed into shape format for the reason that **shp2pgsql** generates 3D data. And concerning Collada files, **dae2pgsql** generates 3D data as a set of TIN (Triangulated Irregular Network). Figure 4 describes all files formats and the result of their integration in a PostGIS database.
Since the original format of such files is not suitable with the data structure of the 3D GIS, the PostGIS database is transformed into the SDB format. The conceptual model of the SDB will be described in the next section. The main steps of the cleaning and restructuring of PostGIS data are: i) Points identification. ii) Polygons identification. iii) Solids identification. iv) Topology identification. v) Thematic data integration.

The Approach Principle

Since buildings are composed of a set of walls, the 3D conceptual model will be based on the concept of “wall” as a unitary geometric object. It has different geometric representations depending on the LOD (Level Of Detail). After, the assembling of walls into a building is based on topologic constraints. Figures 5 and 6 illustrate the possible geometric representations of a wall which can be:

- A surface or a composed surface: Walls, doors and windows are represented by surfaces.

- A solid or a composed solid: Walls are no longer described by surfaces, they are rather solids having a non negligible thickness. Wall thickness is noticed through their detail objects (doors, windows, intrusions…).
Spatial Data Modeling

`shp2pgsql` generates one big table in the PostGIS database. Solids are represented through polygons whose vertices coordinates are known. Their modeling with surface-based representation can be very useful. With this purpose, B-Rep is used in this work to model buildings for the 3D GIS.

In this section, some conceptual models in the form of UML diagrams are proposed. There exist two types of models: geometric and topological models.

- **Geometric model**: The geometric model is defined in the UML diagram as illustrated in Figure 7.

There are two complementary representations for 3D objects: B-Rep and CSG representations. Four principal B-Rep geometric objects are used: `Point`, `Line`, `Surface` and `Solid` (these primitives are used in CityGML, ISO19107, 3D FDS, etc.). A `line` is defined by two end `points`. Many `lines` that share an end `point` compose a `linestring`. `Linestring` is used in ISO19107, GML and CityGML. In this paper, a `linestring` can be a `closed_linestring` or an `open_linestring`. `Surface` is defined by a `closed_linestring` (contour) and can be orientable (has two faces: front and back) or not (one face is visible). In TEN and UDM, a surface has two faces. In CityGML, only orientable surfaces are introduced. But, in this project, a `surface` can be orientable or not, i.e. the surface of wall modeled as a full solid is non-orientable. A `surface` can be full or empty. The notion of `empty_surface` is introduced in this paper to model some objects such as openings which can be equivalent of the notion of hole in Oo-model [4]. Many surfaces in the same plan form a `composite_surface` i.e. a façade and its doors and windows. A B-Rep solid is defined by a set of boundary surfaces. It can be full (e.g. wall) or empty (e.g. building body). An `empty_solid` is composed of a set of

![Figure 6. Solid-based representation of walls, doors and windows of a building.](image)

![Figure 7. Geometric Conceptual Model (UML Diagram).](image)
orientable surfaces. However, a full solid is composed of a set of non_orientable_surfaces. The interior of an empty solid can be modeled. However, the interior of a full solid cannot be modeled. On the other hand, CSG primitives can be cubes, cylinders, spheres, etc. A cube is obtained by the extrusion of a B-Rep surface. A CSG composite solid is the union, intersection or difference of two solids. The interior of such solids is not modeled. In the proposed model, B-Rep and CSG modeling approaches are combined. B-Rep allows representing solid as a set of surfaces sharing common edges. It gives information about solid boundaries (vertices coordinates, contour, etc.). On the other hand, CSG allows representing a wall as an indivisible solid. It is used when no information about the solid boundaries are recommended. The motivation of the combination of B-Rep and CSG in the same model is: i) B-Rep defines very well the exterior boundaries of the building trough surfaces sharing common edges. So that geometric transformations (translation, rotation, etc.) of CSG solids are not necessary because the B-Rep surfaces extruded are well placed and directed in the space. ii) Obtaining CSG solids by extrusion is more intuitive than the construction of all boundary surfaces of a solid. iii) More details on building means more surfaces to represent. Solid-based representation allows representing such detail with the minimum of data stored in the database.

• Topological model

Topological conceptual model is based on three Boolean operations: Union, Difference and Intersection of geometric objects. Such operations are applicable on the four geometric objects mentioned in the geometric model (figure 11). To simplify, a class diagram is defined for each Boolean operation.

Union

![Figure 8. Topological Model: Union Operation.](image)

In this model, the concepts of multisolid, multisurface, multiline and multipoint of CityGML [11] are used. Union operations can be achieved only between similar objects. For example, the union of a solid and a point is meaningless. The union of two or many geometric forms generates a group of objects that can share or not common parts.

