ALGORITHMS FOR TETRAHEDRAL NETWORK (TEN) GENERATION

LI Qingquan
LI Deren

KEY WORDS  3-D GIS; tetrahedral network (TEN); generation algorithm

ABSTRACT  The Tetrahedra Network (TEN) is a powerful 3-D vector structure in GIS, which has a lot of advantages such as simple structure, fast topological relation processing and rapid visualization. The difficulty of TEN application is automatic creating data structure. Although a raster algorithm has been introduced by some authors, the problems in accuracy, memory requirement, speed and integrity are still existent. In this paper, the raster algorithm is completed and a vector algorithm is presented after a 3-D data model and structure of TEN have been introduced. Finally, experiment, conclusion and future work are discussed.

1 Introduction

A full 3-D GIS is needed in many geoscience application fields, such as geology, mine, oil and environmental engineering, etc. Not only modeling and visualization but also manipulation are required in these applications. But the traditional 2-D GIS has difficulty in satisfying these requirements, especially in the representation of vertical information. Although a number of studies have been done by researchers in the world (Molenaar, 1992; Raper and Kelk, 1991; Li Rongxing, 1994; Gong Jianya, 1996, et al.), there are not sufficient development in 3-D GIS. The reasons are the complexity of 3-D geometry and topology and the lack of powerful 3-D data model and structure.

For a long time, people worked on the raster structure, e.g., octree, and, data structures coming from Computer Aided Design (CAD), e.g., Boundary Representation (BR), Constructive Solid Geometry (CSG), have been more discussed. Due to the difference between CAD and GIS, it is obvious that these data structures can not readily be applied to geoscience modeling.

As we know, the raster structure in GIS has two disadvantages: the one is too low position accuracy, the other is without representation for topological relations which are necessary for some spatial analysis. In recent years, much attention is paid to the 3-D vector structure (Edmund Sides and Robert Hack, 1995; Morakot Pilouk et al., 1994; Chen Xiaoyong, 1994). Among these studies, there are a lot of works in Tetrahedral Network (TEN) (3-D TIN) which was firstly proposed by Carlson in 1987 and concerned as a useful 3-D vector data structure in GIS.

The concept of TEN can be readily formed from a 2-D TIN. After an extension, the 2-D Voronoi diagram is extended to 3-D, then the TEN can be derived from 3-D Voronoi polyhedrons in the same way as deriving the TIN from 2-D Voronoi polygons. Comparing with other solid models, the TEN has some advantages, such as simple structure, easy geometric transformation, fast topology relation processing and rapid visualization.

The TEN can be divided into two types: the one is ordinary TEN, the other is constrained TEN.
Similar to the 2-D TIN, there are three kinds of constraints which are line constraint, e.g., break line on earth surface, surface constraint, e.g. sub-surface of geological layer and volume constraint, e.g. excavated area in mine engineering. Actually, the line and surface constraint can be considered as one kind because one surface constraint can be formed by a set of line constraints.

In practical application, the key problem is automatic creating algorithm of the TEN structure. Although a raster algorithm has been proposed by Morakot Pilouk et al. (1994) and Chen Xiaoyong (1994), some problems in this algorithm on accuracy, memory requirement, speed and integrity still exist. In this paper, to solve these problems a vector algorithm of the TEN formation is proposed. The paper is organized as follows; A 3-D data model and the structure of TEN are introduced in section 2. The raster algorithm is completed and discussed in section 3. A vector algorithm is presented in section 4. Finally, some experiments, conclusions and future work are discussed in section 5.

2 3-D data model and structure of TEN

2.1 3-D Data Model

A comprehensive 3-D data model, consisting of geometric and semantic description of 3-D objects, allows for the processing of 3-D topology and includes raster and vector structures, it can get data from the existing 2-D GISs after data conversion and is convenient for 3-D visualization. After an extension of 3-D FDS(Molenaar, 1992), a 3-D data model is presented in Fig. 1 in which the TEN is integrated. In this data model, position, shape and size, and topology information of the objects can be represented.

