An iterative algorithm for the determination of Voronoi vertices in polygonal and non-polygonal domains on the plane and the sphere

Francois Anton*, Jack Snoeyink**, and Christopher Gold*
*Industrial Chair of Geomatics - CRG 0722 Casault, Universite Laval
Sainte-Foy, Quebec, Canada, G1K 7P4 Fax: (+1-418) 656-7411
E-mail: francois@gmt.ulaval.ca & Christopher.Gold@scg.ulaval.ca
**Department of Computer Science University of British Columbia 201-2366
Main Mall Vancouver, B.C., Canada, V6T 1Z4 Fax: (+1-604) 822-5485 E-mail: snoeyink@cs.ubc.ca

Abstract

1. Introduction

The Voronoi diagram has many applications in a variety of disciplines, and has been widely treated in the literature (see Okabe and *al.* [4] and Aurenhammer [1] for a general survey). The Voronoi diagram has been named after the rusian mathematician Georgii Fedeorovitch Voronoi, who wrote a treatise on quadratic forms theory (see Voronoi [6], [7], [8]). The ordinary point Voronoi diagram is a partition of the plane, in the way that each object (point) partitions the euclidean plane into a region, that is the locus of points which are closer from that object than from any other object (see Preparata and Shamos [5]).

The Voronoi diagram for a set of geometric objects (points, curves, surfaces) is defined by the generalization of the ordinary point Voronoi diagram by extending the set of objects S to any geometric element. This partition of the plane (or the sphere) forms a net, whose vertices are called Voronoi vertices, and whose edges are called Voronoi edges. Each Voronoi vertex is the common intersection of exactly three edges, and therefore each Voronoi vertex is equidistant from its three nearest objects. Tradi-tionnal computational geometry methods needed to know the shape of the geometric objects, and they were limited by the complexity of the curves or surfaces. An iterative algorithm has been proposed for computing the vertices of a Voronoi diagram for geometric objects, on the euclidean plane (see Ferruci and *al.* [2]), without having to know the exact shape of the geometric objects. In their algorithm, the only assumption is "to be able to answer to queries of the form "given a point p and an object S, determine the closest point on S from p"" (Ferruci and *al.* [2]). Starting from a point p on the plane, they compute the closest point on each object. Then, they compute the circumcentre of these three points, that will be the point p for the next iteration. They have denned a sufficient condition of convergence, based on the fact that the smallest circle containing three points and whose centre is inside the triangle formed by these three points is the circle circumscribed to the three points. The sufficient condition is that the next point p is inside the triangle formed by the closest point of each one of the three objects from the previous point p. This sufficient condition is rather restrictive, insufficient to induce a general algorithm (see [3] and figure 1), and it is not a necessary and sufficient condition of convergence. In this paper, we give a necessary and sufficient condition of convergence for a similar iterative algorithm, that works both on the plane and on the sphere.

2. Preliminaries

Let O_1, O_2, and O_3 be three objects.

The iterative algorithm (see figure 2) starts with three

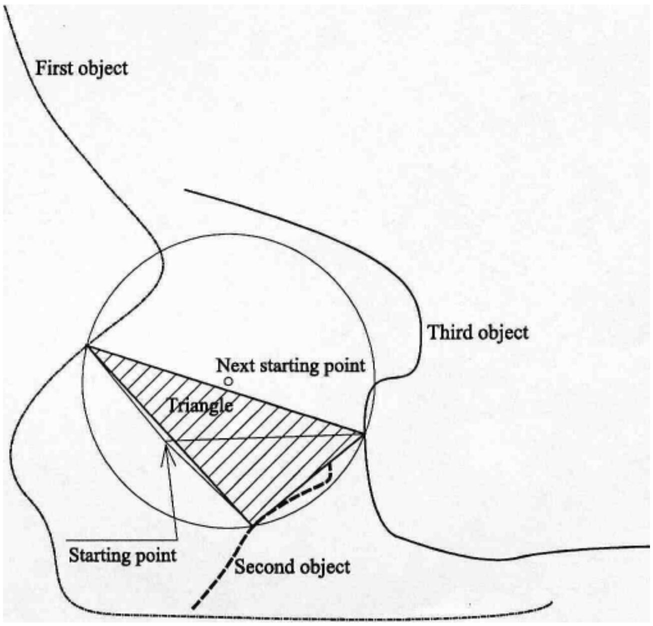

Figure 1: A case where Ferruci and *al.'s* sufficient condition is not respected

arbitrary points (called feet) taken on each one of the three objects: F_{1_0}, F_{2_0} and F_{3_0} The centre *C*o of the circle *C*o circumscribed to the triangle (planar or spherical) formed by these three feet is computed. Then, each one of the closest point of O$_1$, O$_2$, and O$_3$ from *Co:* F_{1_1}, F_{2_1} and F_{3_1} is computed and used as the starting point (foot) for the next iteration. The iterations stop when the distance between the present centre and the last one is smaller than a user-defined tolerance. The searched Voronoi vertex is the centre of the circle that "touches" the three objects in the expected order (first O$_1$, then O$_2$, and finally O$_3$).

