Dynamic Voronoi Diagrams and Delaunay Triangulations

Chanderjit L. Bajaj * *William J. Bouma*†

Department of Computer Science,
Purdue University,
West Lafayettte, IN 47907

June 19, 1990

We present a topologically robust algorithm for dynamically maintaing a Voronoi diagram and its dual Delaunay triangulation for a set of points moving independently in the plane along trajectories which are analytic functions of time. Using rational parametrizations of surfaces one is also able to map the dynamic planar Delaunay triangulations to dynamic Delaunay triangulations on surfaces. Constant velocity motion of n points in the plane requires $O(\log n)$ time per topological change to maintain the Voronoi diagram. The number of topological changes is bound by $O(n)$ for a unit time step within which points can simultaneously move at most into a single neighboring Voronoi cell. The algorithm has been implemented in Common Lisp on a Symbolics 3620.

1 Introduction

For a static point set in the plane or in space, there exist various solutions to numerous geometric problems including the computation of Voronoi diagrams and Delaunay triangulations, (see for e.g.[5, 7]). For a dynamic set of points in the plane, many fewer algorithms are known. These include algorithms for computing the time at which the convex hull of algebraically moving points in the plane reaches a steady state [1, 4], computation of all possible point coincidences for points moving in straight lines [6] and one and two dimensional dynamic obstacle avoidance problems [8].

In this paper we describe a simple algorithm for dynamically maintaining a Voronoi diagram and its dual Delaunay triangulation for a set of points moving independently in the plane along trajectories which are analytic functions of time. Using rational parametrizations of surfaces [2] one is also able to map the dynamic planar Delaunay triangulations to dynamic Delaunay triangulations on surfaces.

The algorithm relies on the key fact that changes in topology of the Voronoi diagram are occurrences or events that happen at discrete intervals of time even though the points are moving continuously as analytic functions of time. The basic idea then is to determine for an interval of time $\mathcal{T}$ the discrete times $\tau \epsilon \mathcal{T}$ when such topological events occur and in time-stamp order update the Voronoi diagram at τ to a valid topology for time after τ.

*Supported in part by NSF grant DMS 88-16286 and ONR contract N00014-88-K-0402
†Supported in part by NSF grant CCR 86-19817

Our motivation in considering the dynamic Voronoi problem arose from our simulation of motion coordination trajectories of multiple objects, simultaneously moving and avoiding collisions. Each object avoids collisions with only his Voronoi neighbors. The dual dynamic Delaunay triangulations, particularly for surfaces, provides dynamic triangular meshes for finite element calculations of time dependent problems.

2 Algorithmic Details

2.1 Notation

We use the term sites for the given points. The Voronoi diagram of the sites consists of a set of line segments called bisectors. Each point on a bisector lies equidistant from two of the sites. An endpoint of a bisector, a Voronoi vertex, is equidistant from three sites which are known as the generators of the vertex. Thus each Voronoi vertex is the center of the circumscribing circle of its generators.

2.2 The Method

Let the given moving sites s_i have coordinates $(s_i.px(t), s_i.py(t))$, $1 \leq i \leq n$. The bisector B_{ij} for each of two adjacent sites s_i and s_j in the Voronoi diagram is said to *vanish* at the moment its two vertex endpoints v_1 and v_2 become equal. The vanishing of any bisector B_{ij} signals a topological occurence or event. Furthermore all topological changes in the Voronoi diagram must involve the vanishing of some bisector.

Detection of a topological event is achieved as follows. Let Voronoi vertices v_1 and v_2 be the endpoints of a bisector just prior to its vanishing. Consider v_1 as being generated by s_i, s_j, and some site s_k. Likewise consider v_2 as being generated by s_i, s_j, and some other site s_l. It is easy to see from the Delaunay circle property that $v_1 = v_2$ at precisely the time all the generators s_i, s_j, s_k, s_l lie on the same circle. Testing if four points lie on a circle can be done by checking if the 4x4 determinant below is zero.

$$J(s_i, s_j, s_k, s_l) = \begin{vmatrix} 1 & s_i.px & s_i.py & (s_i.px^2 + s_i.py^2) \\ 1 & s_j.px & s_j.py & (s_j.px^2 + s_j.py^2) \\ 1 & s_k.px & s_k.py & (s_k.px^2 + s_k.py^2) \\ 1 & s_l.px & s_l.py & (s_l.px^2 + s_l.py^2) \end{vmatrix}$$

