For Fourth Canadian Conference on Computational Geometry

Numerically Robust Incremental Algorithm for Constructing Three-Dimensional Voronoi Diagrams

Hiroshi Inagaki[1] Kokichi Sugihara[2] Noboru Sugie[3]

[1]Toyota College of Technology, JAPAN
e-mail inagaki@tctice.toyota-ct.ac.jp

[2]Faculty of Engineering, University of Tokyo, JAPAN
e-mail sugihara@simplex.t.u-tokyo.ac.jp

[3]Faculty of Engineering, Nagoya University, JAPAN
e-mail sugie@sugie.nuee.nagoya-u.ac.jp

Abstract

This paper presents a numerically robust incremental algorithm for constructing three-dimensional Voronoi diagram. In the algorithm the higher priority is placed on the consistency of the topological structure than on numerical values, so that, no matter how large numerical errors may take place, the algorithm will never come across topological inconsistency and thus can always complete its task. The output is in general an approximation of the Voronoi diagram, but it converges to the true diagram as the precision in computation becomes higher.

The algorithm was implemented into a computer program. Our program can construct three-dimensional Voronoi diagram even in highly degenerate cases which are numerically unstable for conventional algorithms.

1 Introduction

Many efficient geometrical algorithms have been proposed. However, these algorithms were designed on the assumption that numerical errors do not take place, and hence computer programs based on these algorithms often fail because of inconsistency due to computational errors [1].

Here, a numerically robust algorithm for constructing three-dimensional Voronoi diagram is proposed. In this algorithm higher priority is placed on the topological structure than on numerical values, so that, no matter how large numerical errors may take place, the algorithm will never come across topological inconsistency and thus can always complete its task. For Voronoi diagram in the plane, some algorithms on the same basic idea have been proposed [2, 3]. However, in the case of three-dimensional Voronoi diagram, some new ideas are necessary because the topological structure is much more complicated.

Our experimental computer program based on this algorithm can construct three-dimensional Voronoi diagrams stably even in numerically ill-conditioned cases.

2 Three-Dimensional Voronoi Diagram and the Incremental Construction

For two points p and q, let $d(p,q)$ denote the Euclidean distance between p and q. For a finite set $P = \{p_1, p_2, \cdots, p_n\}$ of points in the three-dimensional space, region $V(p_i)$ is defined by

$$V(p_i) = \{p \mid d(p,p_i) < d(p,p_j) \ for\ any\ j(\neq i)\}$$

and called Voronoi region of p_i. A Voronoi region is a polyhedron, and its vertices, edges, and faces are called Voronoi points, Voronoi edges, Voronoi faces, respectively. Voronoi regions $V(p_1), V(p_2), \cdots, V(p_n)$ make a partition of the space, and this partition is called the Voronoi diagram for P. An element of P is called a generator of the Voronoi diagram.

Among many algorithms for the construction of two-dimensional Voronoi diagrams, a rather sophisticated implementation of the incremental-type algorithm is most practical [4]. It runs, on the average, in $O(n)$ time for n generators. The incremental-type algorithm starts with a simple Voronoi diagram for several generators, and modifies it step by step by adding new generators one by one.

However, the incremental-type algorithm as well as other algorithms is unstable because of computational errors when degeneracy takes place or when the situation is very close to degeneracy. Here we re-design the incremental-type algorithm so as to make it work well in finite-precision environment.

3 Design of a Robust Algorithm

We assume that errors take place in numerical computation, so that no judgement based on numerical computation is absolutely reliable. In this circumstance we can rely only on combinatorial computations, and hence let us concentrate on combinatorial and/or topological properties. The three-dimensional Voronoi diagram has the following topological properties.

P1 A Voronoi diagram partitions the space into as many regions as the generators.

P2 A Voronoi region is simply connected.

P3 Two Voronoi regions do not share two or more faces as a common part of their boundaries.

Since numerical errors are inevitable, we cannot construct the Voronoi diagram correctly. What we can do is to try to construct a three-dimensional diagram which shares the topological properties P1 $\sim$ P3. For this purpose the basic structure of the algorithm is designed only in terms of combinatorial computation, and numerical results are used as lower-priority information for selecting a more probable structure of the diagram from among all the possible. The algorithm thus designed does not come across any topological inconsistency. It always carries out its task and gives some output.

