

# RESEARCH ON OBJECT-ORIENTED THREE DIMENSIONAL DATA MODEL

Xiaojun Tan<sup>a</sup>, Fuling Bian, Jun Li

Wuhan University,  
School of Remote Sensing and Information Engineering, Wuhan, 430079, China

<sup>a</sup> txj72@263.net

## ABSTRACT:

3D GISs were developed simultaneously in the late 1980s in a number of different disciplines. The development of these 3D GISs can not yet meet all needs for new representations and analytical tools in 3D environments. In this paper, the data structures, especially solid-based data representations are introduced. Then, a typical object-oriented data model is put forth to show a new data model. And at last, two data access methods are proposed to make the data model clear.

**KEY WORDS:** 3D GIS, Object-Oriented, Spatial Data Model, Data Structure

## 1. Introduction

3D GISs were developed simultaneously in the late 1980s in a number of different disciplines. Especially in hydrocarbon exploration and mining engineering (Raper, 1989; Turner, 1992) various commercial systems have been developed to meet the specific needs in these fields (e.g. Earth Vision, gOcad, Stratamodel and Lynx) and are widely used in other fields now. However, the development of these 3D GISs can not satisfy all needs for new representations and analytical tools in 3D environments. Hence a variety of new systems has been developed in the research community, for example, to create tools for the superimposition of 3D objects on terrain (Beckmann, 1990), to develop techniques of 3D octree representation (Prissang, 1992), and to develop systems to handle heterogeneous 4D databases (O'Conaill et. al., 1992).

Given the wide range of application where 3D GIS can be used, it is clear that much more research is required. Examples of necessary work include the integration of new data types into 3D GIS models and better object-oriented data models in the 3D environment so that users can manipulate and extend models easily. In this paper, the data structure, especially solid-based data representation is introduced at first. Then, a typical object-oriented data model is issued to show a new data model. And at last, two data access methods are proposed to make the data model clear.

## 2. 3D Data Structure

3D geometric representations provide geometric descriptions of objects for storage, geometric processing, display, and management of 3D spatial data by computers in GIS. According to the geometric characteristics of data structures, 3D geometric representations to objects could be classified into two categories: surface-based representation and solid-based representation. For surface-based representations, geometric characteristics of objects can be described by micro surface cells, or in other words, surface primitives. These representations are characterized by Grid method, Shaped model, Boundary representation, and Facet model. The outside looking, instead of inside looking, is emphasized and can be used in applications such as terrain, and other objects with well-proportioned entrails. As for another one, which describes the interior of objects by using solid information, instead of surface information, solid-based representations are typically characterized by methods such as 3D Binary Arrays and Needle Models, Octree Models, and Constructive Solid Geometry (CGS), etc.

### 2.1 3D Binary Arrays and Needle Model

A 3D array has its elements of either 0 or 1, where 0 means the background and 1 indicates the occupation of the element by objects. Suppose that an object is scanned in the 3D binary array whose elements are initialized with 0. This scanning procedure results in a 3D binary array with its elements of value 1 representing solid information of the object. The higher the scanning resolution, the greater the dimension of the array. Thus in case of large volume of data the high resolution makes the data handling difficult. Therefore, an effective encoding and compress algorithm is necessary to represent and store 3D objects. Needle Model is often used to represent solid information, for example, multi-layers of geological subsurfaces.

## 2.2 Octree Model

A more compact and efficient solid-based representation is an octree (Samet, 1990) which is an extension of quadtree. The octree representation describes an object hierarchically. As a general octree, an original octant is defined by the smallest cube containing the object. At the first level this original octant is divided into 8 sub-octants by halving the original octant in three directions. Each sub-octant is then checked to decide if it is occupied by the object. The sub-octants are classified into three categories: 1) F=FULL (occupied by the object); 2) E=EMPTY (no object elements in the sub-octant); and 3) P=PARTIAL (partially occupied). P-octants will be further subdivided into 8 sub-octants at the next level, which are classified furthermore. The partition procedure continues until all sub-octants are either F-octants or E-octants. The above three are all approximate representations of spatial objects, with the increase of spatial resolution, data volume increase dramatically, but the computation of Octree Model decreases much in comparison with other models because of its simple structure and convenient operation.