Difference

![Figure 9. Topological Model: Difference Operation.](image)

In this model, like the union operation, difference operation can be carried out only between similar objects. We are interested only in solid and surface objects. For example, to obtain an opening in a wall, there are two cases: 1) If the wall is a surface, the difference is calculated between two surfaces (corresponding respectively to the wall and the opening). 2) If the wall is a solid, the difference is calculated between two solids (corresponding respectively to the wall and the opening). Difference operation is used also to add some details on walls (e.g. intrusion, extrusion, etc.).
Intersection

![Figure 10. Topological model: intersection Operations.](image)

Detecting intersection between city objects is very important. It shows their common parts. Table 1 explains the results of intersection of the different geometric objects.

It’s also important, in an urban space, to define the neighborhood relationships (adjacencies, distance to 1 or n neighbors, etc.) between buildings or groups of buildings. The relationships between two buildings can be:

- Physic relationship: buildings can share common parts (generally intersection) or being geographically adjacent.
- Semantic relationships: sharing the same properties (e.g. located in the same residential area, the same city, etc.).

<table>
<thead>
<tr>
<th>Intersection</th>
<th>Solid</th>
<th>Surface</th>
<th>Line</th>
<th>Point</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Solid</strong></td>
<td>Solid, surface, line, point</td>
<td>Surface, line, point</td>
<td>Line, point</td>
<td>*</td>
</tr>
<tr>
<td><strong>Surface</strong></td>
<td>Surface, line, point</td>
<td>Surface, line, point</td>
<td>Line, point</td>
<td>*</td>
</tr>
<tr>
<td><strong>Line</strong></td>
<td>Line, point</td>
<td>Line, point</td>
<td>Line, point</td>
<td>*</td>
</tr>
<tr>
<td><strong>Point</strong></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>

Modeling Algorithm

The algorithm consists in the surfaces extrusion to get solid walls. Since polygons vertices coordinates are known, the extrusion algorithm is based on these coordinates. The idea is very simple: two vectors of each surface are calculated, and then their vector product is calculated and normalized. As the extrusion must be always calculated forward the building interior as described in Figure 11, surface vertices are ordered counter clockwise.

![Figure 11. Surface extrusion.](image)

The extrusion is done vertically to the plan of the surface to extrude. But, surface is generally interconnected to other surfaces. The extrusion of many interconnected surface can generate invalid shapes. Table 2 explains the different case of intersection of two surfaces and the result of their extrusion.
Table 2. The result of 2 interconnected surfaces depending on their intersection angle

<table>
<thead>
<tr>
<th>Angle = 90°</th>
<th>Angle &gt; 90°</th>
<th>Angle &lt; 90°</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="ss.png" alt="Image" /></td>
<td><img src="ss.png" alt="Image" /></td>
<td><img src="ss.png" alt="Image" /></td>
</tr>
</tbody>
</table>

Extrusion (Wall thickness)

| Valid shape | Valid shape | Invalid shape |

This table shows that when the intersection angle between two surfaces is less than 90°, the shape obtained by the two surfaces extruded will be invalid. Such shape has to be corrected as described in Figure 12. Surface 2 is used as a section plan for the solid obtained from the extrusion of surface 1.

![Image](ss.png)

**Figure 12. invalid shapes correction.**

The main steps of the extrusion algorithm are described as follows:

<table>
<thead>
<tr>
<th>For each surface of the solid Begin</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Calculation of two vectors corresponding to three points of the surface: ( \vec{V}_1 ) for P₁ and P₂, and ( \vec{V}_2 ) for P₁ and P₂: ( \vec{V}_1 = (x_2-x_1, y_2-y_1, z_2-z_1) ) and ( \vec{V}_2 = (x_n-x_1, y_n-y_1, z_n-z_1) )</td>
</tr>
<tr>
<td>2) Calculation of the vector product ( \vec{V}_3 ) of ( \vec{V}_1 ) and ( \vec{V}_2 ): ( \vec{V}_3 = \vec{V}_1 \wedge \vec{V}_2 )</td>
</tr>
</tbody>
</table>
| 3) Normalization of the vector \( \vec{V}_3 \) to obtain a final vector \( \vec{V} \):
  \[
  \vec{V} = \left( \frac{x}{\sqrt{x^2 + y^2 + z^2}}, \frac{y}{\sqrt{x^2 + y^2 + z^2}}, \frac{z}{\sqrt{x^2 + y^2 + z^2}} \right)
  \]
| 4) For each vertex of the surface Begin
  \[ P_i = P_i + \vec{V} \times \text{solid thickness} \]
  End
| 5) For each edge of the surface Begin
  If Surface intersects another surface and intersection angle < 90°
  Correct shape with a section plan
  End If
  End
| End
| End
Discussion

The aim of this work is the establishment of a 3D GIS for buildings. All building information (geometry, topology, attributes, etc.) are stored in a SDB. Such 3D GIS have two main objectives: i) The minimum of data are stored in the SDB. ii) The 3D model has to present the maximum of realism. With this purpose, models proposed in this work are very simple. A light SDB means fast data querying and simple data interpretation. The SDB must be also extensible. It can be enriched by complementary data when other analysis needs are required.

