アルゴリズム　1－2
（1988. 5. 23）

移動する点集合のVoronoi図の変化

徳山　豪
日本アイ・ビー・エム㈱　東京基礎研究所

母点が動く時のボロノイ図の変化を調べる問題は、平面の勢力図の変形のシュミレーションに関係して、興味深い研究対象である。特に、母点が固定した点集合Ｐと移動する点集合Ｑ(t)に分かれている場合に、ボロノイ図の位相変化を考える。　時間ｔでのボロノイ図Ｖ(t)が位相変化する瞬間の状態のリストを、ボロノイ図の歴史という。歴史を用意することにより任意の時間での位相を知ることができる。この講演では、Ｑ(t)が平行移動する時にボロノイ図Ｖ(t)の歴史を求めるＯ(n^2log n)のアルゴリズムを与える。各時点での位相をＯ(n)で取り出すために、Ｏ(n^2)の領域を用いて歴史を蓄える。

DEFORMATION OF MERGED VORONOI DIAGRAMS WITH TRANSLATION

Takeshi TOKUYAMA
IBM Research, Tokyo Research Laboratory
5-19 Sanban-cho, Chiyoda-ku, Tokyo 102, Japan

(英称住所)

A Voronoi diagram of moving data points is an interesting target to apply to simulated deformation of planar subdivision. As a special case, we consider the deformation of the topological structure of a Voronoi diagram of points moving continuously in two groups.

Let P be a set of static points and Q(t) be a set of moving points with time parameter t, where $0 \leq t \leq T$. The Voronoi diagram V(t) of the point set $P \bigcup Q(t)$ is deformed as time passes. The history of V(t) means the list of all 'catastrophic epochs'.

In this paper, we give an algorithm for processing the history of V(t) when Q(t) moves by simultaneous translation. Our algorithm needs $O(n^2 \log n)$ time and $O(n^2)$ space to store the history in order to retrieve the $V(t)$ of any epoch in linear time.

Deformation of Merged Voronoi Diagram with Translation.

Introduction

The treatment of dynamic data is an important theme in computational geometry. Various kinds of deformation of data must be considered separately dependent on the applications. For example, the following three kind of deformation must be treated with different methods :

1. Discrete update (insertion and deletion).
2. Continuous movement.
3. Others (e.g. probabilistic movement).

Discrete update has been well studied, and many beautiful algorithms and data structures have been created. On the other hand, continuous movement of data seems to have been less studied, although it has important applications to graphics, robotics, and other areas involving moving objects. In this article, we shall be concerned with the deformation of the geometric structure of Voronoi diagrams caused by continuous movement of data. Since Voronoi diagrams are used to simulate many planar objects, a mobile Voronoi diagram is an interesting target to apply to simulated deformation of planar objects.

Let us begin with more general problems. Let $S(t)$ be a data set moving continuously on the 'time' parameter t. Let us use $A(t)$ to denote for the output we get from $S(t)$ at t. The following two problems occur naturally.

Problem 1 -- Fast construction.
Find an algorithm with suitable preprocessing on $S(t)$ such that it output $A(t)$ for any given t faster than static algorithm.

Problem 2 -- Investigation of history.
when the output $A(t)$ takes discrete values with respect to t , report all catastrophic epochs and changes of $A(t)$ during $t_0 \leq t \leq t_1$.

A mobile version of the sorting problem of points on a line is a problem of sorting the edges of planar graphs. This edge sorting plays a key role in the point location problem on planar graphs. Thus, much is known about both fast construction and investigation of history [2], [4], [7].

For the two- (or three-) dimensional case, many algorithms for fast construction are known for convex polytopes; for example, there are collision detection algorithms for convex polyhedra [3].

Some theoretical results are given on investigation of history, based on the theory of the Davenport-Schinzel sequence [1], [5].