Let $(F_{1_n})_{n\in\mathbb{N}}$, $(F_{2_n})_{n\in\mathbb{N}}$, and $(F_{3_n})_{n\in\mathbb{N}}$ be the sequences of the points (called feet) on each one of the three objects O$_1$, O$_2$, and O$_3$, closest to the centre of C$_{n-1}$ except for *n* = 0 where the foot are arbitrary points on each one of the objects. Let C_n be the circle passing through F_{1_n}, F_{2_n} and F_{3_n} for n $\geq$ 0.

3. A necessary and sufficient condition of convergence

We will consider for each object O_i, the portion O_{i_n} of O_i inside the disk (or the sphere) D_n, whose boundary is C_n (see figure 3).

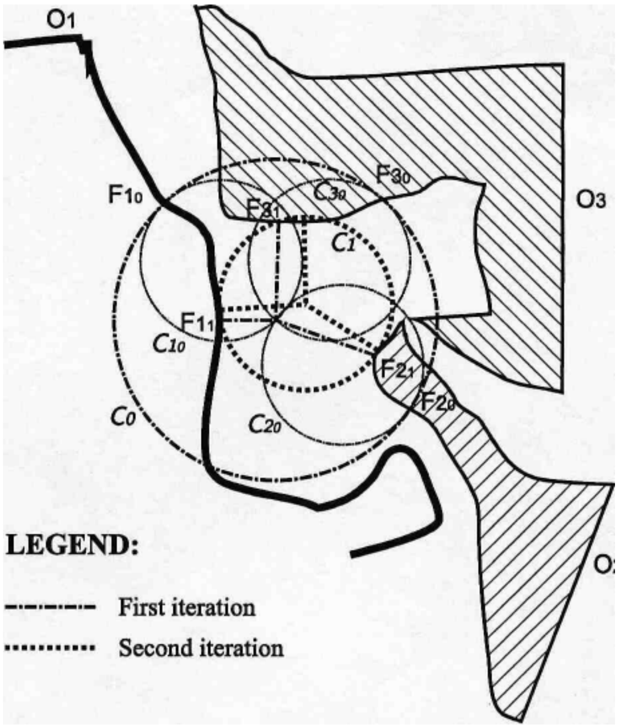

Figure 2: The iterative algorithm

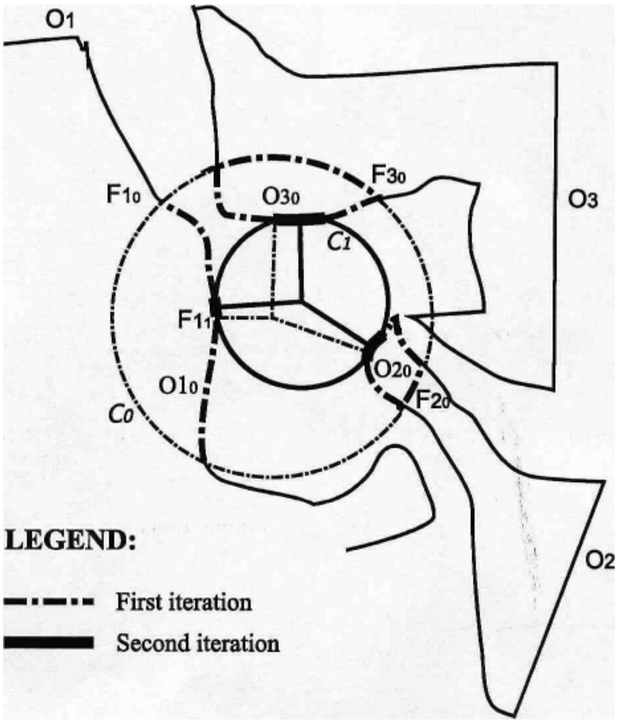

Figure 3: O_{i_n} : the portion of O_n n = 1,2,3

We proved the necessary and sufficient condition:

Theorem 1 The necessary and sufficient condition of convergence of this iterative algorithm is the following one:

$$\exists p \in \aleph / \forall n \geq p : \overrightarrow{F_{1_n}F_{2_n}} \cdot \overrightarrow{F_{1_n}F_{3_n}} \geq 0 \qquad (1)$$