Compared to the general octree, a linear octree, which is developed to overcome the disadvantages hidden in the general octree, only stores F-octants and their contents including location, size, and attributes. Usually Morton code is used to encode the address of octants. Suppose a spatial object with dimension as  $2^n \times 2^n \times 2^n$  and resolution as 1, then the Morton code in any octant can be represented as :

$$M_q = q_{n-1}8^{n-1} + \wedge + q_i8^i + \wedge + q_08^0$$

While  $q_i = \{0,1,2,3,4,5,6,7\}$

In 3D data structures, 3D Run Length Encoding, which is an extension to 2D Run Length Encoding, is used to represent the data structure. In 3D Run Length Encoding system, octants can be encoded like in octree, i.e. Morton code. When using decimal system, the code (actually numbers) can be linked as a 1D continuous natural number queue, and furthermore, the sequence of the numbers describes the spatial neighborhood relationship between octants. Therefore, 3D Run Length Encoding can be treated as the further compression of a linear octree, and any compressed element can be retrieved from its neighbor elements (Li, 1997).

Compared to linear octree, 3D Run Length Encoding can save more storage space. Since it uses natural number as sequence labels, the operation such as query, insert, delete, etc. can be speeded up and simplified. Although it shares many characteristics of octree, it still lacks some structural features of octree. Therefore, it can be only used for voxel objects representation.

## 2.3 Constructive Solid Geometry(CSG)

A representation of Constructive Solid Geometry (CSG) represents an object by a combination of predefined primitives which are regularly shaped in volumetric instances such as cubes, cylinders, cones and complex primitives. Relationships among these primitives include geometric transformations and Boolean operations (and, or, not, etc. ). Usually, a CSG has a tree structure where leaf nodes correspond to Boolean set operations and root nodes correspond to query and index operations.

In one word, solid-based geometric representations of 3D binary arrays, Needle Model and octree are capable of representing objects with regular shapes. Conversely, CSG representations are well suited for irregularly shaped objects through combinations of tiny regularly shaped ones (Li, 1994).

## 3. 3D Data Model

Object-oriented technique has been proven to be an excellent tool for data modeling and it has been widely used in this area. Following the object-oriented modeling technique (Rumbaugh et al., 1991), not only abstract geometric primitives, such as points, curves and surfaces but also the real world entities such as drilling wells, geological sections can be modeled and maintained. In 2D data model, object-oriented technique has been used, and now this method may be extended into 3D data situations, in which two basic notions will be primarily dealt with: spatial objects and a collection of spatial objects which have a pointer to a space. On the abstract level a spatial object is defined as a point set in the 3D Euclidean space. Various geometric operations can be applied to a spatial object. There is a direct analogy with the object-oriented modeling capabilities. A concrete object is modeled as a specialization of the abstract spatial object class that is extended by some additional features (representation). The spatial object class specifies only the interface, which is inherited by all concrete spatial objects. A concrete class provides an appropriation for the object as well as the implementation of the functions.