Voronoi diagram is not a discrete object. However, its 'topological structure' is discrete. Since the deformation of Voronoi diagram of a general mobile point set is rather expensive to maintain and is very naive for data degeneration, we shall deal mainly with a more specialized case, namely, the merging of two mobile Voronoi diagrams.

We note the existence of an algorithm running in linear time [6] for merging two static Voronoi diagrams. Hence, it is impossible to improve the theoretical cost with any preprocessing. Therefore, our main target is to investigate the history of the topological structure of the merged Voronoi diagrams. We can report all catastrophic epochs in $O(n^2 \log n)$ time. By storing the changes at these epochs in $O(n^2)$ space, we can write the Voronoi diagram at any epoch in $O(n)$ time.

1. Formulation of the problem.

Voronoi diagram.

Let $S = \{p_1, p_2, \ldots, p_n\}$ be a point set in plane. For any element p_i of S, the Voronoi region (or dominating region) of p_i is

(1.1) $R(p_i) := \{x\colon\ d(x, p_i) \leq d(x, p_j)\ \text{ for } \forall j = 1,2,..,n\}$,

where $d(x,y)$ is the Euclidean distance between a and b. The plane is then subdivided into the Voronoi regions of points of S. The union of the boundaries of Voronoi regions is regarded as a planar graph. We usually call this planar graph and its corresponding plane subdivision the Voronoi diagram of S, and denote it by $V(S)$. It is well-known that the Voronoi diagram $V(S)$ is constructed in $O(n \log n)$ time (optimal) and stored in $O(n)$ space [7]. By adding a point ∞ representing infinity, we can regard a Voronoi diagram as a graph on a sphere.

Suppose a Voronoi vertex v of $V(S)$ is surrounded by Voronoi regions $R(q_1), R(q_2), \ldots, R(q_k)$. Then all the points q_i, $i = 1,2...,k$ are located on a circle with center v. Hence, the multiplicity at the Voronoi vertex v is usually 3. We say that the vertex v has k-cocycular degeneration if its multiplicity k is more than 3. Note that if one of points around a k-cocycular degenerating vertex is ∞, then other surrounding points are located on a line. We often call this case (k-1)-colinear degeneration.

The dual graph $D(S)$ of $V(S)$ is called Delaunay triangulation, which is a real triangulation if $V(S)$ is nondegenerate.

'History' of deformation of Voronoi diagrams.

Let $S(t) = \{p_1(t), p_2(t), \ldots, p_n(t)\}$ be a set of moving points with respect to the time parameter t . $V(S(t))$ is its corresponding Voronoi diagram. To know the complete process of deformation of the topological structure of a geometric object, it is necessary to know all 'catastrophic epochs ' -- the moments when the object becomes singular.

Let us therefore consider the 'history ' of $V(S(t))$ during $0 \leq t \leq T$ for a fixed T. Intuitively, it is the list of all pictures when an edge appears or vanishes in the animation of deformation of the plane subdivision. The following well-known results are the key to our method.

Lemma 1.1.
If t_1 is a catastrophic epoch of the Voronoi diagram $V(S(t))$, then $V(S(t_1))$ contains a 4-cocycular degeneration.

Definition 1.2. Projective transformation [4].
We embed the plane as the x-y plane in a 3-dimensional space. For a point $p = (a, b)$ on the plane, we consider its projective dual plane (with respect to a conic) $H(p)$ defined by the following equation:

$L(p)\colon\quad z = ax + by - (a^2 + b^2)$.

Lemma 1.3. (4-cocycular test)
If a 4-cocycular degenerating vertex v is surrounded by the four regions corresponding to the points (q_1, q_2, q_3, q_4) of S, then the four planes $H(q_1)$, $H(q_2)$, $H(q_3)$, $H(q_4)$ meet at at least one point. In other words, the determinant of the corresponding affine coefficient matrix is zero.