Let V_{l-1} denote the Voronoi diagram for $p_1, p_2, \cdots, p_{l-1}$. From a topological point of view, we redesign the procedure for changing V_{l-1} to V_l on addition of a new generator p_l. This task is done by the next procedure.

Procedure A

1. Select a subset, say T, of the vertex set of V_{l-1} (see Figure 1(a); the solid circle in this figure represents a Voronoi point belonging to T).

2. For every edge connecting a vertex in T with a vertex not in T, generate a new vertex on it (see (b); hollow circles represent newly generated vertices).

3. For every region on which new vertices are generated in Step 2, connect new vertices by new edges to form a cycle on the boundary of the region, and generate a new face bounded by this cycle (see (c)).

4. Remove the vertices in T, and the edges and faces incident to them (see (d)).

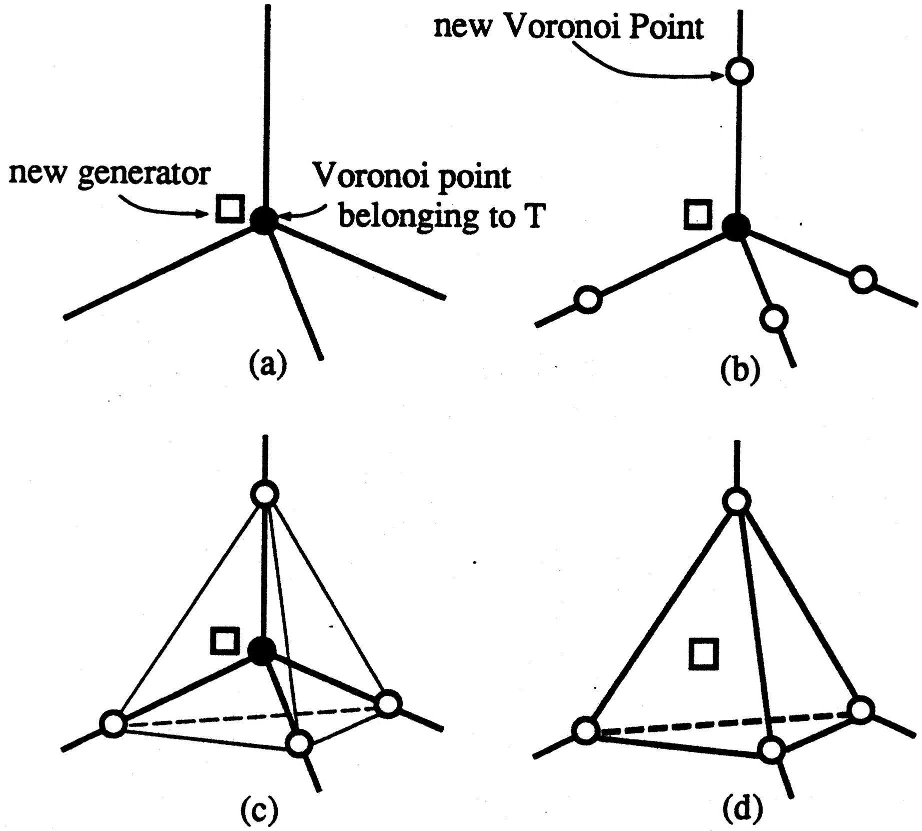


Figure 1: Change of the topological structure on addition of a new generator

Note that Procedure A is described in purely combinatorial terms, so that this procedure is not affected by numerical errors. However, there is ambiguity in the choice of T in Step 1. Next, we consider what conditions should be satisfied by T in order for Procedure A to be the correct procedure for constructing the Voronoi diagram. T should satisfy at least the following conditions, where $\overline{T}$ represents the set of Voronoi points that do not belong to T.

C1 T is nonempty.

C2 The subgraph of V_{l-1} that consists of the vertices in T and the edges connecting two vertices in T is connected.

C3 At least one vertex on each Voronoi region belongs to $\overline{T}$.