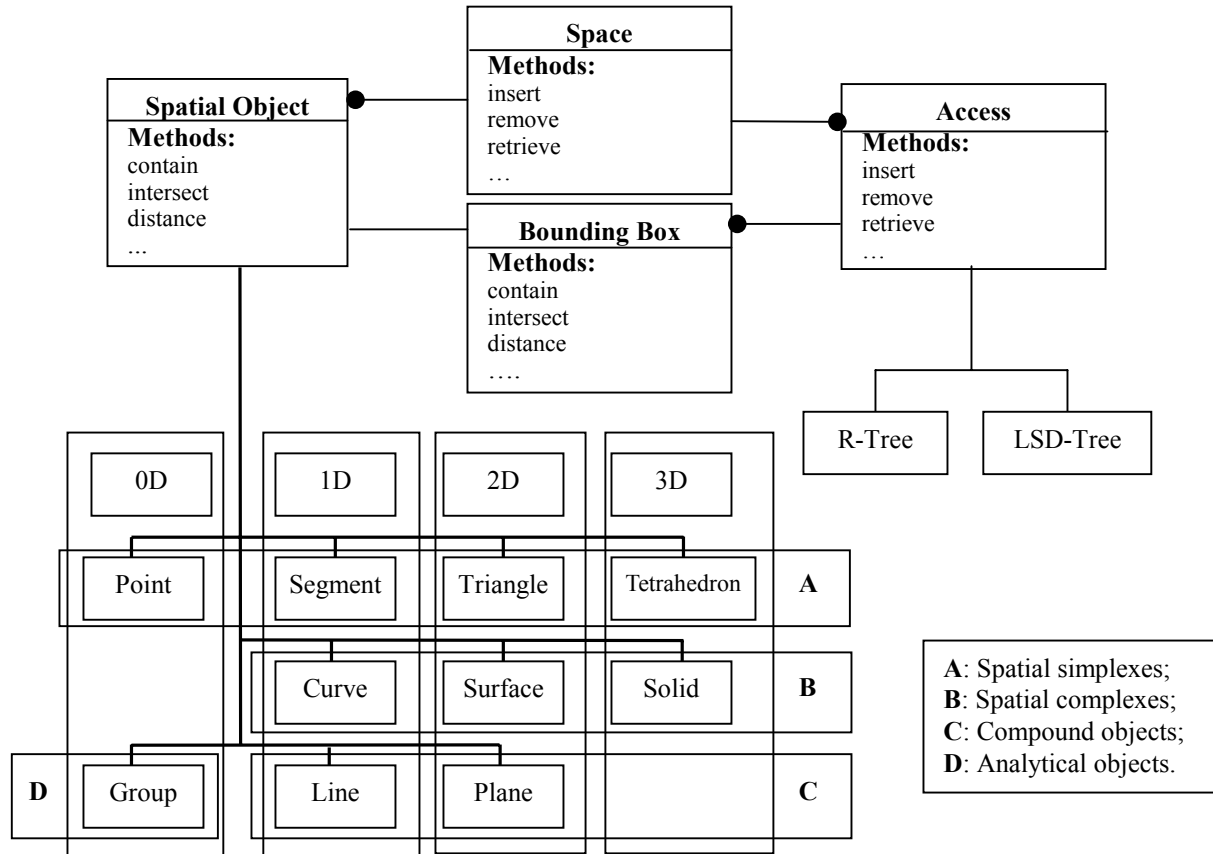


Fig1. Object-Oriented data model

There are 4 kinds of objects in Fig.1 that represent different representations respectively in the real world: spatial simplexes, spatial complexes, compound objects and analytical objects. Usually complexes are approximated and represented as homogeneous collections of simplexes. A curve (1-complex), for example, is approximated through a polyline, a surface is represented as triangle network and a solid as a tetrahedron network. On the other hand, spatial objects of different types can be collected into a heterogeneous integration, a group, which is further treated as a single object. Here a group is a necessary construction to represent the results of geometric operations. 3D data model should define descriptions on geometry, semantics and topology of complexes or objects, these include descriptions on data index and spatial relation. Data index points out how to trace all elements in a set from a given node or condition. Spatial relation supplies deductive evidences for this process. On the implementation level a spatial object and its representation could be different. A spatial object exists independently of the objects, it may be accessed from other objects, while a representation cannot be referred to other than through the object itself. As we know, an object may need multiple representations, e.g. one for the compact storage, another for the efficient computation. Therefore, the ability to have multiple representations and to change them without changing the object identify is extremely important in the database context since the object can be accessed from multiple sources.

There are three main groups of spatial access methods: 1) methods which transform rectangles into points in a space of higher dimension; 2) methods which use quadtrees or other space-filling structures; and 3) methods using trees. The third way that uses trees is very popular now and has been proven a very efficient way to organize 3D data model.

### 3.1 R-Tree

R-tree (Guttman, 1984) is an extension of B-tree (Bayer, 1972) that store multi-dimensional data, and is generally accepted to be one of the best data structures for range and point queries (which involve seeing what shapes intersect a rectangular range).

