Topological Considerations for the Incremental Computation of Voronoi Diagrams of Circular Arcs

Martin Held* Stefan Huber†

Abstract

We study the computation of Voronoi diagrams of points, straight-line segments and circular arcs in the two-dimensional Euclidean plane. Our algorithm is based on a randomized incremental insertion of the sites and makes use of the topology-oriented approach by Sugihara et alii. It was implemented and integrated into the Voronoi package VRONI. However, in this extended abstract we focus only on the topological and graph-theoretic details of an insertion.

1 Introduction

1.1 Motivation

Voronoi diagrams of straight-line segments turned out to be useful in a variety of applications with a geometric flavor. We note that supporting circular arcs is important for the practical application of a Voronoi algorithm: offsetting a polygon introduces circular arcs, and it is generally requested that the result of an offsetting operation can again be used as input for a Voronoi algorithm. Handling circular arcs as genuine arcs is imperative in the PCB[1] business, since PCB data may be huge; typically, one cannot afford to replace every arc by tens or even hundreds of straight-line segments as this would cause the memory footprint of a Voronoi-based application to sky-rocket.

1.2 Basic Definitions

For two points $p, q \in \mathbb{R}^2$, let $d(p,q)$ denote the Euclidean distance between p and q. If $Q \subset \mathbb{R}^2$ is a set then $d(p,Q)$ is defined as $\inf\{d(p,q) : q \in Q\}$. Similarly, if $\mathcal{Q}$ is a set of a finite number of sets, then $d(p,\mathcal{Q}) := \min_{Q\in\mathcal{Q}} d(p,Q)$.

In the sequel we explain how to compute $\mathcal{VD}(S)$, where S is a set of points, straight-line segments and circular arcs. For technical reasons, we regard a line segment or a circular arc as the union of three objects: an open segment/arc and its two end points. Furthermore, we assume that every arc is oriented counterclockwise (CCW), and that no arc is greater than a semi-circle[2]. Points, open straight-line segments and open circular arcs are called *sites*. If for every open segment and arc of S its end points also belong to S and if no sites intersect pairwise then S is called a *proper* input set.

For a vector v and a point p, let $H(p,v)$ be the half-plane $\{q \in \mathbb{R}^2 : q \cdot v \geq p \cdot v\}$. The result of the rotation of v around the origin by 90^o is denoted by v^{CCW}, while v^{CW} stands for a rotation by -90^o. Following [3], the *cone of influence* $\mathcal{CI}(s)$ of a site s is defined as $\mathcal{CI}(s) := \mathbb{R}^2$ if s is a point, $\mathcal{CI}(s) := H(a, b-a) \cap H(b, a-b)$ if s is a segment with end points a and b, and $\mathcal{CI}(s) := H(c,(s-c)^{\mathrm{CCW}}) \cap H(c,(e-c)^{\mathrm{CW}})$ if s is an arc centered at c with start point s and end point e. We define the *Voronoi cell* of a site $s \in S$ as $\mathcal{VC}(s,S) := \mathrm{cl}\{q \in \mathrm{int}\,\mathcal{CI}(s) : d(q,s) \leq d(q,S)\}$, where $\mathrm{int}\,Q$ denotes the (topological) interior of the set Q and $\mathrm{cl}\,Q$ stands for the closure of Q. (The consideration of the interior and exterior in the definition of $\mathcal{VC}(s,S)$ is a technical twist[3] in order to avoid undesired "one-dimensional" portions of a Voronoi cell if two circular arcs meet tangentially in an end point such that the interiors of their cones of influence overlap.) The *Voronoi polygon* $\mathcal{VP}(s,S)$ is given by the boundary of $\mathcal{VC}(s,S)$, and the *Voronoi diagram* $\mathcal{VD}(S)$ of S is defined (as usual) as $\mathcal{VD}(S) := \bigcup_{s\in S}\mathcal{VP}(s,S)$. For two sites $s_1, s_2 \in S$, the *bisector* $b(s_1,s_2)$ is defined as the loci of points out of $\mathcal{CI}(s_1)\cap\mathcal{CI}(s_2)$ which are equidistant to s_1 and s_2. A *Voronoi edge* between s_1, s_2 is a connected portion of $\mathcal{VP}(s_1,S)\cap\mathcal{VP}(s_2,S)$; it lies on $b(s_1,s_2)$. *Voronoi nodes* are points where three or more Voronoi edges meet. The *clearance disk* $\mathcal{CD}(p,S)$ of a point $p \in \mathbb{R}^2$ is the closed disk centered at p with *clearance* radius $r := d(p,S)$.