C4 Let $T(p_i)$ denote the subset of T that are on Voronoi region $V(p_i)$. For any $i = 1, 2, \cdots, l-1$, the subgraph of V_{l-1} that consists of the vertices in $T(p_i)$ and the edges connecting two vertices in $T(p_i)$ is connected.

C5 Let $\overline{T}(p_i)$ denote the subset of $\overline{T}$ that are on Voronoi region $V(p_i)$. For any $i = 1, 2, \cdots, l-1$, the subgraph of V_{l-1} that consists of the vertices in $\overline{T}(p_i)$ and the edges connecting two vertices in $\overline{T}(p_i)$ is connected.

In Step 1 of Procedure A, the subset T should be chosen in such a way that the conditions C1 $\sim$ C5 are satisfied. Note that all of C1 $\sim$ C5 are combinatorial conditions, and hence we can check them without worrying about numerical errors.

However, the subset T satisfying C1 $\sim$ C5 is not unique. To resolve this ambiguity we need to employ numerical computation. Let q be the Voronoi point shared by the boundaries of four Voronoi regions $V(p_\alpha), V(p_\beta), V(p_\gamma), V(p_\delta)$, and let S be the sphere passing through $p_\alpha, p_\beta, p_\gamma, p_\delta$.

Let us call S the sphere associated with q. From the definition of the Voronoi diagram, q is the center of S, and hence q should be removed on the addition of the new generator p_l (i.e., q should belong to T) if and only if p_l is inside S. Keeping this in mind, we create the vertex set T in the following way.

Procedure B

1. Among the old generators $p_1, p_2, \cdots, p_{l-1}$, find the one, say p_α, that is nearest to the new generator p_l.

2. From the vertices on the boundary of $V(p_\alpha)$, select the one whose associated sphere is most likely (according to numerical computation) to include p_l, and initialize T as the set consisting of this vertex.

3. Augment T by adding the vertices whose associated spheres contain p_l and whose addition to T does not violate C2 $\sim$ C5, until T cannot be augmented any more.

Thus, our algorithm is basically Procedure A, in which Procedure B is called in order to construct the set T of vertices to be deleted.

4 Computational Experiments

The proposed algorithm was implemented into a computer program, and many computational experiments were done. Sun4/370 with UNIX operating system was used, and all the floating-point computations were carried out in single precision.

Figure 2(a) shows the output of the program for fifty generators placed at random in the unit cube $\{(x, y, z) \mid 0 \leq x, y, z \leq 1\}$. This output can be considered as a correct Voronoi diagram in the sense that we visually find no difference between the output and the correct Voronoi diagram. (b) shows one Voronoi region near the central portion of the unit cube. The output is represented in such a way that the three-demensional structure can be perceived when the output on the right side is seen by the left eye, and the left side by the right eye.

Figure 3 shows the time required for the computation for various numbers of generators up to 1000. This graph shows the program runs in $O(n)$ average time for n generators. Therefore, topological check in the algorithm does not increase the average time complexity.

Figure 4(a) shows the output of the program for one hundred generators placed at random on the sphere inscribed in the unit cube $\{(x, y, z) \mid 0 \leq x, y, z \leq 1\}$. (b) represents the central portion of the same output magnified by 10^6. (a) seems to be a correct Voronoi diagram, but, as (b) shows, the central portion of the diagram is far from the correct Voronoi diagram. This set of generators gives a highly degenerate case, and the existence of such crisscross structures at the central portion seems natural as the output obtained in single precision. It should be noted that even for such a degenerate set of generators the program carried out its task and gave the output.

5 Concluding Remarks

We have proposed a numerically robust incremental algorithm for constructing the three-dimensional Voronoi diagram. In the algorithm the higher priority is placed on the consistency of the topological structure than on numerical values, and hence it always gives a topologically consistent output. Furthermore, the output "converges" to the true Voronoi diagram as the precision in computation becomes higher.

The algorithm was implemented into a computer program, and numerical robustness of the algorithm has been proved by computer experiments.

This work is partly supported by the Hori Information Science Promotion Foundation.

