A QTM-based Algorithm for Generation of the Voronoi Diagram on a Sphere

Xuesheng Zhao[1,2], Jun Chen[3], Zhilin Li[1]

Department of Land Surveying and Geo-Informatics , The Hong Kong Polytechnic University, Kowloon, Hong Kong, Tel: (+852) 2766 5960, Fax: (+852) 2330 2994, lsllz@polyu.edu.hk[1]
China University of Minging and Technology (Beijing), D11 Xueyuan Road, Beijing, China, 100083, zxs@mail.cumtb.edu.cn[2]
National Geometrics Center of China, 1 Baishengcun, Zizhuyuan, Beijing, China, 10004, chenjun@nsdi.gov.cn[3]

Abstract

To efficiently store and analyse spatial data at a global scale, the digital expression of the Earth's data must be global, continuous and conjugate, i.e., a spherical dynamic data model is needed. The Voronoi data structure is the only published attempt and only solution (which is currently available) for dynamic GIS. The complexity of the Voronoi algorithms for line and area data sets in a vector-based context limits its application in dynamic GISs. As yet, there is no raster-based Voronoi algorithm for objects (including points, arcs and regions).

To overcome this deficiency, an algorithm for generating a spherical Voronoi diagram, that is a Voronoi diagram on a spherical surface, is presented based on O-QTM (Octahedral Quaternary Triangular Mesh). The basic idea is to apply the dilation operation developed in mathematical morphology to objects on the sphere in an effort to produce the effect of distance transformation. The distance contours of objects will form the Voronoi boundaries of the spherical objects.

The algorithm presented in this paper can handle point, line and area objects. Additionally, it has been tested and concluded that the processing time required for this algorithm with point, arc and region data is proportional to the levels of complexity of the spherical surface tessellation. The difference (error) between the *great circle distance* and the QTM cells distance is related to the spherical distance.

Keywords: O-QTM, neighbour triangular, Voronoi diagram on sphere, recursive dilation

1 Introduction

It is been argued by Li et al (1999) that the Voronoi data structure is the only possible solution (which is currently available) for dynamic GISs. This view is also similar to that suggested by Wright and Goodchild (1997), who point out that the Voronoi methods are the only published attempts of which we are aware that are well suited to achieving a dynamic GIS. This is because Voronoi diagrams (VD) have many excellent properties in spatial analysis (Gold 1992, Edwards 1993), dynamic operation (e.g. to add or delete objects without destroying the bubble structure of the cells) (Gold and Condal 1995; Gold and Mostafavi 2000), and computational geometry (Aurenhammer 1991; Okabe et al. 2000), etc.

So far, the Voronoi diagram on a spherical surface has been applied to some areas, such as global spatial indexing (Lukatela 1987, 2000), interpolation on a sphere (Watson 1988,1998) and dynamic operations (Gold 1997; Gold and Mostafavi 2000), *etc.* For example, Lukatela (1987) sets up a digital geo-positioning model and develops an operational software package that provides geometrical and geo-relational functions to applications that manipulate spatial objects. A Voronoi tessellation is used as a base for a highly efficient indexing system to increase the speed of data manipulation (Lukatela 1987) (Fig. 1a). Watson (1988,1998) develops the MODEMAP system and uses a point-set Voronoi diagram for interpolation on a spherical surface (Fig. 1b). Gold and Mostafavi (2000) attempt to develop a global dynamic data structure with the Voronoi diagram. In this type of structure, the Voronoi diagram is a basic data model that dynamically maintains spatial relationships.

(a) (b)

Fig. 1. The Voronoi diagram of spherical objects, **(a)** as a spatial index (Lukatela 1987), and **(b)** for interpolation (Watson 1989)

It is evident that the Voronoi diagram is a spatial data structure, which has become increasingly important. Considerable efforts have been spent on the development of the algorithms, however, most of them are applicable for vector data and are based on point sets in a planar surface. The algorithms for generating the Voronoi diagrams of line and area sets in vector data are very complex. This complexity has greatly limited the application of the Voronoi data model and only limited advances have been achieved. On the other hand, only a few algorithms are available for generating Voronoi diagrams for point-sets, but no algorithms for arc-sets (or curve-face sets) have yet been thoroughly demonstrated for a spherical surface.