It is also necessary to justify the interest of linking the surface-based and solid-based representations in the same 3D model and why building walls are extruded. We believe that every component of the building (walls, windows, doors, extrusions, etc.) has in the reality a non negligible thickness. It can so be represented by a solid. Therefore, the concept of surface cannot be very meaningful if more details are required. Thus, the idea consists in the representation of buildings with B-Rep in low LODs and with CSG in high LODs.

EXPERIMENTATION AND RESULTS

To test the approach, a shape file of the Montpellier city is exported to the SDB. Then a Java application is realized to visualize and query the SDB. Some extracts of the 3D model of Montpellier city are shown in Figure 13.

Figure 13. Spatial data viewer for the 3D GIS (Buildings and DTM- Digital Terrain Model).

PostGIS data are stored in the following manner: Surfaces are represented by polygons. Polygons are represented by their vertices coordinates. There is a primary key per polygon, but points haven’t primary keys. Points are represented by their coordinates. If a point p (x, y, z) belongs to two surfaces, it will be mentioned twice. When loading this data from the PostGIS database to the SDB, polygons and points are identified. Then the relationship between a polygon and its vertices is based on their primary keys. The next Figure shows a statistic about the export of a shape file to a PostGIS database, and then data exported from this PostGIS database to our SDB containing a PostGIS layer.

Figure 14. Data storage comparison between PostGIS and our SDB.

The Figure 14 shows that the same points are stored many times in the PostGIS database. In the SDB, each point is stored only once and data are optimized. In a building, walls have a non negligible thickness which can be noticed by doors and windows falling within the wall.

With surface-based representation, windows and doors are obtained by the Boolean intersection of two surfaces (surface corresponding to the wall and surface corresponding to the window or the door). However, with solid-based representation, windows and doors are obtained by the Boolean difference between two solids (solid corresponding to the wall and the solid corresponding to the window or the door). The Figure 16 shows doors and windows existing on a wall modeled as a surface and as a solid.

Table 4 demonstrates that only with two added data (wall and roof thicknesses) surfaces are transformed to solids and so more realism. One might ask: the extrusion of a surface generates several other surfaces from which some are hidden, so can this decrease the performance of the calculation? The response is no because only visible
surfaces are calculated. Otherwise, hidden surfaces are calculated only when they appear in the screen. So the performance of the application will not be affected by the number of surfaces. If we try to represent the building walls as solids with surface-based representation and compare it with solid-based representation, we obtain the Table 5.

![Building modeled with Surface-Based Representation (left) and Solid-based representation (right).](image)

**Figure 15.** Building modeled with Surface-Based Representation (left) and Solid-based representation (right).

![Doors and windows on a wall represented as a surface (left) and as a solid (right).](image)

**Figure 16.** Doors and windows on a wall represented as a surface (left) and as a solid (right).

With a pragmatic point of view, the building shown in the table is stored in the database as follows:

<table>
<thead>
<tr>
<th>Table 4. building storage in the database</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Surface-based Representation</strong></td>
</tr>
<tr>
<td>Facades</td>
</tr>
<tr>
<td>Roofs</td>
</tr>
<tr>
<td>Doors</td>
</tr>
<tr>
<td>Windows</td>
</tr>
<tr>
<td>Total</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 5. Building details represented with surface-based and solid-based representations</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Surface-based Representation</strong></td>
</tr>
<tr>
<td>Facades</td>
</tr>
<tr>
<td>Roofs</td>
</tr>
<tr>
<td>Doors</td>
</tr>
<tr>
<td>Windows</td>
</tr>
<tr>
<td>Total</td>
</tr>
</tbody>
</table>
This table proves that if B-Rep is used to represent more details on the building, there are many surfaces to store in the SDB. However, with CSG, such details are interpreted from the initial model and the data storage is optimized.

**Limits**

Because 3D spatial data are huge, the data loading from the SDB to the Java application for visualization and interrogation is slow (it becomes slow for models containing more than 10000 polygons). This problem can be resolved if data are loaded part by part. Using a parallel programming (threads) can be also useful to optimize data interrogation. Solving this problem is one of our future works.

**CONCLUSION AND FUTURE WORK**

This work aims the establishment of a 3D GIS for buildings. Since data modeling is the first stage in the creation of all information system, this paper presents a simplified hybrid approach of spatial data modeling. It combines surface-based and solid-based representations. Experimentation proves that the linking of such approaches has brought many advantages for both data visualization and data analyses. We believe that the model proposed in this paper is not the most complete and not the most optimized, but it reflects a personal point of view of building in a 3D GIS. It meets the majority of the application needs and can be enriched when other analysis are required. The contribution of this work is not the extrusion because it is a technique widely used in CAD tools. The main contribution is the use of this technique to pass from surface-based to solid-based representation. The future work consists in the enrichment of geometric and topological data by thematic information according to the analysis needs.

**REFERENCES**

Molenaar, M., 1990. A formal data structure for three dimensional vector maps, 4th International Symposium on Spatial Data handling, Zurich, Switzerland.


Zlatanova, S. 2000. 3D GIS for Urban development, PhD thesis at Graz University of Technology.


http://qxx.planetquake.gamespy.com/bsp
http://www.webopedia.com/TERM
http://www.yourdictionary.com