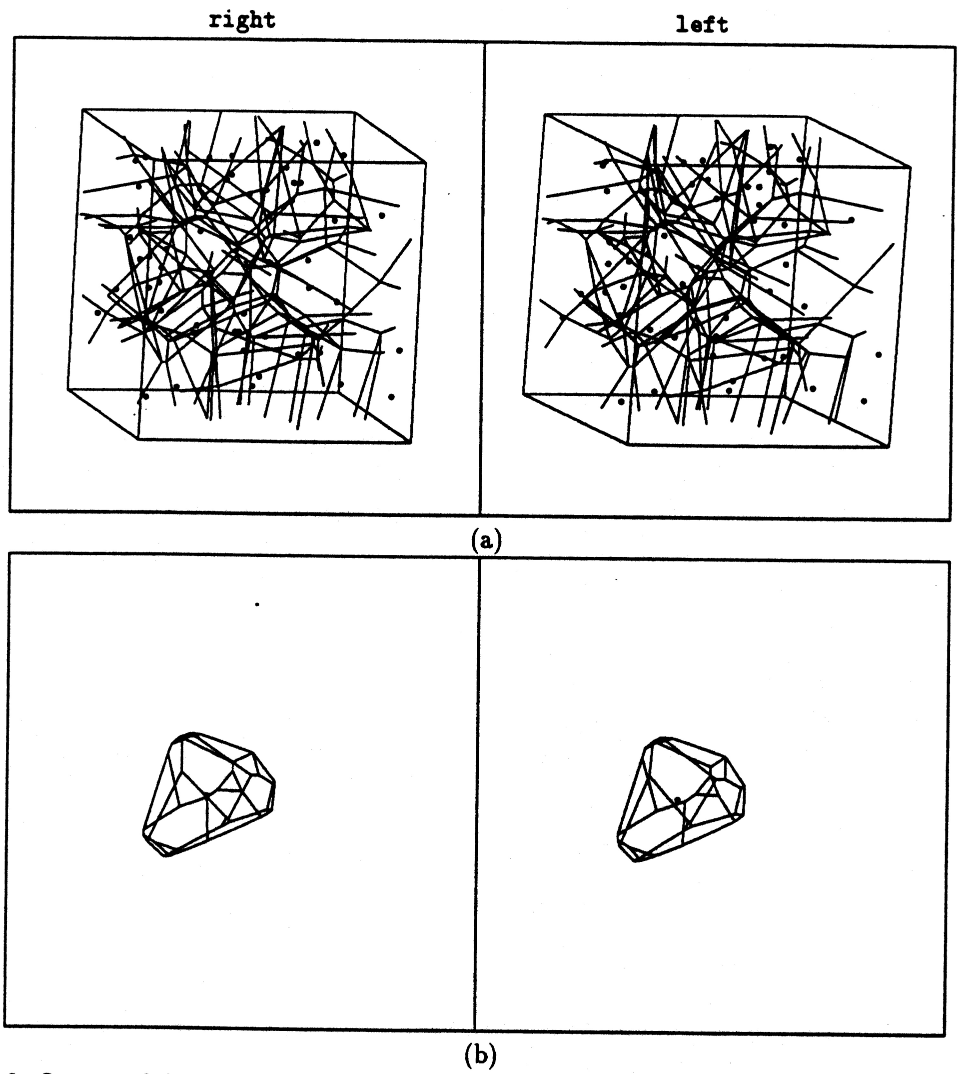


Figure 2: Output of the program for fifty generators placed at random in the unit cube: (a) output; (b) an example of a Voronoi region near the central portion.