1.3 Prior and Related Work

A worst-case optimal $O(n \log n)$ algorithm for the computation of the Voronoi diagram of n points, straight-line segments and circular arcs was intro-

*Universität Salzburg, FB Computerwissenschaften, A–5020 Salzburg, Austria; held@cosy.sbg.ac.at; Partially supported by Austrian FWF Grants L43-N12 and L367-N15.

†Universität Salzburg, FB Computerwissenschaften, A–5020 Salzburg, Austria; shuber@cosy.sbg.ac.at; Supported by Austrian FWF Grant L43-N12.

[1]Printed-circuit board.

[2]We split arcs greater than semi-circles.

[3]Yap [5] resorts to ϵ-neighborhoods. Alt and Schwarzkopf [1] consider cells that are partially open; as a consequence, the intersection between adjacent cells may be empty.

duced by Yap [5]. At least in theory, Fortune's sweep-line algorithm [2] is also applicable to circular arcs. However, for both algorithms we are not aware of an actual implementation that handles circular arcs.

More recently, Alt and Schwarzkopf [1] studied Voronoi diagrams of so-called "harmless sites" (which include circular arcs). They first select one point out of the relative interior of each curve and then insert the curves in a randomized order, obtaining an expected running time of $O(n \log n)$. However, their paper focuses mostly on establishing a theoretical basis for the definition of Voronoi diagrams of planar curves, while the actual algorithmic aspect of the insertion of a curve is only sketched. In any case, no implementation of their algorithm is known.

1.4 Survey of the Voronoi Algorithm

The success of Held's Voronoi package Vroni [3] motivated us to extend its construction scheme to points, straight-line segments and circular arcs in the two-dimensional Euclidean plane. Once again we resort to the topology-oriented approach by Sugihara et al. [4]. Starting with an initially empty set of processed sites, the final Voronoi diagram is obtained by incrementally adding one new site at a time to the set of processed sites and updating the Voronoi diagram accordingly.

Every update of the Voronoi diagram is performed by deleting old Voronoi nodes (and creating new Voronoi nodes). As in the case of segment Voronoi diagrams, care has to be taken in order to prevent the removal of cycles of Voronoi edges while deleting Voronoi nodes during an incremental update. However, the insertion of circular arcs causes problems to surface that do not occur for segment Voronoi diagrams. In the sequel, we discuss all topological and graph-theoretical extensions of the incremental insertion needed for handling circular arcs.

Our new algorithm has been implemented in ANSI C and integrated into Vroni. We emphasize, though, that the basic scheme presented below for incrementally inserting a circular arc into a Voronoi diagram is not bound to the limits of Vroni.

2 The Algorithm

Let S be a set of sites and consider an open arc $s \notin S$ that is to be inserted into $\mathcal{VD}(S)$. Let $S^+ := S \cup \{s\}$. We assume that S^+ is a proper input set and that $\mathcal{VD}(S)$ is known. In order to insert the arc s into $\mathcal{VD}(S)$, we proceed as follows:

1. We mark a Voronoi node ("seed node") of $\mathcal{VD}(S)$ whose clearance disk is intersected by s.

2. We scan $\mathcal{VD}(S)$ and mark all other nodes whose clearance disks are intersected by s.

3. We remove all those Voronoi edges of $\mathcal{VD}(S)$ which have both nodes marked, and compute new nodes[4] on the edges with only one node marked.

While the second task boils down to a simple graph traversal that is identical to the case of segment Voronoi diagrams [3], the first and the third task require some caution, as explained below.