This paper presents a method for computing a spherical Voronoi diagram for arbitrary input objects (including points, arcs and curve faces) based on the Quaternary Triangular Mesh (QTM). It makes use of the dilation operation through the neighbour triangles. It is simplified by using their labelling codes. Advantages of the QTM-based method are that the QTM data structure is seamless, hierarchical and numerically stable everywhere on the surface of the sphere. Its hierarchical data structure can be used to efficiently manage multi-resolution global data and it allows spatial characteristics be studied at different levels of detail in a consistent fashion across extensive regions of the sphere (Lee and Samet 2000).

The next section critically reviews a selection of algorithms. Section 3 will introduce the selection of tessellation methods on a spherical surface as a reference system. Section 4 will present in detail, a searching method of spherical neighbour triangles.. In section 5, an algorithm for generating the Voronoi diagram for objects on the sphere by means of a dilation operation in QTM is presented..

2 A Critical Examination of Algorithms for Generation of the Voronoi Diagram on a Sphere

The Voronoi diagram has been one of the research highlights in the area of computational geometry since it was introduced to the computer domain by Shamos and Hoey (see Okable et al. 2000) as an efficient data structure. Most methods of the Voronoi diagram generation are based on point sets in planar space, such as the incremental method, the divide and conquer method, the indirect generating method and the parallel method (Aurenhammer 1991; Li et al 1999; Okabe et al. 2000). There are only a few algorithms for generating spherical Voronoi diagrams (Geyer, 2000). In this section, such algorithms will be examined.

2.1 Voronoi Diagrams on a Sphere: Basic Concepts

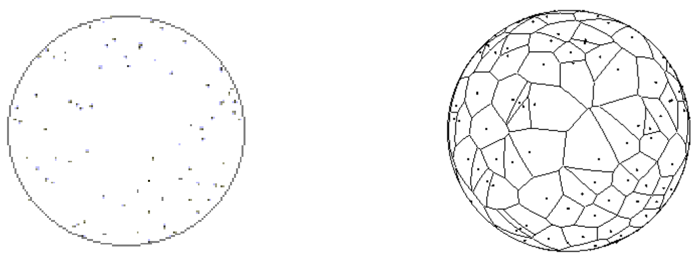
The distance on a sphere is different from that of Euclidean space. As a result, the Voronoi diagram on a sphere will have a different definition (Okabe, et al. 2000; Lee and Samet 2000): Let $P = \{ p_1, p_2, \ldots, p_n \}$ $(2 \leq n < \infty)$ be distinct points on a sphere $\boldsymbol{S}$ with the unit radius centred at the origin, and $\boldsymbol{X}$ and $\boldsymbol{X_i}$ be the location vectors of $p \in \boldsymbol{S}$ and $p_i \in \boldsymbol{S}$, respectively. The shortest distance from p to p_j on $\boldsymbol{S}$ is defined by the length of the shortest arc on the great circle (the circle whose centre is at the Center of $\boldsymbol{S}$) passing through p and p_i. Mathematically, this distance is written as

$$d_{gc}\left(p, p_i\right) = \arccos\left(X^T X_i\right) \leq \pi \qquad (2.1)$$

This distance is called the *great circle distance*. The bisector defined with the great circle distance is given by the great circle that *perpendicularly* passes through the mid-point of the great circular arc combining p_j and p (*perpendicularly* means that sufficiently small segments of the two great circles around the mid-point are orthogonal). This bisector divides the sphere $\boldsymbol{S}$ into two disjoint hemispheres. Thus the bisector defined with the great circle distance is well-behaving, and

$$V(p_i) = \left\{d_{gc}(p, p_i) \leq d_{gc}(p, p_j), j \in I_n \setminus \{i\}, p \in S\right\} \qquad (2.2)$$

gives a non-empty set in $\boldsymbol{S}$. This set is called the spherical Voronoi polygon associated with p_i. The set of resulting spherical Voronoi polygons gives a generalised Voronoi diagram, which is called the spherical Voronoi diagram defined by p_i on $\boldsymbol{S}$. Fig. 2 shows a spherical Voronoi diagram.



(a) (b)

Fig. 2. A Voronoi diagram on a sphere based on points sets (Geyer et al 2000), **(a)** discrete points on sphere, **(b)** a Voronoi diagram on spherical points sets

A *Voronoi edge* is the boundary between two Voronoi regions and a *Voronoi vertex* is the intersection of three or more Voronoi edges.

