Discrete Comput Geom 2:365-393 (1987)

Discrete & Computational Geometry

An $O(n \log n)$ Algorithm for the Voronoi Diagram of a Set of Simple Curve Segments*

Chee K. Yap

Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, NY 10012, USA

Abstract. Let X be a given set of n circular and straight line segments in the plane where two segments may interest only at their endpoints. We introduce a new technique that computes the Voronoi diagram of X in $O(n \log n)$ time. This result improves on several previous algorithms for special cases of the problem. The new algorithm is relatively simple, an important factor for the numerous practical applications of the Voronoi diagram.

1. Introduction

The ubiquitous Voronoi diagram has been studied in areas such as biology, solid-state physics, pattern recognition, geography, stock-cutting, wire layout, geometric optimization, facilities location, computer graphics, and robotics (see [5], [9], [12] for extensive references). In some literature, the alternative terminology of *Thiessen* or *Dirichlet tesselation* is used. This paper gives an algorithm for computing the Voronoi diagram of a set of planar objects under the Euclidean metric. When restricted to the interior of a simple polygon, this diagram is known as the *medial axis* or *internal skeleton* of the polygon. Many variations of the problem studied here have been investigated. The following illustrates the range of possibilities:

(a) Using the general L_p-metric instead of the usual Euclidean metric [11], [14]. An unusual metric that arises in computational fluid dynamics [3] is the problem of computing the Voronoi diagram for a set of points in the plane under the following metric:

$$D(p, q) = \min\{d(p, q+r): r \in \mathbf{Z}^2\},$$

* This work was supported by NSF Grants No. DCR-84-01898 and No. DCR-84-01633.

where $\mathbf{Z}$ are the integers, and $d(p, q)$ is the Euclidean distance between points p and q. We may interpret this as computing the Voronoi diagram on the torus.[1] This problem can be linearly reduced to the standard Voronoi diagram for a set of points.

(b) Power diagrams or Laguerre geometry [8]. Another direction is to assign additive or multiplicative weights to points [1].

(c) Generalization to higher-dimensional spaces. An algorithm for the Voronoi diagram of point sets in higher dimensions follows (by a certain transformation) from Seidel's work on the convex hull [23]. An unusual space is $\mathbf{R}^2 \times \mathbf{S}^1$ where $\mathbf{R}$ is the real line and $\mathbf{S}^1$ the unit circle: the Voronoi diagram here is used for planning the motion of a line segment [19]-[21].

(d) Voronoi diagrams arising from convex distance function [4], [6], [15].

We note here three recent applications of Voronoi diagrams for a set of straight and circular segments, all arising in robotics:

(i) In [18] we show that planning the motion of a disk amidst polygonal objects can be reduced to searching in the Voronoi diagram of these objects. This result clearly extends to the case where the obstacles are bounded by line segments or circular arcs. (As an example of the use of circular arcs, the mobile robot in [17] approximates itself and the obstacles by disks; an $O(n^4)$ time algorithm was implemented there.) The importance of the case of a disk arises from the algorithm's efficiency compared with the best algorithm for even slightly more complicated shapes (e.g., [19]). Indeed, computing the Voronoi diagram can be regarded as a preprocessing cost in which case the actual path planning for a disk becomes a linear-time process. Thus an algorithm for moving a disk can serve as an important initial heuristic in general motion-planning algorithms. But until the availability of an easily implementable $O(n \log n)$ algorithm for computing the Voronoi diagram, this importance remains mostly theoretical.

(ii) Sharir [25] discusses the problem of detecting if any two differently colored circles from a set of colored circles intersect each other. This problem can be easily solved if we have the Voronoi diagram of these circles.

(iii) Baker *et al.* [2] show how to find all stable three-fingered grasping positions of a nonconvex polygon using the Voronoi diagram of a shape composed of straight lines and circular arcs.

Previous Work. Before the advent of computational geometry, a number of algorithms for various cases of the problem considered in this paper were proposed (mostly running in time $\Omega(n^2)$). Here we review recent results that are asymptotically efficient and which rely on the techniques of computational geometry.

[1] The points on the torus represent (moving) markers for solving Navier-Stokes equations.