2.1 Selecting a Seed Node

A *seed node* is a node of $\mathcal{VD}(S)$ which lies in $\mathcal{VC}(s, S^+)$. Thus, its clearance disk is intersected by s and it needs to be removed. We search for a seed node in $\mathcal{VP}(p, S) \cap \mathcal{CI}(s)$, where p is the start point of s. If one or more candidates for a seed node exist in $\mathcal{VP}(p, S) \cap \operatorname{int} \mathcal{CI}(s)$, then we select the node whose clearance disk is violated the most: We pick the node v such that $d(v, S) - d(v, s)$ is minimized. A second seed node is selected within the Voronoi polygon of the end point of s. In any case, we do not select a node as seed node, or mark it in the subsequent scan of $\mathcal{VD}(S)$, if it coincides with the start or end point of s. If, however, no node of $\mathcal{VP}(p, S)$ lies within $\operatorname{int} \mathcal{CI}(s)$ then one can prove that there exist nodes of $\mathcal{VP}(p, S)$ on $\operatorname{bd} \mathcal{CI}(s)$, and we distinguish the following cases.

2.1.1 Selecting a Seed Node in the Presence of Tangential Sites

Suppose that s meets exactly one site $s' \in S$ tangentially in the common end point p. Let e_1, e_2 be the two Voronoi edges that emanate from p, see Fig. 1. We note that e_1 and e_2 lie on the same supporting line g through p.

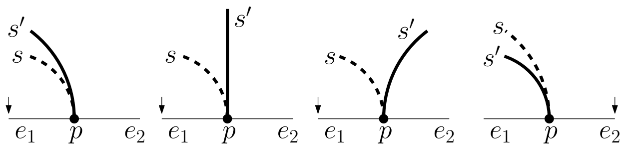

Figure 1: Selecting a seed node if sites meet tangentially.

The start node shared by e_1 and e_2 is excluded from further consideration because we do not select as seed node a node that coincides with the point p. Since $\mathcal{VP}(p, S) \cap \mathcal{CI}(s) \subset g$, the two other nodes on e_1 and e_2 are the only nodes of $\mathcal{VP}(p, S)$ that can be selected as seed node. Suppose that the center of s is on the side of e_1 relative to g and p. We base our decision on the relative order of the sites incident upon p:

- Case: s' is an arc.
  If the center of s' is on the side of e_2 then the node on e_1 is admissible as seed node. If the

[4] The computation of the new Voronoi nodes is explained in the full version of this paper.

center of s' is on the side of e_1 and the radius of s' is greater than the radius of s, then the node on e_1 is admissible; otherwise, the node on e_2 is admissible.

- Case: s' is a segment.
  The node on e_1 is admissible.

In Fig. 1 the little arrows point to the edge on which we select the seed node. If, however, more sites are incident upon p then we need to handle spikes; see below.

2.1.2 Selecting a Seed Node in the Presence of Spikes

Suppose that several segments or arcs meet in a common end point p. We scan all nodes of the Voronoi polygon $\mathcal{VP}(p, S)$. If $\mathcal{VP}(p, S)$ has a node v with a clearance greater than zero then v does not coincide with p. In this case we proceed as normal: if v lies on $\mathrm{bd}\,\mathcal{CI}(s)$ then we evaluate $d(v, S) - d(v, s)$. If some sites meet tangentially at p we also have to check whether v is admissible; see above. Again, we select that (admissible) node as seed node whose clearance disk is violated the most.

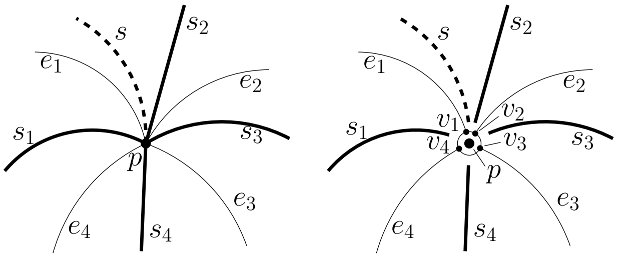

Figure 2: Selecting a seed node when multiple sites meet in a common end point. Left: Geometric view. Right: Topological view.

Otherwise, if no such (admissible) node exists, then we scan the Voronoi edges that are incident upon the nodes which coincide with p. (In Fig. 2, these nodes are numbered $v_1, \ldots, v_4$.) For such a node v_i we consider the Voronoi edge e_i incident upon v_i whose second node does not belong to $\mathcal{VP}(p, S)$. (If no such edge is incident upon v_i then we originate a recursive search in $\mathcal{VD}(S)$, starting at v_i.) For every such edge e_i it is tested whether s intersects the clearance disk of its second node. (One can prove[5] that such a suitable node always exists.)