2.2 Voronoi Diagrams on a Sphere: Algorithms

Augenbaum (1985) gives an insertion method for computing the Voronoi diagram of a set of n points on a sphere with time complexity $O(n^2)$, and Robert (1997) presents an incremental algorithm, which can be constructed with time complexity $O(nlogn)$. As pointed out by (Gold 1992; Gold and Condal 1995), the vector-based methods are good for point sets, but complex for line or area input sets. This is also true as far as the spherical surface is concerned. This deficiency impedes the Voronoi data structure from being widely applied in dynamic GISs.

To solve this problem, Yang and Gold (1996) presented a *point-line* model. In their model, the complex objects are decomposed to points and lines. Voronoi diagrams for points and lines are generated first, and then translated to Voronoi diagrams of the complex objects by removing the Voronoi edges between points or lines of the same object. More recently, Gold and Mostafavi (2000) extend this model onto a spherical surface. The advantage of this method is that it can generate a Voronoi diagram of relatively complex vector-based objects and deals with dynamic changes of topological relations. However, this result can only be reached through many additional steps, such as *decompose*, *calculation*, *remove, compose*, etc. As a result, their algorithm and the data structure is relatively complex. More importantly, the data structures of these vector-based methods lack hierarchical expression. Therefore, it is very difficult to deal with the hierarchical expression of a large volume of spherical data.

Due to the complexity of vector-based methods, alternative methods, i.e. raster-based methods, on planar have been approached (Dehne 1989; Embrechts and Roose 1996; Okabe et al 2000). These algorithms make use of distance transformations, such as city block, chessboard, and Octagon, etc. It is such that the approximation of these raster distances to the Euclidean distance becomes poorer and poorer when the distances becomes larger. To solve this problem, Li et al (1999) present a method based on raster dynamic distance transformation, which employs the dilation operator which was developed in mathematical morphology. In their algorithm, the distance errors are confined to one pixel.

These raster-based algorithms, however, are limited to planar surfaces and cannot be translated onto spherical surfaces directly, simple because planar and spherical spaces are not homoeomorphic. In this paper, a spherical mesh-based method is presented for the generation of the Voronoi diagram on a spherical surface. In this method, the spherical surface is subdivided into triangles by QTM tessellation, which is similar to the planar raster. The spherical hexagon distance (to be discussed in section 5.1) is used for the dilation operation to compute the spherical Voronoi diagram of the arbitrary spatial objects (including points, arcs, and curve faces).

3 Selection of a Tessellation Method for a Spherical Reference System

To generate a Voronoi diagram on a sphere, a tessellation method for spherical surfaces and labelling schemes should be selected first. The selection criteria are based on the transformation efficiency between triangle code and its spherical coordinates and the encoding method, which best suits a search of neighbouring triangles.

3.1 Selection of a Tessellation Method for a Sphere's Surface -- O-QTM

Originally, the concept of spherical surface tessellation was presented by Fuller, a German cartographer, for studying the mapping projection in the 1940s (Dutton 1996). Since then, many researchers have approached this problem to project, analyse and index global data. Many methods are based on *inscribed polyhedron*, such as tetrahedron, cube (Snyder 1992), *octahedron* (Dutton 1996, 1999a, 1999b; Goodchild and Yang 1992, 1992; Otoo and Zhu 1993; Clarke and Mulcahy 1995), *dodecahedron* (Wickman and Elvers, 1974), *icosahedron* (Fekete 1990; White et al. 1992; Lee and Samet 2000), as shown in Fig. 3. Edges of the polyhedron are projected to the spherical surface and form the edges of spherical triangles.

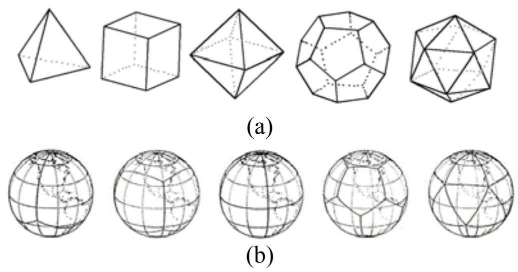


Fig. 3. Spherical surface tessellation based on inscribed polyhedrons (White et al 1992), **(a)** five polyhedrons, **(b)** projected to the spherical surface

In this study, the octahedron is chosen as a basis for an O-QTM. The reason for this selection is that it can be readily aligned with the conventional geographic grids of longitude and latitude. When this is done, its vertices occupy cardinal points and its edges assume cardinal directions, following the equator, the prime meridian, and the 90th, 180th and 270th meridians, making it simple to determine which facet a point on the sphere occupies (Dutton, 1996). In addition, each facet is a right spherical triangle and one subdivision line of each face is parallel to the equator.

