CONSTRUCTION OF EUCLIDEAN VORONOI DIAGRAM FOR 3D BALLS BY TRACING EDGES

Youngsong CHO [1] , Donguk KIM [1] and Deok-Soo KIM [2]

1) *Voronoi Diagram Research Center, Hanyang University, Seoul 133-791, KOREA*

2) *Department of Industral Engineering, Hanyang University, Seoul 133-791, KOREA*

Corresponding Author : Deok-Soo KIM, dskim@hanyang.ac.kr

ABSTRACT

Despite its important applications in various disciplines in science and engineering, the Euclidean Voronoi diagram for balls in 3D space has not been studied as much as it deserves. In this paper, we present an algorithm to compute the Euclidean Voronoi diagram for 3D balls with different radii. The presented algorithm follows Voronoi edges one by one until the construction is completed in O(mn) time in the worst-case, where m is the number of edges in the Voronoi diagram and n is the number of spherical balls.

EUCLIDEAN VORONOI DIAGRAM FOR BALLS

Voronoi diagram has been known for its capabilities to handle various applications in science and engineering. The ordinary Voronoi diagram for points has been studied extensively and its properties are well-known in 2 and higher dimensions [1]. However, the Voronoi diagram for balls, often called an *additively weighted Voronoi diagram* in the computational geometry community, has not been explored sufficiently even though it has significant potential impacts on diverse applications in both science and engineering [2–5].

Let $B = \{b_1, b_2, \ldots, b_n\}$ be a set of generators for a Voronoi diagram where b_i is a 3-dimensional spherical ball. Hence, $b_i = (c_i, r_i)$ where $c_i = (x_i, y_i, z_i)$ and r_i denote the center and the radius of a ball b_i, respectively. We assume that no ball is completely contained inside another ball even though intersections are allowed between balls. Associated with each ball b_i, there is a corresponding *Voronoi region* VR$_i$ for b_i, where VR$_i$ $= \{p \mid dist(p, c_i) - r_i \leq dist(p, c_j) - r_j, i \neq j\}$. Then, EVD($B$)= {VR$_1$, VR$_2$,. . ., VR$_n$} is called a *Euclidean Voronoi diagram* for B. In this paper, the ordinary L_2-metric is used to define Euclidean Voronoi diagram. In other words, $dist(p, c_i) = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2}$.

As in ordinary point set Voronoi diagrams, Voronoi regions corresponding to balls on the boundary of the convex hull of B are unbounded. Other regions are bounded by a set of faces, called *Voronoi faces* which is defined by two immediately neighboring balls. Note that a Voronoi face is always a subset of hyperboloid of two sheets. A Voronoi face intersects another Voronoi face to form a *Voronoi edge* which is a conic section. When Voronoi edges intersect, a *Voronoi vertex* is defined. Note that there is always one-to-one correspondence between a Voronoi vertex and a sphere tangent to neighboring balls defining the vertex. Hence a tangent sphere is always centered at the corresponding vertex. In that case, the tangent sphere is called *empty* since it never intersects or contains any other ball except at tangent points with its generating balls. We assume that the generators are in general position.

EDGE-TRACING ALGORITHM

The basic idea of the edge-tracing algorithm is quite simple yet powerful. Based on this idea, we recently reported a full implementation of the edge-tracing algorithm with discussions on various applications [6]. Our edge-tracing algorithm is as follows. The algorithm first locates a true Voronoi vertex by computing an empty tangent sphere defined by four appropriate nearby balls. Then, four edges emanating from the initial vertex is identified and pushed into a stack called *Edge-stack*. Note that those edges have the initial vertex as their starting vertices. After popping an edge from the stack, the algorithm computes the end vertex of the popped edge. Note that a vertex can be found by computing a tangent sphere from each of n-3 balls plus three balls defining the popped edge and testing if the corresponding tangent sphere is empty. Once the end vertex of currently popped edge is found, it is also possible to define three more edges emanating from this new vertex. Hence, three edges are created and the new vertex is used as the starting vertex of three new-born edges. Note that these edges are also pushed into Edge-stack. By following this process until Edge-stack is empty, the computation of Voronoi diagram is completed. Even though the idea is simple, designing a correct and efficient algorithm is not so easy at all.

The worst-case time complexity for the whole Voronoi diagram takes O(mn) where n is the number of balls and m is the number of edges in the diagram. Note that m can be O(n^2) in the worst-case. However, the running time can be reduced as low as O(n) by employing various acceleration techniques.

ACKNOWLEDGEMENTS

This research was supported by the Creative Research Initiatives from the Ministry of Science and Technology in Korea.

REFERENCES

1. Okabe, A., Boots, B., Sugihara, K. and Chiu, S.N., *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams, 2nd ed.*, John Wiley & Sons, 1999.
2. Montoro, J.C.G. and Abascal, J.L.F., "The Voronoi polyhedra as tools for structure determination in simple disordered systems", *The Journal of Physical Chemistry*, Vol. 97, No. 16, 1993, pp. 4211-4215.
3. Goede, A., Preissner, R. and Frömmel, C., "Voronoi cell: new method for allocation of space among atoms: elimination of avoidable errors in calculation of atomic volume and density", *Journal of Computational Chemistry*, Vol. 18, No. 9, 1997, pp. 1113-1123.
4. Angelov, B., Sadoc, J.-F., Jullien, R., Soyer, A., Mornon, J.-P. and Chomilier, J., "Nonatomic solvent-driven Voronoi tessellation of proteins: an open tool to analyze protein folds", *Proteins: Structure, Function, and Genetics*, Vol. 49, No. 4, 2002, pp. 446-456.
5. Voloshin, V.P., Beaufils, S. and Medvedev, N.N., "Void space analysis of the structure of liquids", *Journal of Molecular Liquids*, Vol. 96-97, 2002, pp. 101-112.
6. Kim, D.-S., Cho Y. and Kim D., "Euclidean Voronoi diagram of 3D balls and its computation via tracing edges", *Computer-Aided Design*, Vol. 37, No. 13, 2005, pp. 1412-1424.