We emphasize that nodes which coincide with an input point are never deleted during an incremental update. Therefore, it is guaranteed that in the final Voronoi diagram every point site will have a Voronoi region associated with it; it may have zero area, though.

[5]All proofs are given in the full version of this paper.

2.2 Removing a Tree of Voronoi Edges

Assume that two seed nodes have been determined. Starting at one seed node we recursively scan the Voronoi diagram $\mathcal{VD}(S)$ and mark all those nodes whose clearance disks are intersected by s. Obviously, all those nodes need to be deleted since they cannot belong to $\mathcal{VD}(S^+)$. Similarly, it seems natural to remove a Voronoi edge if both of its nodes have been marked for deletion. However, we have to check whether a cycle exists within the portion T of $\mathcal{VD}(S)$ which is marked for deletion. It can be shown that those portions of edges of $\mathcal{VD}(S)$ which are completely contained in $\mathcal{VC}(s, S^+)$ form a tree. In other words, T contains a cycle if and only if T contains an edge of $\mathcal{VD}(S)$ which has both nodes marked but needs to be preserved partly. Figure 3 depicts a (dashed) circular arc whose insertion would cause the removal of all nodes of the Voronoi cells of its two end points.

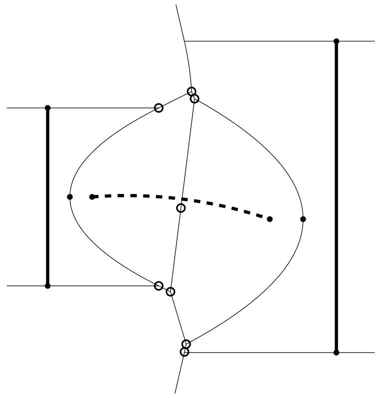

Figure 3: The insertion of the dashed arc causes the nodes depicted by circles to be marked. Splitting the two parabolic arcs at their apices avoids the (incorrect) removal of two Voronoi cells.

Suppose that e is an edge whose two nodes are marked for deletion but which needs to be preserved partly. We can distinguish two cases, depending upon whether or not e is completely contained in $\mathcal{CI}(s)$.

2.2.1 Voronoi Edge Partly Outside of Cone of Influence

Consider a Voronoi edge and assume that the apex of its supporting conic lies on the edge. As suggested in [3], we insert a degree-two node in order to split a conic Voronoi edge at its apex. (See Fig. 3.) Hence, for the sequel we may assume that no Voronoi edge has the apex of its supporting conic in its (relative) interior. Then one can prove that every Voronoi edge of a segment Voronoi diagram either does not have both nodes marked for deletion or is completely contained in the current cone of influence of the new segment to be inserted. Unfortunately, once we deal with circular arcs the insertion of apex nodes is of limited help for avoiding cycles: even if all Voronoi edges are split at the apices of their supporting conics, it still is possible that both nodes of an edge e are marked for deletion while e is not completely contained in $CI(s)$, as illustrated in Fig. 4.

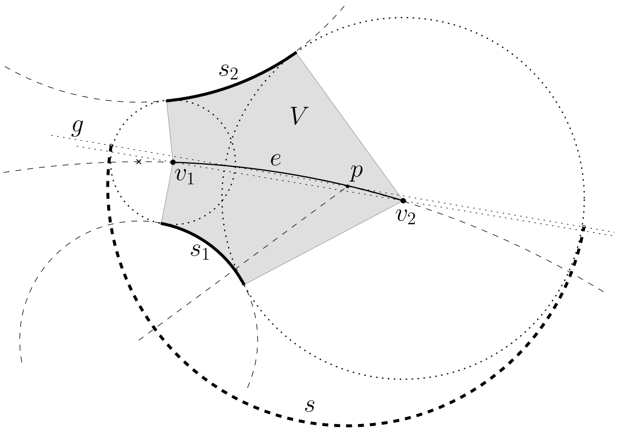

Figure 4: The insertion of the arc s causes both nodes of e to be marked although some portion of e lies outside of $CI(s)$.