According to the geometry, spatial objects in 3-D GIS can be classified into four kinds of objects: point, line, surface and body. The geometric primitives and their description are listed in Table 1.

The position information is contained in nodes. The shape and size information is included in line objects, surface objects and body objects. The topological relations between objects are defined with the geometric primitives and the links between them. The raster structure is required in the data model for some spatial analyses and data forming some resources, and the conversion between vector and raster exists in the data model.

| Type | Dimension | Description |
|------|-----------|-------------|
| node | zero      | point objects and parts of line objects |
| arc  | one       | line objects and boundaries of faces |
| face | two       | surface objects and boundaries of bodies |
| body | three     | body(solid) objects |

Table 1 3-D geometric primitive

2.2 Structure of TEN

In the 3-D data model described in this section, the TEN is integrated. The topological data structure of the TEN can be implemented in a relational database and a C-like description which is used in this research is listed as follows.

```c
struct POINT{
    float xyz[3]; /* X, Y and Z coordinates */
    long next; /* Next POINT in block */
};
```
struct EDGEI
    long nn[2]; /* From-node and To-node */

struct TRIANGLE
    long en[3]; /* Three ENDS of Triangle */
    long lr[2]; /* Left and Right of TENs */

struct TEN
    long tn[4]; /* Four TRIANGLES of Tetrahedron */
    long at[4]; /* Four adjacent Tetrahedrons */
    float xc[3]; /* X, Y & Z of circumcenter */
    float r[2]; /* Square of circumradius */

3 Raster Algorithm for TEN Formation

So far, the raster algorithm is the only way used to form the TEN structure from a set of 3-D points. The basic idea is that the 3-D space is represented by a 3-D array completely and the spatial points are represented by voxels after the conversion of a vector to raster, and a 3-D Voronoi diagram is formed by means of a Distance Transformation (DT) (Borgefors G, 1984), then the TEN structure is derived from the 3-D Voronoi diagram.

3.1 Algorithm

Once a given set of points is mapped onto a 3-D array by a vector to raster conversion the 3-D Voronoi diagram can be constructed by a DT. A DT is an operation that converts a binary picture (2-D or 3-D), consisting of feature and non-feature elements, to a picture where each element has a value that approximates the distance to the nearest feature element. Different distances can be used for a DT. According to Borgefors (1984), the Chamfer (3,4,5) distance approximates the Euclidean distance quite well and only a local distance mask of 3 x 3 x 3 voxels is required. In the DT procedure the 3-D array is initialized by two values: zero for feature voxel and infinity otherwise. In the feature array the point number is assigned to each feature voxel and a unique value to background voxels. While the DT is applied to the distance array, the background voxels in the feature array are overwritten by feature ones. As a result, a 3-D Voronoi diagram is formed. The points are mapped onto two arrays as voxels. The distance array is initialized by two values: zero for feature voxel and infinity otherwise. In the feature array the point number is assigned to each feature voxel and a unique value to background voxels. While the DT is applied to the distance array, the background voxels in the feature array are overwritten by feature ones. As a result, a 3-D Voronoi diagram is formed. When the 3-D Voronoi diagram of given points is available, the TEN can be derived by use of a 2 x 2 x 2 operator. The details explained in the work of Morakot
Before our study, only the ordinary TEN and the volume constrained TEN can be formed. There are difficulties in the formation of line constrained TEN. The reason is that line constraint is not implemented by use of the method similar to Tang L. (1992) during the DT procedure. To solve this problem a Dynamic Constraint Method (DCM) is developed by authors in which the line constrained TEN can be formed. The principle of DCM is that constraint increases within the DT procedure. It can be explained by a simple example of 2-D line constraint in Fig. 3.