There are some methods of recursive tessellation that satisfy the different requirements. It is well recognised that the transformation efficiency between triangle code and its spherical coordinates (latitude, longitude) is an important function in an application system (White and Kimerling 1998). As such the *Latitudes and longitudes average method* is selected in this study, as done by Dutton (1996). Fig. 4 illustrates levels 1, 2 and 3.

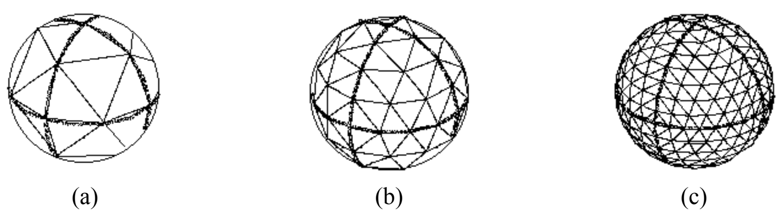
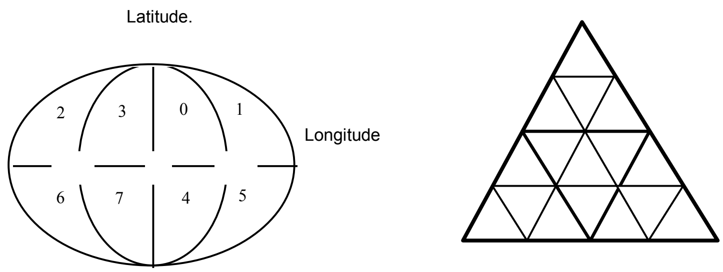


Fig. 4. Hierarchical tessellation of the spherical facet based on octahedron (Dutton 1996), **(a)** level 1, **(b)** level 2, and **(c)** level 3

3.2 Selection of an Encoding Method

The addressing methods and data structures of triangular regions are related to the operation and indexing on spherical surfaces. Some methods, such as SQT (Sphere Quadtree) (Fekete 1990), SQC (Semi-quadcode) (Otoo et al 1993), THDS (Triangular Hierarchical Data Structure) (Goodchild et al 1991) and QTM-IDs (Dutton 1996, 1999) have been applied.

The triangle labelling method selected in our study is similar to the one used for the octahedron (Goodchild et al 1991). The address code of each QTM cell consists of an octant number (from “0” to “7”) followed by up to 30 quaternary digits (from “0” to “3”), which names a leaf-node in a triangular quadtree rooted in the given octant. At the *k-th* level of decomposition, the triangle address A is represented by: $A = a_0a_1a_2a_{3\ldots}a_k$, where a_1 to a_k are k quaternary digits and a_0 is an octal digit representing the initial octahedral decomposition at level 0.



(a) (b)

Fig. 5. Surface partition and encoding, **(a)** original partition of the earth's surface, **(b)** ecnoding in 0 unit

4 A Method for Searching Neighbours in QTM

4.1 Different Types of Neighbours

The neighbours with shared edges are called *edge-neighbour-triangles.* Those only with common vertices are called *vertex-neighbour-triangles*, as shown in Fig. 6a. In the data structure of O-QTM, triangle neighbours, that are located in two adjacent octants, must also be specifically considered since the search methods for different locations of *border triangles* are different. From Fig. 7b, it can be seen that, if a triangle has edge(s) at the border of an octant, the triangle will have *edge-neighbour-triangle*(s) and *vertex-neighbour-triangles* in its neighbouring octant(s). On the other hand, if a triangle has vertices(s) at the border of an octant, the triangle will only have *vertex-neighbour-triangle*(s) in its neighbouring octant(s). Border triangles can be classified into 4 categories: *edge, sub-edge, corner* and *sub-corner triangles* and can be defined as follows (Goodchild et al 1991), see Fig. 6b:

- *Edge triangle (1)* --if it has exactly one edge-neighbour-triangle in the adjacent octant.
- *Sub-edge triangle (2)* --if it has exactly three vertex-neighbour-triangles in the adjacent octant.
- *Corner triangle (3)* --if it has exactly two edge-neighbour-triangles in the adjacent octant.
- *Sub-corner triangle (4)* --if it has exactly six vertex-neighbour-triangles in the adjacent octant.