Lemma 1.3 gives us a computable necessary criterion for degeneracy. Suppose t_1 is a catastrophic epoch of $V(S(t))$. Then at least one of the determinants corresponding to the four point subset of $S(t)$ is zero at $t = t_1$. Thus, a catastrophic epoch is a member of the set of roots of equations arising from those determinants.

Hence,the catastrophic epoch can be found by the following procedure.

1.4. Procedure.

1. *Solve the equations of the cocycular tests.*
2. *Select real catastrophic epochs from among those candidates.*
3. *Analyze the topological change at each epoch.*
4. *Store the change so that it can be retrieved efficiently.*

There are two drawbacks to studying general mobile configurations in this way. Firstly, we must solve equations on ${}_nC_4$ four-point subsets of S, notwithstanding the high cost of solving quadratic (or more complex) equations. Secondly, the analysis of the topological change at an epoch may become very heavy if the data are highly degenerate.

In the following sections, therefore, we deal with more specialized case, the problem of merging two mobile Voronoi diagrams.

Merging mobile Voronoi diagrams.

Let $P = \{p_1, p_2, \ldots, p_m\}$ and $Q = \{q_1, q_2, \ldots, q_n\}$ be two sets of fixed points. $Q(t) = \{q_1(t), q_2(t), \ldots, q_n(t)\}$ is a set of simultaneously moving points, defined as follows:

$$q_i(t) = R(t)q_i \; + f(t)$$

for rotation matrix function $R(t)$ and translating vector function $f(t)$ determined independently of i.

It suffices to assume some weak conditions on $R(t)$ and $f(t)$ in order to follow most parts of our procedure below. However, for simplicity, we deal here with a very simple case where $R(t)$ is the identity transformation and $f(t)$ is a linear translation.

$$q_i(t) = q_i + t\vec{\alpha}$$

for a fixed vector $\vec{\alpha}$. We often write simply q to represent $q(t)$ of $Q(t)$. Our target is the (merged) Voronoi diagram $V(t) = V(S(t))$ of the set $S(t) = P \cup Q(t)$.

To avoid unnecessarily high degeneration, we assume the following condition.

Condition 1.5.

1. *Both $V(P)$ and $V(Q)$ are non-degenerate.*
2. *For any t and for any i,j, $p_i \neq q_j(t)$, i.e. points are never overlapped.*

Our algorithm is divided into three procedures, principally following procedure 1.4.

1.6. Main Algorithm.

1. *FIND: Find the candidates of catastrophic epoch solving equations occurring in 4-cocycular tests.*
2. *SELECT: Select the real catastrophic epoch and determine the type of change.*
3. *STORE: Store the history efficiently.*

2. Algorithm

Find candidates using the 4-cocycular test.

We noted that $O(n^4)$ 4-cocycular tests should be done as a rule. However, in the case of the merging problem, we can show that it suffices to solve $O(nm)$ equations. Furthermore, since points are linearly translated, the equations are quadratic.

Lemma 2.1. *If the 4 points of $S(t)$ are 4-cocycular around a vertex of $V(t)$, then one of the following cases occurs:*

1. *Three of those points are points of P surrounding a Voronoi vertex of V(P), and the other is a point of Q(t).*
2. *Three of those points are points of Q(t) surrounding a Voronoi vertex of V(Q(t)), and the other is a point of P.*
3. *Two of those points are points of P whose Voronoi regions in V(P) share a Voronoi edge, and the other two are points of Q(t) whose Voronoi regions in V(Q(t)) share a Voronoi edge.*

Owing to the lemma 2.1, it suffices to check the 4-cocycular tests for the following patterns:

2.2. Tested patterns

1. *A Voronoi vertex of V(P) and a point of Q.*
2. *A Voronoi vertex of V(Q) and a point of P.*
3. *A Voronoi edge of V(P) and a Voronoi edge of V(Q) .*

We see examples of those patterns in Figure 2, in which dual terminology is used for visual simplicity. Since $V(P)$ and $V(Q)$ are spherical graphs with vertex multiplicity 3, the total numbers of test cases in 2.2 is $13nm - 11n - 11m + 9$.
$O(nm)$ catastrophic epochs can actually occur in the worst case. It is interesting and important in practice to diminish the average number of 4-cycular tests, since it is heavy work solving the quadratic equations. However, we shall not discuss this problem in the present article.