![Distance array](image) ![Voronoi array](image)

Fig. 3 Dynamic constraint method (DCM)

In the distance array, "0" represents the end point of constraint. The value of "d" depends on the density of points and the size of voxel. In the Voronoi array, "2" and "3" represent the point numbers. During the DT procedure constraint line is growing.

3.2 Discussion

According to our study in the raster algorithm there are three problems which can not be solved by the algorithm itself:

a) A 3-D point must be converted to a voxel representation in the algorithm. Because the size of voxel, in some case, is large and a point may be situated at any position of the voxel, TEN is an approximation and the larger the voxel is, the lower the accuracy is.

b) When the 3-D points have a non-uniform distribution, in order to separate the points which are near to each other, the voxel must be a smaller one. It leads to the slower speed of DT and requirements of the larger computer memory. Therefore, the processing of a larger number of points is impossible.

c) While a tetrahedron of TEN with a special shape, e.g. it is thin, in the boundary of polyhedron the intersection of Voronoi diagram may be out of the Voronoi array. In this case, the tetrahedron may not be formed and the TEN is of unintegrity.

4 Vector algorithm for TEN formation

A lot of vector algorithms for the TIN generation have been proposed. According to the procedures in creation, the algorithms can be classified into three categories: (1) the divide-and-conquer methods; (2) the incremental insertion method; (3) the triangulation growth method. Based on the analyses of these algorithms, a vector algorithm for the TEN generation is developed from the incremental insertion method which has two steps.

4.1 3-D Convex Hull computation

The Convex Hull of a set of 3-D points is defined to be the smallest Convex polyhedron constructed by a Delaunay triangulation containing all points and it has the property that any triangle connecting three points inside the polyhedron must lie entirely inside the polyhedron. It is the natural extreme boundary of the point set and a part of the TEN. At first, partitioning the set of points into $N/k$ blocks, where $k$ is the selected average number of points per block (default $k = 5$), which enables the points to be accessed fast. Because the 3-D sorting is lavish and unnecessary in the algorithm that they are to be accessed in the same order. A one-dimensional array is used to store the index of the first point in each block, then store the index of next point lying in the same block and so on. Fig. 4 shows the partitioning block and the point structure.

After partitioning the set of points into blocks the 3-D Convex Hull is computed by means of the following procedure:

a) Find four points on the Convex Hull such as maximum $x+y$, $y+z$, $x+z$ and minimum $x+y+z$. Form an initial triangulation. Store these points into a TIN structure and establish topological relations.
b) For each triangle find blocks intersecting within or out of the triangle. The direction of triangle is defined by right hand rule. Calculate the distance to the triangle of the points in these blocks and find the point with the largest distance.

| Index | 1st Pt | Point |
|-------|--------|-------|
| Block | X      | Y     | Z     | Next |
| 0     | 6      |       |       |      |
| 1     | 2      |       |       |      |
| 2     | 1      |       |       |      |
| 3     | 5      |       |       |      |
| 4     |        |       |       |      |
| 5     |        |       |       |      |
| 6     |        |       |       |      |
| 7     |        |       |       |      |

Fig. 4 Block partitioning, index and point structure

c) On condition that the largest distance is positive or zero and the point is projected inside the triangle, three new triangles are created and the original triangle is deleted. Meanwhile the topological relations are modified in the TIN structure. Otherwise no change happens in the TIN structure.

d) Repeat b), c) until no point is found. The points stored in the TIN structure are all points of the Convex Hull.

e) Use the Delaunay criterion for all triangles in the TIN structure, if necessary, apply lawson’s LOP in diagonal swapping. At last the Delaunay triangulation of the Convex Hull is created.