(a) (b)

Fig. 6. Definitions of neighbour triangles and border triangles, **(a)** edge-neighbour-triangles and vertex-neighbour-triangles, **(b)** classification of a border triangle

4.2 A Method of Searching Edge-Neighbour-Triangles

All triangles have three edge-neighbour-triangles in QTM on a sphere. We use the codes *t, l, r* to represent the three edge-neighbour-triangles with common top, left and right edges for a given triangle *U* inside an octant, the code *T, L, R* to represent the edge-neighbour-triangle of a top, left and right edge triangle lying in the adjacent octant. Different searching methods used for the triangles at the different locations are in one octant. Border triangles can be classified into 7 categories and are shown in Fig. 7.

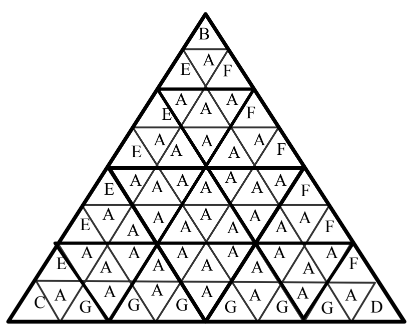
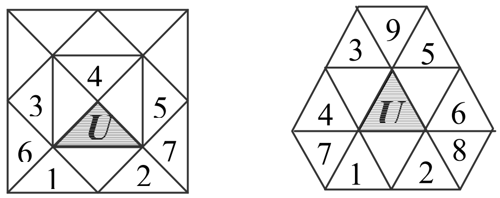


Fig. 7. Categories of triangles by different searching algorithm of edge-neighbour-triangle

The data strings *t, l, r, T, L* and *R* can be obtained by the triangular address *U.* The details can be seen in (Goodchild et al 1991).

4.3 A Method of Searching Vertex-Neighbour-Triangles

For all triangles in an octant, a corner triangle has 7 vertex-neighbour-triangles (Fig. 8-a), and each of the others have 9 vertex-neighbour-triangles, as shown in Fig. 8-b. There are several methods to search vertex-neighbour-triangles. In our approach, they have been obtained through their edge-neighbour-triangles.



(a) (b)

Fig. 8. Angle-neighbour-triangles, **(a)** corner triangle, **(b)** no-corner triangle

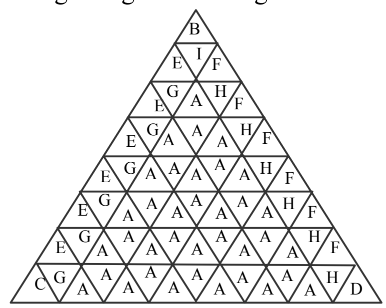


Fig. 9. Nine different types of searching methods of angle-neighbour-triangles

The searching algorithm of vertex-neighbour-triangles varies with the locations of *U* in the octant. In particular, the border triangles are much more complex. These triangles can be classified into 9 different types (shown as Fig. 9).

5 A QTM- Based Algorithm for Generating the Voronoi Diagram

The algorithm for the generation of the Voronoi diagram on a sphere is based on the dilation operator of spherical triangles. In this section, we will develop the triangle-dilation-operator in QTM according to the principle of the raster dilation operator in mathematical morphology. From this and the neighbour searching method described in section 4, the *triangle cell distance* diagram, which consists of a number of distance contours, radiated from each object on the sphere, is obtained. The most distant contours form the approximate boundaries of spherical the Voronoi diagram.