To find the time a bisector vanishes we substitute the site equations into the determinant and solve the resulting polynomial for t. Practically the 4x4 determinant calculation is replaced with an equivalent but more numerically stable calculation involving three 2x2 determinants associated with Voronoi vertices (see [9]). For efficiency, when we compute the three 2x2 determinants we store them with the associated Voronoi vertex for future use in incident bisector vanish calculations. Suppose $s_i.px(t)$ and $s_i.py(t)$ are the polynomials describing the motion of site s_i in the plane. The determinants for v generated by s_i, s_j, and s_k are then:

$$J_2(t) = \begin{vmatrix} s_i.py - s_k.py & \frac{1}{2}((s_i.px - s_k.px)^2 + (s_i.py - s_k.py)^2) \\ s_j.py - s_k.py & \frac{1}{2}((s_j.px - s_k.px)^2 + (s_j.py - s_k.py)^2) \end{vmatrix}$$

$$J_3(t) = \begin{vmatrix} s_i.px - s_k.px & \frac{1}{2}((s_i.px - s_k.px)^2 + (s_i.py - s_k.py)^2) \\ s_j.px - s_k.px & \frac{1}{2}((s_j.px - s_k.px)^2 + (s_j.py - s_k.py)^2) \end{vmatrix}$$

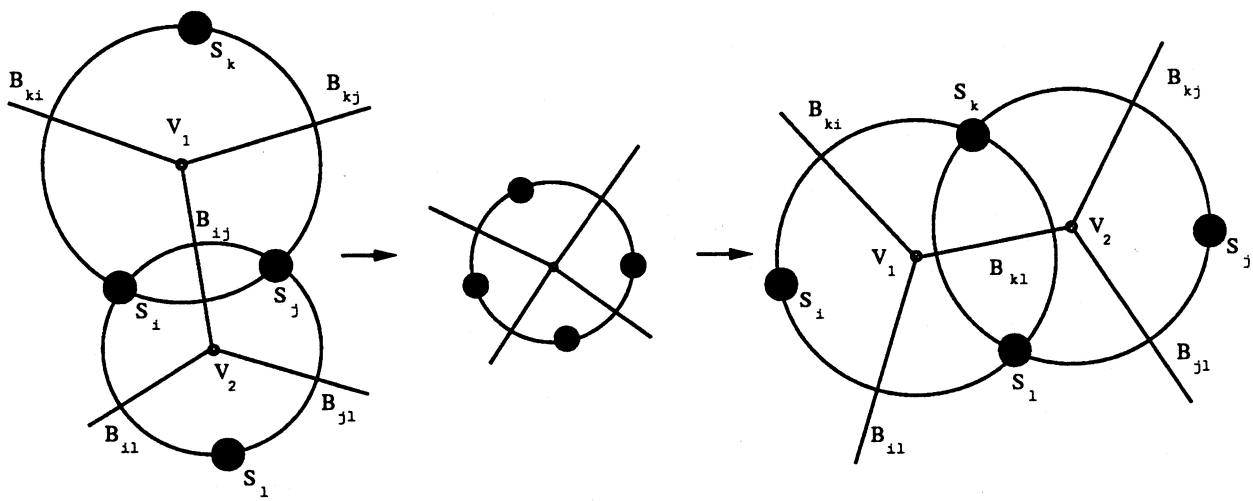
$$J_4(t) = \begin{vmatrix} s_i.px - s_k.px & s_i.py - s_k.py \\ s_j.px - s_k.px & s_j.py - s_k.py \end{vmatrix}$$

If s_l is the other site (besides the generators of v) that is generating B_{ij}, the polynomial that determines when B_{ij} will vanish is then :

$$\begin{aligned} J(t) &= J2 * (s_l.px - s_k.px) - J3 * (s_l.py - s_k.py) \\ &\quad + \frac{1}{2} * J4 * ((s_l.px - s_k.px)^2 + (s_l.py - s_k.py)^2) \end{aligned}$$

If each px and py are algebraic curves of maximum degree d in space-time coordinates, then the degree of polynomial J is at most $4d$. All other detection methods of the bisector vanishing that we considered yielded polynomials of much higher degree.

Again, because of constant velocity motion of the sites, just after the vanishing of bisector B_{ij}, a new bisector B_{kl} is formed between s_k and s_l. Thus a vanishing bisector event does not change the total number of bisectors in the graph, and can be handled by a simple change of topological information for the associated generator sites, which we call a *swap*. The two other bisectors incident to v_1 are B_{ki} and B_{kj} and the bisectors incident to v_2 are B_{il} and B_{jl}. The data structure for the bisector holds its two vertices and the two sites it divides. The structure for a vertex contains its three incident bisectors. The Voronoi diagram is thus updated by replacing s_i and s_j with s_k and s_l in B_{ij} to yield B_{kl}. We swap B_{kj} and B_{il} in v_1 and v_2 and then swap v_1 and v_2 in B_{kj} and B_{il}.



2.3 Rational Algebraic Surfaces

Having obtained a dynamic Delaunay triangulation of moving sites in a plane one is able to transfer this to a dynamic triangulation of points on a rational algebraic surface. Such triangulations prove invaluable for time varying finite element analysis. A *rational* algebraic surface can be represented by the triple $(x = G_1(s,t), y = G_2(s,t), z = G_3(s,t))$, where G_1, G_2 and G_3 are rational functions in s and t with a common denominator. See for example [2] for definitions and an extensive bibliography. Rational surfaces include all degree two surfaces (e.g. spheres, cones, ellipsoids, hyperboloids, etc.), most degree three surfaces and special degree four and higher degree surfaces (e.g. steiner). The sites are chosen in the $s-t$ plane and correspond in a one-to-one manner to points (x,y,z) on the surface. Of course, bad points which are zeroes of the denominator $h(s,t)$ of the rational functions $G_i(s,t) = \frac{g_i(s,t)}{h(s,t)}$, are avoided. These bad points are

confined to a single real algebraic curve $h(s,t) = 0$ in the $s-t$ plane. Additionally a few other finite set of bad points arise from the common real zeroes of $g_i(s,t) = h(s,t)$, $i = 1,...,3$.