This sample arrangement of input sites is constructed as follows: We consider the hyperbolic bisector between two circles that are disjoint. We choose two arcs s_1, s_2 on these circles and the corresponding Voronoi edge e on their bisector such that e does not contain the apex in its interior and such that $\mathcal{VC}(s_1, S)$ contains all secants of e. We denote by v_1 (resp. v_2) that node of e which has smaller (resp. larger) clearance. We want to insert an arc s such that the clearance disks of v_1 and v_2 are intersected by s, even though e is not completely contained in $\mathcal{CI}(s)$.

Let V be the union of all line segments resulting from the normal projection of points of e onto s_1 and s_2. Now consider the supporting line g of a secant of e that is parallel to the line through v_1 and v_2: We get that $g' := g \setminus \mathrm{int}(V \cup \mathcal{CD}(v_1) \cup \mathcal{CD}(v_2))$ consists of two parts because the set $g \cap \mathrm{int}(V \cup \mathcal{CD}(v_1) \cup \mathcal{CD}(v_2))$ is connected. Within each part of g' we choose a point close enough to the neighboring clearance disk such that the line through this point orthogonal to g intersects this clearance disk. All that remains to do is to use these two points as the end points of a semi-circle (which has to lie on that side of g which contains s_1). By construction, s intersects the clearance disks of v_1 and v_2. Also by construction, some portion of e does not lie on the side of v_1, v_2 relative to g. Thus, this portion of e is outside of $CI(s)$! This construction scheme can be adapted to nearly every combination of input sites s_1, s_2 as long as e does not take on the form of a straight-line segment.

We solve this problem by inserting a dummy degree-two node p that breaks up the cycle: We always find a proper point p on e by considering the normal projection of the center of s onto s_1, and by intersecting the resulting projection line with e. The resulting node p need not lie outside of $\mathcal{CI}(s)$ but one can prove that it will never lie in the future Voronoi cell $\mathcal{VC}(s, S^+)$.

2.2.2 Voronoi Edge Completely Contained in Cone of Influence

Now suppose that e is a Voronoi edge of $\mathcal{VD}(S)$ which does not lie completely within $\mathcal{VC}(s, S^+)$ although both of its nodes are marked and although it is contained completely in $\mathcal{CI}(s)$. As Fig. 5 illustrates, the site s_1 enclosed by the cycle that contains the Voronoi edge e may be a segment or arc. Both nodes of e are marked even though a point p exists on e whose clearance disk is not violated by s.

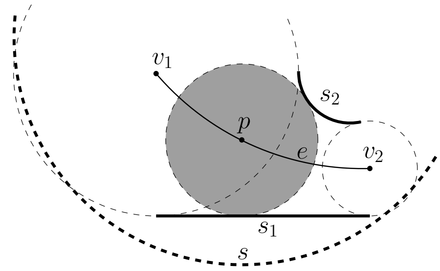

Figure 5: Both nodes v_1, v_2 of the edge e are marked even though some portion of e has to be preserved.

Fortunately, the same strategy that we used in Sec. 2.2.1 to break up a cycle is applicable once more: we consider the normal projection of the center of s onto s_1 and intersect the resulting projection line with e in order to obtain a split point p. Summarizing, we get a uniform strategy for breaking up cycles.

References

[1] H. Alt and O. Schwarzkopf. The Voronoi Diagram of Curved Objects. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 89–97, Vancouver, BC, Canada, 1995.

[2] S. Fortune. A Sweepline Algorithm for Voronoi Diagrams. *Algorithmica*, 2(2):153–174, 1987.

[3] M. Held. VRONI: An Engineering Approach to the Reliable and Efficient Computation of Voronoi Diagrams of Points and Line Segments. *Comput. Geom. Theory and Appl.*, 18(2):95–123, 2001.

[4] K. Sugihara, M. Iri, H. Inagaki, and T. Imai. Topology-Oriented Implementation - An Approach to Robust Geometric Algorithms. *Algorithmica*, 27(1):5–20, 2000.

[5] C. Yap. An $O(n \log n)$ Algorithm for the Voronoi Diagram of a Set of Simple Curve Segments. *Discrete Comput. Geom.*, 2(4):365–393, 1987.