5.1 Distance Contours in QTM Generated by a Dilation Operator

Dilation and erosion are two of the basic operators in mathematical morphology and have wide applications in digital image processing as well as geographical information science (Su et al. 1997, Li et al 1999). These two basic operators in QTM cells can be defined as follows:

$$\begin{aligned} &dilation \quad A \oplus B = \cup b \in BA_b \\ &erosion \quad A \ \Theta \ B = \cap b \in BA_b \end{aligned} \tag{5.1}$$

Where A is an original region on sphere and B is a structuring element, examples of dilation and erosion are given in Fig. 10:

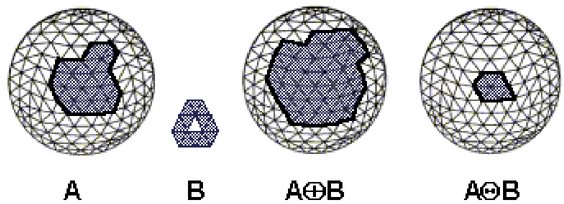
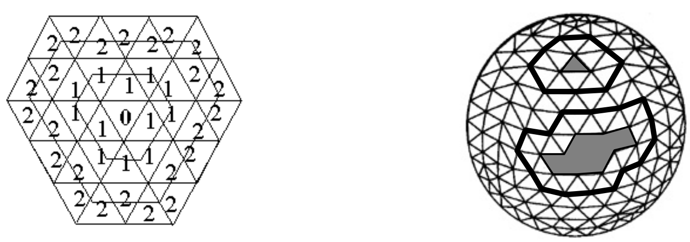


Fig. 10. Dilation and erosion operations based on QTM

Now the concept of distance becomes important. In vector mode, the *distance* on sphere means the great circle (or arc or geodesic) distance. The distance between two points $\boldsymbol{X_1}$ (λ_1, ϕ_1) and $\boldsymbol{X_2}$ (λ_2, ϕ_2) on sphere is defined as formula (5.2):

$$L_{X_1X_2} = R\theta = R\cos^{-1}(X_1 \bullet X_2) \tag{5.2}$$

Where θ is an angle between $\boldsymbol{X_1}$ and $\boldsymbol{X_2}$, and the range of $cos^{-1}\theta$ is taken to be $[0, \pi]$. In QTM, the distance in the integer number is more desirable and thus normally employed. Accordingly, the order of neighbours could be the best candidate to be used as the QTM distance to approximate the *great circle distance* (e.g. hexagon structuring element in Fig. 11a).



(a) (b)

Fig. 11. Hexagon structuring element and dilation on a sphere, **(a)** the hexagon structuring element, **(b)** the triangular dilation on a spherical surface

For an inside triangle, the region expanded is a hexagon with three edges of length $(m-1)\times l$ and three edges of length $m\times l$ if the region does not cross to another octant. Where m is the number of times the procedure is repeated and l is the edge length of the triangle at the given level. The form of the region changes if the region crosses the edge of an octant as shown in Fig. 11-b. The distance between the border of the dilated region and the nearest edge of a given triangle (point) varies from $nl\sqrt{3}/2$ *to nl*, a factor of 0.866, which is larger than in a rectangular raster where the ratio of the edge to the diagonal of a square is 0.717 (Goodchild et al. 1991). The error of forming the dilation region by hexagon structuring element in QTM is smaller than the chess structuring element in rectangular cells in planar space. In addition, the topological and metrical properties of the region are preserved:

- The region generated is connected and there exists no hole.
- The region dilated each time is a stripe region surrounding the old region with the width of $l\sqrt{3}/2$.

The neighbouring triangle search algorithm described in section 4 can be used directly for dilation since the dilation operation requires searching all neighbours (edge neighbours and vertex neighbours) of a point, arc or region.

5.2 The Principle of Generating the Voronoi Diagram on a Sphere in QTM

The Algorithm for generating the Voronoi diagram on a spherical surface is based on the principle of dilation operation in mathematical morphology. In the QTM, a point is represented by a triangle, an arc by a series of neighbour triangles and a region by a series of neighbour triangles on and within its boundary trace. The dilation operation of an arc or region can be simply done by dilation of all triangles by which the arc or region is described. Thus, the process of generating the Voronoi diagram is as follows: First, determine the *edge* and *vertex neighbour triangles* around the object (such as points, arcs and regions) by using the algorithm of searching neighbour triangles presented in section 4. Second, remove all duplicate triangles and generate the dilation trace of the object. The spherical distances are approximate equal from the outer boundary of the dilation trace to the boundary of the object. Next, repeat the dilation operation and stop when the dilation trace is intersected with the other dilation trace. The intersecting trace is just the Voronoi edge between two objects.