The procedure FIND is as follows.

FIND:

1. FIND1: prepare LIST1 , which is the list of candidates of catastrophic epochs derived from the 4-cycular tests of type 1 of 2.2.
2. FIND2: prepare LIST2 , which is the list of candidates of catastrophic epochs derived from the 4-cycular tests of type 2 of 2.2.
3. FIND3: prepare LIST3 , which are the lists of candidates of catastrophic epochs derived from the 4-cycular tests of type 3 of 2.2.
4. Merge: Merge LIST1, LIST2, LIST3 to get LIST_ROOTS, which is a sorted list with respect to time.

The procedures FIND1, FIND2, and FIND3 are essentially the same. Therefore, we only show FIND3.

```
Procedure: FIND3.
List_e(P) < = = =  All Voronoi edges of V(P), length M
List_e(Q) < = = =  All Voronoi edges of V(Q), length N
LIST3   < = = =  Vacant:  output list
   FOR i = 1
     DO
     Pick the i-th element $e_i$ from List_e(P).
     FOR j = 1
       DO
       Pick the j-th element $f_j$ from List_e(Q).
       Solve the quadratic equation corresponding to the 4-cocycular test on the vertex set of $e_i$ and $f_j$.

         IF the equation has a finite number (1 or 2) of roots in the range $0 \leq t \leq T$, put the pair
         consisting of the root (time label) and the quartet of points in LIST3 for each root.
         Moreover, if the root has multiplicity, label it with an asterisk.
         ELSE nop

       END DO
       UNTIL j = N
     END DO
     UNTIL i = M
END Procedure FIND3
```

Selection of real catastrophic epochs.

We now have a sorted list LIST_ROOTS including all catastrophic epochs. To select catastrophic epochs from this list, we apply "history updating " on the Voronoi diagram V(t) from $t = 0$ until $t = T$. First, let us study an easier case in which no pair of elements of LIST_ROOTS have the same time label. It is easy to see that Voronoi diagram V(t) has no Voronoi vertex v of degree more than 4 for any t; in short, no 5-point degeneration occurs.

Suppose that t_0 is an epoch and 4-point degeneration occurs at t_0 around a Voronoi vertex v, and the corresponding 4-points of $S(t_0)$ are a, b, c, d. We can choose a sufficiently small positive number ε such that there is no epoch during $t_0 - \varepsilon \leq t \leq t_0 + \varepsilon$. We then have the following lemma.

Lemma 3.1.

1. *In $V(t_0 - \varepsilon)$, the Voronoi regions $R(a)$, $R(b)$, $R(c)$, $R(d)$ surround an edge β. Without loss of generality, we assume $R(a)$ and $R(b)$ share β as their boundary edge.*
2. *In $V(t_0 + \varepsilon)$, the Voronoi regions $R(a)$, $R(b)$, $R(c)$, $R(d)$ surround an edge γ; moreover, $R(c)$ and $R(d)$ share γ.*

We see topological changes in Voronoi diagrams and their corresponding Delaunay diagrams in Figure 2. This lemma permits processing by the following algorithm.

Procedure: SELECT

```
LIST_EPOCH < = = Vacant: output list
 Construct the Voronoi diagram V(0).  -- Initial diagram.
 V = V(0)
   FOR i = 1
     DO
     (t; a, b, c, d) is the i-th element of LIST_ROOTS.
       IF it is labeled with an asterisk, then nop
       ELSE
         IF the Voronoi region R(a), R(b), R(c), R(d) of V surround an edge e of V, then change the
        topological structure of V into the other possible structure.
         Store (t; a, b, c, d) in LIST_EPOCH
         ELSE nop
     END DO
   UNTIL i = the length of LIST_ROOTS
 END procedure SELECT
```

Singular cases

In general, there may be more than two candidates in LIST_ROOTS with the same time label. Actually, higher degenerations may occur in that case. However, the following lemma shows that we can handle them at a low cost.