(1) The first algorithm in this genre is due to Shamos and Hoey [24] who gave an $O(n \log n)$ algorithm for the Voronoi diagram for a set of points.
(2) In the thesis of Drysdale [5], the problem of the Voronoi diagram for a set of disjoint polygonal and circular objects was first studied. He described and implemented an $O(nc^{\sqrt{\log n}})$ algorithm. This bound is subquadratic but, since it is $\Omega(n \log^k n)$ for any k, Drysdale also posed the problem (solved in this paper) of an $O(n \log n)$ solution.
(3) Subsequently, Drysdale and Lee improved the bound in (2) to $O(n \log^2 n)$ [13].
(4) At the 1979 Symposium on Foundations of Computer Science, Kirkpatrick [9] outlined in $O(n \log n)$ solution. But the technique is complicated enough that the correctness of some of its details remains to be settled [10]. However, Kirkpatrick's ideas ("spokes" and the use of minimum spanning tree) have independent interest.
(5) When restricted to the problem of computing the medial axis of a simple polygon, Lee [12] presented an $O(n \log n)$ algorithm (improving an earlier one in [22]).
(6) Sharir [25] describes an $O(n \log^2 n)$ algorithm for n circles that may intersec. Note that this improves (c) for the case of circles since the solution in (c) assumes that the circles are disjoint.

We refer to a recent review [16] of most of the preceding results as well as the techniques and generalizations known. Recently, Fortune [7] discovered a very different $O(n \log n)$ algorithm for the problems considered in this paper. His elegant method is based on plane-sweep in a transformed space; the fact that an $O(n \log n)$ plane-sweep method exists is, in itself, a pleasant surprise. (Our own results were obtained in the fall of 1984 [26]).

Discussion: Separability Condition. All the above algorithms (except for Fortune's) use the divide-and-conquer paradigm: let X be a set of line segments or circular arcs. To compute the Voronoi diagram of X, divide X into equal subsets X_L and X_R, recursively compute their Voronoi diagrams, and then "merge" the results. The merging is essentially defined by a certain "merge curve" C (intuitively, to one side of C, the Voronoi diagram of X comes from the Voronoi diagram of X_L while to the other side, it comes from X_R). To obtain an $O(n \log n)$ algorithm, the goal is to compute C in linear time. In the Shamos–Hoey algorithm for points, the merge curve is a simple connected infinite curve separating X_L and X_R (we view this as a kind of "separability" property of the two sets X_L and X_R). The work of Drysdale and Lee attempts to recover this separability property when the input is a set of line segments. As they reported, finding a computationally simple separability property remained elusive despite considerable effort. Accepting the fact that the separability property is not easily achieved, C may have many connected components and the issue is now to find in linear time at least one point (called a "starter") in each component of C. From each starter, we can trace out a component of C. The innovation of Kirkpatrick is to show that, to compute the Voronoi diagram of a set of points, no notion of

separability is needed (i.e., X_L and X_R can be arbitrary). His idea is to subdivide each Voronoi cell (by introducing "spokes") into simpler subcells, and to use the fact that a certain minimum spanning tree of X intersects the Voronoi edges and spokes of X_L and X_R in a fashion that allows one to find all the starters in linear time. This idea appears again in Sharir's work on intersecting circles. In some sense, the new idea in this paper is to reintroduce the separability condition in a radical way ("by simply imposing it").

Underlying Technique of This Paper. We now give our basic idea. Recall from the abstract of this paper that we solve the following:

> Given a set X consisting of n straight or circular segments (possibly degenerated to points), where the segments do not intersect except at their endpoints, compute their Voronoi diagram $\mathrm{Vor}(X)$.

It will be shown in the next section that $\mathrm{Vor}(X)$ is composed of straight, parabolic, hyperbolic, or elliptic curves. Since all and only conics can appear in the diagram, our problem is a very natural level of generalization of the original problem for points. (In other words, if we do not wish to handle curves of degree more than 2, then our problem is the most general case to consider.) An $O(n \log n)$ solution to this problem would subsume the above-mentioned works (1)–(5) since polygons and circles can be decomposed into circular and straight segments. However, it would not subsume (6) since, there, the n circles may intersect arbitrarily (giving rise to $\Omega(n^2)$ circular arcs in the worst case).