5.3 Algorithm

Input: tessellation level N and an object data set on a spherical surface: $\Gamma=\{A_1, A_2, A_3, \ldots, A_n\}$

Output: Voronoi diagram of input data set Γ stored in the file *VoronoiData*.

```
CsphVoronoiView::OnCaculateVoronoi( )
{
```

```
step1: LongLatitude_to_QTMcode(Γ);
step2: For every objects A_i in Γ
{
    step2.1: For every QTMcode Q_j in A_i;
        {
            Adjact12(Q_j); //searching neighbour triangles
            if Adjact12(Q_j) are copy code
                Delete_copyQTMcode(Q_j);
Else  Dialation_A[i] ← Adjact12(Q_j)
        }
step2.2: For every Dialation_A[i]
        {
            For every QTMcode Q_im in A[i] and every QTMcode Q_jk in A[j], i≠j
                {
                    if (Q_im = Q_jk)
                    VoronoiData←Q_jk;
                }
        }
}
step3: QTMcode_to_LongLatitude(VoronoiData);
step4: Output( );
}over
```

Based on the algorithm described above, a prototype system has been developed on OpenGL with VC++ language.

A number of experimental tests have been conducted but for the sake of brevity will not be discussed here. An example of the Voronoi diagram for arbitrary objects positioned on a sphere are shown in Fig. 12).

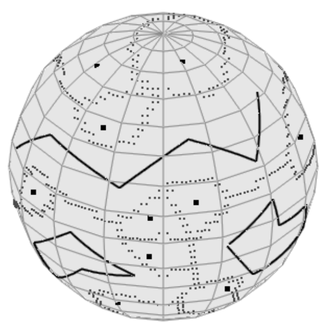


Fig. 12. Voronoi diagrams of arbitrary objects on a sphere based on QTM

6 Conclusions

In this paper, a new algorithm for generating a Voronoi diagram on the sphere is developed by the recursive dilation operation in QTM (Quaternary Triangular

Mesh). This method can easily handle arbitrary composite objects (including arcs and regions).The dilation operation, developed in mathematical morphology, was applied to objects on the sphere, in an effort to provide the effects of a distance transformation. The distance contours of objects will be used to form the boundaries of Voronoi regions of spherical objects. In this case, the principle of dilation is extended to spherical surfaces. A method for spherical distance transformation based on QTM is developed and a detailed algorithm is presented. This algorithm can handle point, line and area objects. It has also been tested and it was observed that the time consumption of this algorithm with input points, arcs and regions are equal, and is proportional to the levels of the spherical surface tessellation. The difference (error) between *great circle distance* and QTM cells distance is related to the spherical distance (better than the raster dilation in the planar space), and is related mainly to the locations of the generating points.

Acknowledgements

The work described in this paper was supported by the National Natural Science Foundation of China (under grant No.69833010) and by the Research Grants Council of the Hong Kong Special Administrative Region (Project No. PolyU 5048/98E).

References

Augenbaum M (1985) On the Construction of the Voronoi Mesh on a Sphere. Computational Physics 59: 177-192

Aurenhammer F (1991) Voronoi Diagram-A Survey of a Fundamental Geometric Data Structure. ACM Computing Survey 23(3): 345-350

Clarke KC, Mulcahy KA (1995) Distortion on the Interrupted Modified Collignon Projection. In: Proceddings of GIS/LIS 95. Nashville, TN, pp 175-181

Dehne F, Hassenklover A, Sake J (1989) Computing the configuration space for a robot on a mesh-of-processors. Parallel Computer 12(2):221—231

Dutton G (1996) Encoding and Handling Geospatial Data with Hierarchical Triangular Meshes, In: Kraak, MJ and Molenaar M (eds) Proceeding of 7th International Symposium on Spatial Data Handling. Netherlands, pp 34-43

Dutton G (1999) A hierarchical Coordinate System for Geoprocessing and Cartography. Lecture Notes in Earth Sciences, Springer-Verlag

Edwards G (1993) The Voronoi Model and Cultural Space: Applications to the Social Sciences and Humanities. In: Frank AU, Compari I (eds) Spatial Information Theory: A Theoretical Basis For GIS: European Conference, COSIT'93, Marciana Marina, Elba Island, Italy, pp 202-214

Embrechts H, Roose D (1996) A Parallel Euclidean Distance Transformation Method. Computer Vision and Image Understanding 63:15-26