From this necessary and sufficient condition of convergence of this iterative algorithm, we get directly the initial conditions for this algorithm:

Lemma 2 If we start from three feet in the expected final order; and $\forall(P,Q,R) \in O_{1_0} \times O_{2_0} \times O_{3_0} : \overrightarrow{PQ} \times \overrightarrow{PR} \times \vec{k} > 0$ If, and only if, there exists a circle touching the three objects in the specified order, and for which there is no intersection with another object between the closest point of each object O_i from its centre and F_{i_0}, the sequences $(F_{i_n})_{n \in \aleph}$ will converge towards the closest points of each one of the objects from the centre of that circle, and the sequence of centres of the circles C_n will converge towards the centre of that circle.

4. The Algorithm: description and statistical validity

An algorithm for the determination of Voronoi vertices for points and line segments has been developped using the precedent lemma. It is subdivided into three steps: two steps are necessary to satisfy the initial conditions, and the last step is the iterative algorithm itself.

In the first step, three starting feet in the expected order (see Preliminaries) are chosen in order to satisfy to $\overrightarrow{F_{1_0}F_{2_0}} \cdot \overrightarrow{F_{1_0}F_{3_0}} \geq 0$. Before trying to choose such feet, the extremities of the objects are checked to assess if it is possible. Then, for each line segment object, a feet is randomly chosen till it is in the good order relatively to the other objects.

In the second step, the choice of the, starting feet is corrected in order to satisfy the initial condition:

$$\forall(P,Q,R) \in O_{1_0} \times O_{2_0} \times O_{3_0} : \overrightarrow{PQ} \times \overrightarrow{PR} \times \vec{k} > 0.$$

To respect this rule, each previously chosen foot at the iteration n is replaced by a feet for which the two extremities of $O_{i_{n+1}}$ are in the good order relatively to the other objects.

The test data was composed of 10000 triples of objects. Among them, 3280 were assessed positively for circumcentre possibility. There was a valid circumcentre for 2997 cases, that is 91.37% of the previous set.

For the first step, we can conclude that 5 (4,82) iterations are needed in average. With a confidence interval of 95%, 6 (5,73) iterations are necessary. For the second step, we can conclude that 2 (1,44) iterations are needed in average. With a confidence interval of 95%, 2 (1,64) iterations are necessary. Finally, for the third step, we can conclude that 7 (6,16) iterations are needed in average. With a confidence interval of 95%, 7 (6,44) iterations are necessary.

5. Conclusions

This algorithm can be easily extended to the n-dimensional Euclidean space, and to spheres of higher dimensions. Moreover, it has many applications in GIS, navigation and robotics. In GIS, the maps that are handled are composed of geometric objects whose shape can be very complex. This method is better suited than traditional Voronoi based methods, because of the fact that we don't need to know the exact shape of the objects, and we are not therefore dependant on the complexity of the geometric objects. In navigation and robotics, the potential applications of this algorithm reside in real-time motion planning and retraction motion planning (see [3]).

References

[1] F. Aurenhammer, Voronoi diagrams - A survey, (Institute for Information Processing, Technical University of Graz, Report 263)

[2] V. Ferruci, M. Overmars, A. Rao, and J. Vieugels, Hunting Voronoi Vertices in Non-Polygonal Domains, (CCCG'94, Saskatoon, Canada, 1994) 45-50

[3] V. Ferruci, M. Overmars, A. Rao, and J. Vieugels, Hunting Voronoi Vertices, Technical Report, University of Rome, Italy, 27 p.

[4] A. Okabe, B. Boots, and K. Sugihara, Spatial Tessellations, Concept and Applications of Voronoi Diagrams, (John Wiley & Sons, 1992) 532 p.

[5] F.P. Preparata, and M.I. Shamos, Computational Geometry, An Introduction, (Springer-Verlag, 1985) 398 p.

[6] G. Voronoi, Nouvelles applications des parametres con-tinus a la theorie des formes quadratiques, Premier Memoire, Sur quelques proprietes des formes quadratiques positives parfaites, (Journal fur die Reine und Angewandte Mathematik, v. 133, 1908) 97-178

[7] G. Voronoi, Nouvelles applications des parametres con-tinus a la theorie des formes quadratiques, Deuxieme Memoire, Recherches sur les parallelloedres primitifs, (Journal fur die Reine und Angewandte Mathematik, v. 134, 1909) 198-287

[8] G. Voronoi, Nouvelles applications des parametres con-tinus a la theorie des formes quadratiques, Deuxieme Memoire, Recherches sur les parallelloedres primitifs, (Journal fur die Reine und Angewandte Mathematik, v. 136, 1909) 67-181