Surface parameterizations are also invertible. That is there exist global rational functions H_1 and H_2 such that $s = H_1(x,y,z), t = H_2(x,y,z)$. See [3] where a method is given to construct such inverse mappings. This inverse mapping enables control of the dynamic sites directly from points on the rational surface in x,y,z space.

2.4 Complexity

Within a unit time step the total number of topological events is dependent on the number of Voronoi cells crossed by all the sites. If we assume that each site crosses over to at most a neighboring Voronoi cell in a unit time step, then the number of topological occurrences for the simultaneous motion of all sites is bound by $O(n)$. There is an inherent time-stamp ordering of topological events via the heap priority-queue which results in an additional $O(n \log n)$ processing time for all events. The update time of the Voronoi per event is bound by $O(\log n)$ spent in maintaining the event priority-queue.

2.5 Implementation

The above algorithm is implemented on a Symbolics 3620 with a few additional enhancements for controlling numerical precision. Bisector vanish times are computed by invoking a simple polynomial real roots solver. To keep the degree of intermediate computation small, first order approximations are considered for complex motion trajectories. Within each unit time step the motion is considered linear, i.e. constant velocity. A unit time step is the maximum time required for any site to move into a neighboring cell. No apriori velocity bounds are made, nor restrictions placed that all time steps are equal, allowing adaptive, curvature dependent linear approximations of the trajectory. With the restriction that within each unit time step the motion of the sites is with constant velocity, bisector vanish polynomial J has a maximum degree of 4. Thus it is not too difficult to solve for all real roots to acceptable accuracy. The plausible bisector vanish times (those between the current clock time and the end of the unit time step) are saved in a heap priorty-queue. To eliminate special cases of Voronoi vertices at infinity, as well as vanishing bisectors at infinity, four dummy sites are placed defining a bounding box enclosing the given sites and all their possible trajectories.

2.6 Degeneracies

There are rare ocassions when the bisector polynomial becomes identically zero during motion of sites (a degeneracy). This occurs when two moving sites are at the same point at the same instant of time. The rareness of this event can be seen from the fact that arbitrary lines in space seldom intersect (if and only if the two lines are coplanar). Furthermore, line segments in space intersect if and only if their four endpoints in space-time are collectively coplanar. Constant velocity motion of sites are straight line segments within space-time coordinates x,y,t with the slope of the lines being the velocity. Knowing the starting and final endpoints of each site and its velocity within a unit time step, the occurrences of degeneracies can be quickly computed by checking for coplanarity of the endpoints (a 4x4 determinant calculation with linear entries). If a degeneracy is possible then, the assigned velocity of the involved sites are slightly perturbed prior to motion to eliminate the degenerate occurrence.

3 Conclusion

An indirect result of our dynamic algorithm is a new method to create Voronoi diagrams and Delaunay triangulations for arbitrary distribution of points in a plane. We achieve this by placing the points initially

in a regular pattern in the plane for which the diagram is obvious, then moving them in one time step to their actual positions. Typically the points need not move very far in the intended applications for an optimal $O(n \log n)$ computation of the Voronoi diagram.

References

[1] Atallah, M., (1985) "Some Dynamic Computational Geometry Problems", *Computers and Math. with Applications*, 11, 1171 - 1181.

[2] Bajaj, C., (1989) "Geometric Modeling with Algebraic Surfaces", *The Mathematics of Surfaces III*, ed. by D. Handscomb, 3 - 48.

[3] Bajaj, C., Garrity , T., and Warren, J., (1988) "Applications of Multi-equational Resultants" *Purdue University, Computer science Tech Rept.* CSD TR-826.

[4] Chazelle, B., (1984) "Fast Searching in a Real Algebraic Manifold with Applications to Geometric Complexity", *Brown University, Computer science Tech Rept.* CS 84-13.

[5] Edelsbrunner, H., (1987) *Algorithms in Combinatorial Geometry*, Springer Verlag, New York.

[6] Ottmann, T., and Wood, D., (1984) "Dynamical Sets of Points", *University of Waterloo, Computer science Tech Rept.*.

[7] Preparata,, F., and Shamos, M., (1985) *Computational Geometry, An Introduction*, Springer Verlag, New York.

[8] Reif, J., and Sharir, M., (1985) "Motion Planning in the Presence of Moving Obstacles", Manuscript.

[9] Sugihara, K., (1990) "Algorithms for Computing Voronoi Diagrams", *Spatial Tesselations - Concepts and Applications of Voronoi Diagrams*, ed., Books, Okabe, and Sugihara, John Wiley (to appear).