R-tree has been widely used in spatial data access and management. It is well known, for example, that processing spatial join queries is an extremely I/O and CPU expensive process, while the new methods using R-trees for spatial joins has a significant performance improvement (up to 50%) over the state-of-the-art approach (Brinkhoff, et al., 1993). Using R-trees for spatial join processing is so effective that R-tree (and its variants) has become a very popular spatial index structure, e.g., Illustra, Intergraph's GIS databases, Postgres, and MapInfo all offer R-tree support. Furthermore, many spatial operations such as join processing based on R-tree has been shown superior performance as compared with alternate index structures (Patel et al., 1996). Many access structures based on R-tree, such as SS-tree, the VAMSplit R-tree, the TV-tree, the SR-tree and the X-tree, etc (Henrich, A., 1998) have been developed. While dealing with high dimensional data, R-tree is not a so efficient access structure.

### 3.2 LSD-Tree

As another tree-based data access methods, the Local Split Decision tree (LSD tree) (Henrich, 1998), a data structure supporting efficient spatial access to geometric objects is used for data modeling, especially in 3D data fields. Its main advantages over other structures are that it performs well for all reasonable data distributions, cover quotients (which measure the overlapping of the data objects), and bucket capacities, and that it maintains multidimensional points as well as arbitrary geometric objects. These properties demonstrated by an extensive performance evaluation make the LSD tree extremely suitable for the implementation of spatial access paths in geometric databases. The paging algorithm for the binary tree direction is interesting in its own right because a practical solution for the problem of how to page a (multidimensional) binary tree without access path degeneration is presented.

## 4. Conclusion

According to the disparity of the objects which need to be represented, there are different representations to describe them. Associated with the methods, the space for storage, efficiency for compression and retrieval, and simplification for manipulation and visualization, are different as well. It is worth noting that it depends on the goals of the application to choose the right data structure to describe 3D objects. As for data model to organize different 3D object types, the method based on object-oriented is the tendency for the future applications. Furthermore, it ties tightly with the future database, i.e. Object-oriented Database System (OODB).

### Reference:

- Bayer R, McCreight EM(1972) Organization and Maintenance of Large Ordered Indices. *Acta Informatica* 1:173-189
- Beckmann N, Kriegel HP, Schneider R, Seeger B (1990) The R\*-tree: An Efficient and Robust Access Method for Points and Rectangles. In: *SIGMOD Conference 1990*
- Brinkhoff T, Kriegel HP, Seeger B (1993) Efficient Processing of Spatial Joins Using R-Trees. In: *SIGMOD Conference 1993*
- Guttman A (1984) R-Tree: A Dynamic Index Structure for Spatial Searching. In: *SIGMOD Conference 1984*
- Henrich A (1998) The LSD-h-Tree: An Access Structure for Feature Vectors. In: *Proceedings of the 14th International Conference on Data Engineering (ICDE'98)*. IEEE, Orlando, Florida, USA, Feb, 1998. pp 362-369,
- Kraak MJ (1992) Working with Triangulation-based Spatial Data in 3D Space. *ITC Journal* 1:20-24
- Li R (1994) Data Structure and Application Issues in 3D Geographic Information Systems. *Geomatica* 48(3): 209-224
- Li QQ, Li DR (1997) Three-Dimensional Run-Encoding(3DRE) for Octree. *Journal of Wuhan Technical University of Surveying and Mapping* 22(2): 102-106
- O'Conaill MA, Bell S, Mason D (1992) Developing A Prototype 4D GIS on A Transputer Array. *ITC Journal* 1: 47-54
- Patel JM, DeWitt DJ (1996) Partion Based Spatial-Merge Join. In: *SIGMOD Conference 1996*

Prissang R (1992) Three-dimensional Predictive deposit Modeling based on the Linear Octree Data Structure. In: Computer Graphics in Geology, Lecture Notes in Earth Sciences 42, Berlin

Rambaugh J, Blaha M, Premerlani W, Eddy F, Lorensen W (1991) Object-oriented Modelling and Design. Prentice Hall, New Jersey

Raper JF (1989) Three Dimensional Applications in Geographic Information Systems. Talyor and Francis, London

Samet H (1990) The Design and Analysis of Spatial Data Structures. Addison-Wesley Publishing Company, Inc., Reading, Massachusetts

Turner K(1992) Three Dimensional Modeling with Geoscientific Information System. Kluwer, Dordrecht