If there are m distinct endpoints among the segments of X, we introduce $m+1$ vertical lines such that each endpoint lies between a unique pair of adjacent vertical lines. The region between any pair of (not necessarily adjacent) vertical lines is called a *slab*. In stage 0, for each slab between a pair of adjacent vertical lines, we compute the Voronoi diagram of the *restriction* of the segments of X to the slab. In stage 1 we "merge" pairs of Voronoi diagrams of adjacent slabs computed in stage 0. In general, at stage $i+1$ we merge pairs of adjacent slabs from stage i. In $\log m$ stages, we would have computed the Voronoi diagram of X. Note that in stage i we compute the Voronoi diagram of slabs that contains 2^i endpoints of X. The obvious implementation of this algorithm may take $\Omega(n^2)$ time, simply because in the initial stages, merging each pair of slabs can take up to linear time.

We overcome this problem by computing only the "essential" part of the Voronoi diagram of a slab, where, roughly speaking, this essential part has size only $O(k)$ if the slab contains k endpoints of X. Furthermore, merging two slabs that collectively contain k endpoints takes only $O(k)$ work. This implies that each stage takes linear time and our stated time bound follows.

In the rest of this paper we present the basic definitions in Section 2. Section 3 describes some simple properties of the process of moving along the Voronoi diagram of objects. Section 4 gives the merging process which is the heart of the algorithm. Sections 5 and 6, respectively, prove the termination and the correctness

of the merge process. After the overall algorithm is given in Section 7, we analyze its complexity in Section 8. Some concluding remarks are given in Section 9.

2. Preliminaries

Following [9] we take our primitive objects to be points, open line segments, and open circular arcs. It is important to remark that this "expedient" of Kirkpatrick is actually a crucial insight that solves several technical problems faced when trying to generalize the original definition of Voronoi diagrams for points: see [5] for a discussion of the issues. For simplicity we restrict the line segments to be finite and the circular arcs to be less than a semicircle. Motivated by robotics applications the open line segments and arcs will be called *walls* and the points will be called *corners*. An *object* is either a wall or a corner. (Note: depending on the context, a corner is treated as a point or a singleton set.) A set X of objects is said to be *proper* if (a) they are pairwise disjoint and (b) for each wall in X its endpoints are corners in X. Note that we allow isolated corners in a proper set. Hereafter, let X denote a proper set of objects. Assumption (b) in the definition of "properness" is a technical convenience ensuring that each Voronoi edge is part of a unique conic (see below).

Some of the following definitions are fairly standard: the *projection* of a point p onto an object s is the point q in the closure $\bar{s}$ of s such that the Euclidean distance $d(p, q)$ is minimized [13]. Note that this term is well defined except in this case: suppose s is an open arc of a circle centered at O and the line L, defined to be the bisector of the segment joining the two endpoints of s. Let $H \subseteq L$ be the closed half-line bounded by O and which does not intersect s. Then the projection of p onto s is undefined precisely when p lies in H. In our applications, whenever we project p onto s then p will in fact be as close to s as to any other objects (including the endpoints of s). It will be shown in Lemmas 1 and 2 below that, under this condition, p does not lie in H. Henceforth, we only use projections when they are well defined.

The *distance* $d(p, s)$ between p and s is defined as $\inf\{d(p, q): q \in s\}$. Define the clearance of an arbitrary point p with respect to X to be the minimum of $d(p, s)$ where $s \in X$. Denote this by $Clearance_X(p)$ or simply $Clearance(p)$. In our proofs it is often convenient to refer to the circle centered at p with radius equal to $Clearance_X(p)$: call it the *clearance circle* at p (with respect to X). The closed region inside the clearance circle is called the *clearance disk* at p.