Lemma 3.2. *There is no 7(or more)-point degeneracy in $V(t)$.*

Hence, it suffices to consider 5 and 6-point degenerations.

First, we consider the case where 5-point degeneration occurs at time t. We assume all combinations of 4 points out of 5 are listed in LIST_ROOTS. Also we omit the easy case in which one of these candidates is labeled with an asterisk. The topological change of Delaunay diagram of {a, b, c, u, w} at t is then one of the two cases in Figure 4 if initial structure is the center pentagon with diagonals a_b and a_c. We see in Figure 4 that both possible terminal structures have the diagonal u_w. Thus it suffices to know the topological structure among {b, c, u, w}. This is determined by checking the sign of the gradient of the 4-cocycular test function at time t.

Next, we consider 6-point degeneration where all ${}_4C_6$ combinations of these points are listed in LIST_ROOTS at t. In this case, we prepare the signs of gradients of all 4-cycular tests of 15 possible combinations. Checking those signs, we can find the correct data structure from the 14 possible structures in Figure 5. We note that if no listed candidates at t is labeled with an asterisk , the rotation distance (cf.[8]) between initial structure and final structure is 3. Hence, the number of candidates of final structure is at most 5 (see Figure 5).

Finally, we consider 5 or 6-point degeneration where one of the combinations of 4 points is not listed in LIST_G. The Voronoi diagram may then have real 4-point degeneracy during the period beginning or terminating at t. It is easy to find the final structure from the initial structure in these cases.

Storing the history.

We have a sorted list LIST_EPOCH of length $O(nm)$. However, it takes O(nm) time to retrieve the topological structure of the Voronoi diagram $V(t)$ at $t = t_0$ from LIST_EPOCH. We can construct a data structure of size $O(nm)$ such that we can retrieve the topological structure at any epoch in linear time. This is done as follows:

1. Choose every $n + m$-th epoch from LIST_EPOCH
2. Construct the (whole) Voronoi diagram at each chosen epoch, and make a list LIST_VOLONOI of diagrams.
3. Store both LIST_EPOCH and LIST_VOLONOI.

Since a Voronoi diagram is stored linearly, the storage size of LIST_VOLONOI is $O(nm)$. Further, it is evident that the time needed to retrieve the topological structure is O(n + m).

Estimation of Algorithm.

The estimation of cost is as follows:

- We need $O(nm)$ operations to find LIST1, LIST2, and LIST3. These operations involve solving quadratic equations.
- We need $O(nm \log nm)$ operations to sort the lists to get LIST_ROOTS.
- We need $O(nm)$ operations to do SELECT.
- We need $O(nm \log(n + m))$ operations to make LIST_VORONOI from LIST_EPOCH.

Acknowledgement: The author thanks Prof. Koukichi Sugihara of Tokyo University for his helpful comments and discussion.

References

[1] M.J.Atallah. "Dynamic computational geometry," *Proc.24-th IEEE Symp. on Foundations of Computer Science,* 1983, pp 92-99

[2] B.Chazelle, "How to search in history," *Information and Control 64* 1985, pp 77-99

[3] D.P.Dobkin and D.G.Kirkpatrick, "Fast detection of polyhedral intersection," *Theoretical Computer Science 27* 1983, pp.241-253

[4] H.Edelsbrunner, *Algorithms in Combinatorial Geometry,* Springer Verlag, 1987

[5] H.Edelsbrunner, J.PAch, J.T.Schwartz, and M.Sharir, "On lower Envelope of bivariate functions and its applications," *Proc.27-th IEEE Symp. on Foundations of Computer Science,* 1987, pp 27-37

[6] D.G.Kirkpatrick, "Efficient computation of continuous skeletons," *Proc. 20th IEEE Symp. on Foundations of Computer Science,* 1979, pp18-27.