4.2 TEN generation

In order to generate the TEN structure of all points, an initial TEN structure which includes all points on the Convex Hull should be formed firstly. Then all other points inside the Convex Hull can be added interactively into the initial TEN structure. By finding influence tetrahedrons of the added point, existent TEN structure is then locally refined. The procedure is as follows:

a) Determine a triangle of the Convex Hull as a base triangle and store it into the TEN structure. Then find another point on the Convex Hull which satisfies the Delaunay criterion that there is no point inside the circumsphere of four points (three points of triangle and a new one).

b) Form three new triangles in the TEN structure and update adjacent topological relations in the TEN structure.

c) Use each of these triangles as a base triangle, repeat a), b), until all triangles are used. Then the initial TEN structure is established.

d) For a new point in the point set find all tetrahedrons whose circumspheres enclose the new point by use of the Delaunay criterion, form an influence TEN of the new point.

e) Delete all internal triangles and their topological relations of the influence TEN in the TEN structure.

f) Form new triangles and tetrahedrons by connecting the new point to the vertices of its influence TEN. Update adjacent topological relations in the TEN structure.

g) Repeat d), e), f) until all inside points are added. Finally the TEN structure of all 3-D points is generated.

5 Experiment and conclusion

To confirm the research results, some test programs in C Language were developed by authors. The raster algorithm was realized in a HP Workstation and the vector algorithm was performed in a PC computer. Some experiments were performed, such as a building model by the raster algorithm shown in Fig. 5, the TEN structure of 500 random 3-D points by the vector algorithm.

For the representation of 3-D topology in the 3-D GIS, the TEN is an important vector structure which has simple structure, easy for geometric transformation, fast for topology relation processing and convenient for rapid visualization. The algo-
The algorithm of TEN generation is realized in raster and vector methods. The raster algorithm has some disadvantages such as low accuracy, large memory requirement, low speed and unintegrity. Here, a vector algorithm is proposed which can form the 3-D natural boundary of a set of 3-D points. And, it is obvious that the vector algorithm has no disadvantages of this kind. However, the disadvantage of the vector method is complex in program design.

The future work includes the generation of constrained TEN, internal boundary clipping, external boundary clipping by the vector algorithm. And, some practical geoscience applications will be focused.

References

1. Borgefors G. Distance transformations in arbitrary dimensions. *Computer, Vision, Graphics and Image Processing*, 1984(27):321–345

2. Larkin J. An ANSI C program to determine in expected linear time the vertices of the Convex Hull of a set of planar points. *Computers & Geosciences*, 1991, 17(3):431–443

3. Sides E, Hack R. Seminar Report: three-dimensional GIS: future developments. *ITC Journal*, 1995(2):151–164

4. Fritsch D. Three-dimensional geographical information systems status and prospects. In: *International Archives of Photogrammetry and Remote Sensing*, Vienna, 1996,(13):215–278

5. Gong J Y, Xia Z G. An integrated data model in three-dimensional GIS. In: *Proceedings of Geoinformatics ’96 Wuhan International Symposium*. 1996, 107–120

6. Manacedonio G, Pareschi M T. An algorithm for the triangulation of arbitrarily distributed points: applications to volume estimate and terrain fitting. *Computer & Geosciences*, 1991,17(7):859–874

7. Molenaar M. A topology for 3D vector maps. *ITC Journal*, 1992(1):25–33

8. Pilouk M, et al. A Tetrahedron-based 3D vector data model for geoinformation. In: *Advanced Geographic Data Modeling*. Netherlands Geodetic Commission, Publications on Geodesy, 1994, 129–140

9. Li Q Q, Li D R. Data structures in three dimensional geographic information system. *Journal of Wuhan Technical University of Surveying and Mapping*, 1996, 21(2):128–133 (in Chinese)

10. Li Q Q, Li D R. Hybrid data structure based on Octree and Tetrahedron in 3-D GIS. In: *International Archives of Photogrammetry and Remote Sensing*, Vienna, 1996,31(14):503–507

11. Li Q Q, Li D R. Study on data structure in 3-D GIS. In: *Proceedings of Geoinformatics ’96 Wuhan International Symposium*, 1997, 229–237