Fekete G (1990) Rendering and Managing Spherical Data With Sphere Quadtree, In: Proceedings of Visualization '90. IEEE Computer Society, Los Alamitos, CA, pp 176-186

Geyer C (2000) Voronoi diagram on the surface of the sphere [online]. Available from: http://www.cis.upenn.edu/~cgeyer/sphr-vor.html

Gold CM (1992) The Meaning of Neighbour. In: Frank AU, Campari I, Formentini U (eds) Theories and Methods of Spatio-Temporal Reasoning in Geographic Space, Lecture Notes in Computing Science, 639, Springer-Verlag, pp 220-235

Gold CM (1997) The Global GIS. In: Proceeding of the International Workshop on Dynamic and Multi-Dimension GIS. Hong-Kong, China, pp 80-91

Gold CM, Condal AR (1995) A Spatial Data Structure Integrating GIS and Simulation in a Marine Environment. Marine Geodesy 18: 213-228

Gold CM, Mostafavi M (2000) Towards the Global GIS. ISPRS Journal of Photogrammetry and Remote Sensing 55(3): 150-163

Goodchild MF, Yang Shiren (1992) A Hierarchical Data Structure for Global Geographic Information Systems. Computer Vision and Geographic Image Processing 54(1): 31-44

Goodchild MF, Yang Shiren, Dutton G (1991) Spatial Data Representation and Basic Operations for a Triangular Hierarchical Data Structure. NCGIA report, 91-8

Lee M, Samet H (2000) Navigating through Triangle Meshes Implemented as Linear Quadtree. ACM transactions on Graphics 19(2): 79-121

Li C, Chen J, Li Z (1999) Raster-based Methods for the Generation of Voronoi Diagrams for Spatial Objects. International Journal of Geographic Information Science 13(3): 209-225

Lukatela H (1987) Hipparchus Geopositioning Model: An Overview. In: Proceedings of the Eighth International Symposium on Computer-Assisted Cartography. Baltimore, Maryland, 87-96

Lukatela H (2000) Ellipsoidal Area Computations of Large Terrestrial Objects [online]. Available from: http://www.ncgia.ucsb.edu/globalgrids/papers

Okabe A, Boots B, Sugihara K, Chiu S (2000) Spatial Tessellations: Concepts and Applications of Voronoi Diagrams (2nd Edition). John Wiley and Sons Ltd

Otoo E, Zhu H (1993) Indexing on Spherical Surfaces Using Semi-Quadcodes. In: Abel J, Beng CO (eds) Advances in spatial Databases 3th International Symposium, SSD`93. Singapore, pp 509-529

Robert JR (1997) Delaunay Triangulation and Voronoi Diagram on the Surface of a Sphere. ACM Transactions on Mathematical Software 23(3):416-434

Snyder JP (1992) An Equal-Area Map Projection for Polyhedral Globes. Cartographica 29(1):10-21

Su B, Li Z, Lodwick G, Muller JC (1997) Algebraic Models for the Aggregation of Area Features Based upon Morphological Operators. International Journal of Geographical Information Science 11(3): 233-246

Watson DF (1988) Natural Neighbour Sorting on the N-Dimensional Sphere. Pattern Recognition 21(1): 63-67

Watson DF (1998) Modemap: An Implementation of Natural Neighbour Interpolation on the Sphere [online]. Available from: http://members.iinet.net.au/~watson/modemap.html

White D, Kimmerling AJ (1998) Comparing Area and Shape Distortion on Polyhedral Based Recursive Tessellations of the Sphere. International Journal of Geographical Information Science 12(8): 805-827

White D, Kimmerling J, Overton WS (1992) Cartographic and Geometric Components of a Global Sampling Design For Environment Monitoring, Cartography & Geographical Information Systems 19(1): 5-22

Wickman FE, Elvers E (1974) A system of domains for global sampling problems. Geografiska Annaler 56(3/4): 201-212

Wright D, Goodchild MF (1997) Data from Deep: Impliications for the GIS Community. International Journal of Geographical Information Science 11(6):523-528

Yang W, Gold C (1996) Managing Spatial Objects With the VMO-Tree, In: Kraak, MJ, M Molenaar (eds) Proceeding of 7th International Symposium on Spatial Data Handling. Netherlands, pp 15-31