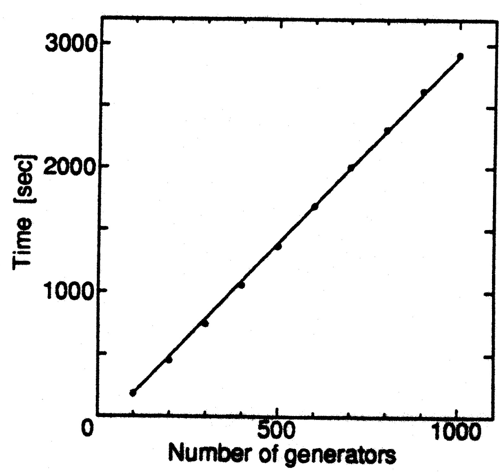
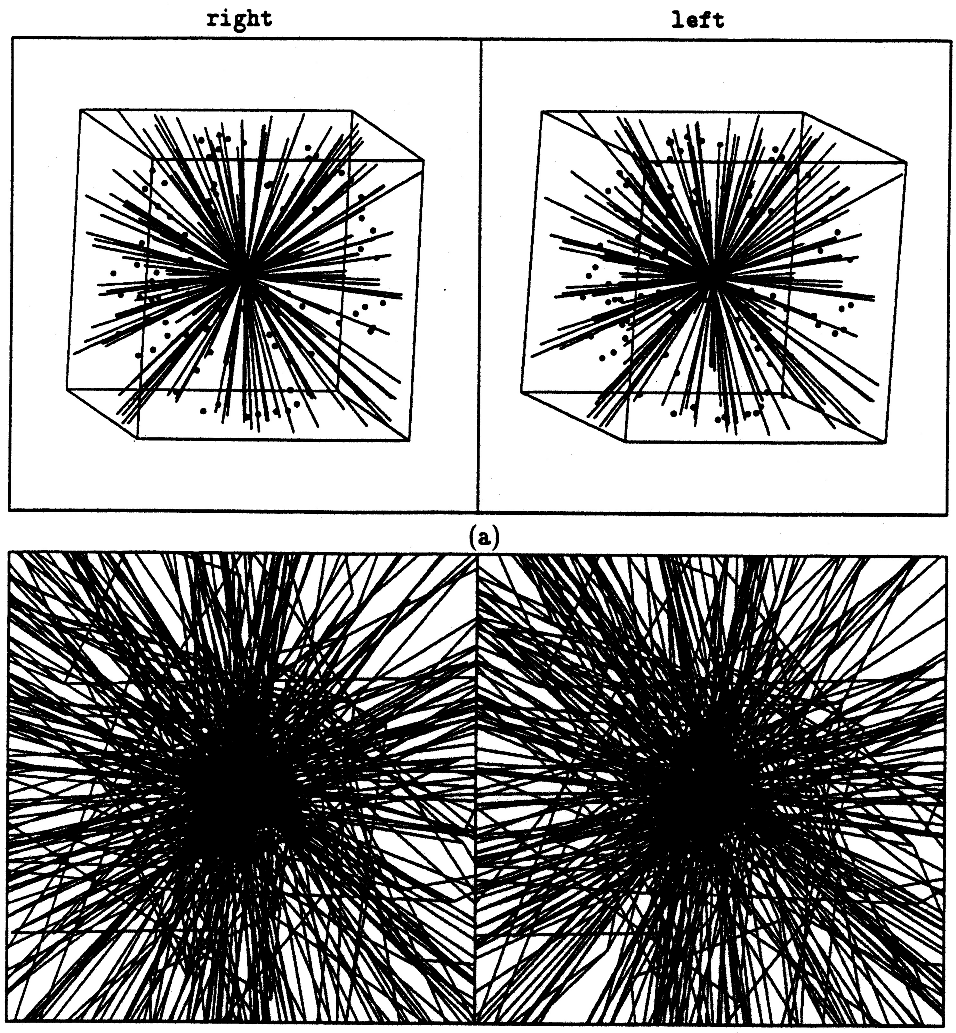


Figure 3: Average time complexity



(a)

(b)

Figure 4: Output of the program for one hundred generators placed at random on the sphere inscribed in the unit cube: (a)output; (b)central portion magnified by 10^6)

References

[1] C. M. Hoffman: The problems of accuracy and robustness in geometric computation. IEEE Computer, vol. 22, no. 3 (March 1989), pp. 31–41.

[2] K. Sugihara and M. Iri: Two design principles of geometric algorithms in finite-precision arithmetic. Applied Mathematics Letters, vol. 2 (1989), pp. 203–206.

[3] K. Sugihara and M. Iri: Construction of the Voronoi diagram for "one million " generators in single-precision arithmetic. Proceedings of the IEEE (to appear).

[4] T. Ohya, M. Iri and K. Murota: Improvemens of the incremental method for the Voronoi diagram with computational comparison of various algorithms. Journal of the Operations Research Society of Japan, vol. 27 (1984), pp. 306–336.