[7] F.P.Preparata and M.I.Shamos, *Computational Geometry - An Introduction* Springer Verlag, 1985

[8] D.D.Sleator, R.E.Tarjan, and W.P.Thurston, "Rotation distance, triangulation, and hyperbolic geometry," *ACM Symp. on Theory of Comput.,* 1986, pp.122-135

Figure 1. Deformation of Voronoi diagram. Translating marked points, left diagram is deformed into right one.

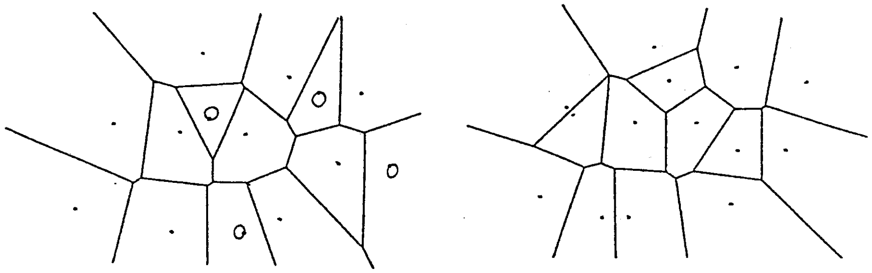
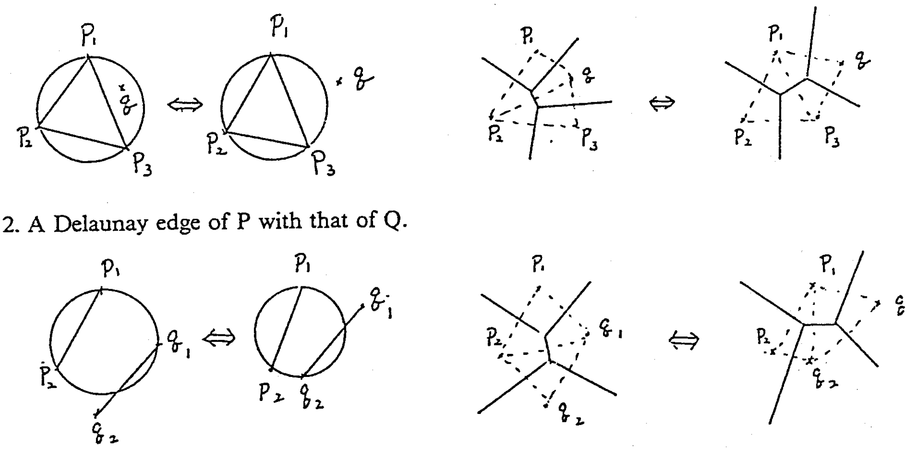


Figure 2. Patterns of local catastrophe.

1. Topological change of Voronoi diagram among three neighbored points (Delaunay triangle) of P and a point of Q.



2. A Delaunay edge of P with that of Q.

Figure 3. Catastrophe at 5-cocycular degeneration.

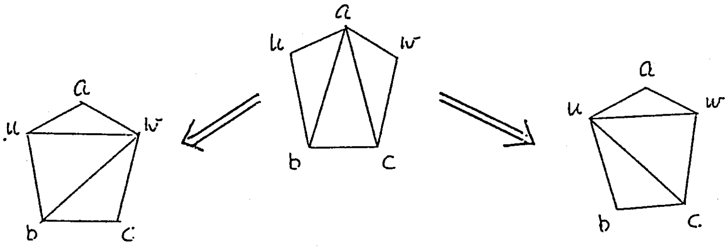


Figure 4. Rotation graph of Delaunay triangulation on 6-points [8].