We consider two ways to define the Voronoi diagram. If we regard X as a set of points obtained as the union of the objects in X then we have a very simple "intrinsic" definition as in [18] or [20]. Precisely, if $\bigcup X$ is the set of points in objects of X, then the *intrinsic Voronoi diagram* of X is the set of points p with positive clearance such that the intersection of the clearance circle at p with the closure of $\bigcup X$ is a disconnected set.[2] The definition of the intrinsic diagram

[2] This elegant definition of Voronoi diagrams is one simple way to overcome the previously mentioned difficulties faced when defining Voronoi diagrams, as discussed in Drysdale's thesis.

serves to motivate the next definition of the Voronoi diagram, which is similar in spirit to [9]. Say a point p is $*$-*close* to an object s in X if for all $\varepsilon > 0$ there is a point q in the ε-neighborhood of p such that (i) $Clearance(q) = d(q, s)$ and (ii) the projection of q onto s is actually in s (rather than in $\bar{s} - s$). We can characterize $*$-closeness in this more explicit form: p is $*$-close to s iff

(i) $Clearance(p) = d(p, s)$ and
(ii) if the projection of p onto s is not in s then s is a wall with p on the normal through an endpoint of s.

Definition 1. The *Voronoi diagram* $\mathrm{Vor}(X)$ of X is the set of points p such that there exist at least two objects s_1 and s_2 that are $*$-close to p.

Example 1. Let X consist of the walls s_0, t_0 and the corners s_i, t_i $(i = 1, 2)$, as in Fig. 1. $\mathrm{Vor}(X)$ consists of all the points on the dashed curves. The intrinsic diagram of X consists of all the points of $\mathrm{Vor}(X)$ minus those that lie on the straight line segments perpendicularly through each of the corners. Thus it is the curve $p_0 p_1 \ldots p_8$ in Fig. 1.

By the above characterization of $*$-closeness, and from the example, it is intuitively seen that $\mathrm{Vor}(X)$ is simply the intrinsic Voronoi diagram of X augmented with additional line segments lying along the normals to the endpoints of walls. If X is not proper then $\mathrm{Vor}(X)$ is not necessarily decomposable into a collection of curves and this can be problematic. For computational purposes, and for most of this paper, we prefer to use $\mathrm{Vor}(X)$; but we will have occasion to consider the intrinsic Voronoi diagram.

For any pair of objects s, s', the (s, s')-*bisector* is the Voronoi diagram of the set $\{s, s'\}$. If s and s' are objects from X then the properness of X implies that the bisector is a simple curve that divides the plane into two infinite regions. But, in general, the bisector may contain branch points as illustrated by the following: if s and s' are two straight walls such that one endpoint q of s is in the relative interior of s' then the bisector consists of three branches emanating from q. (This situation is excluded by the properness of X since q would have to be an object in X but q and s' are not disjoint.)

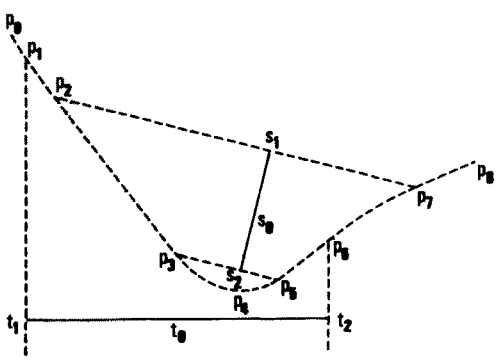

Fig. 1. Voronoi diagram.

Let us briefly note the types of bisectors. In case the objects are corners and straight walls, the bisector is familiar from previous work:

[CC] If s and s' are both corners then the bisector is a line.

[CW] If s' is a straight wall and s is its endpoint then the bisector is the line through s and normal to s'.

[WW] If they are disjoint straight walls (except for a common endpoint) then the bisector is a curve that may be composed of up to seven sections of straight or parabolic lines. This is illustrated in Fig. 1: the bisector of the walls s_0 and t_0 is identical to the intrinsic diagram of the points in the figure.

Next we illustrate the basic types of interactions involving circular arcs. To do this, we consider infinite lines and full circles instead of line segments and arcs. We allow these circles and lines to intersect freely here, and we will use the intrinsic Voronoi diagram. (It is shown later that edges of $\mathrm{Vor}(X)$, except for those normal through an endpoint of a wall, are portions of such intrinsic diagrams.) It is then easy to verify the following:

(1) *Two nonintersecting circles, each external to the other*: the bisector is one branch of a hyperbola (unless the two radii are equal, in which we have a straight line).

(2) *Two intersecting circles*: the bisector is the union of an ellipse and one branch of a hyperbola, both passing through the two intersecting points of the circles (see Fig. 2). Degeneracy occurs when the two circles touch. There are two cases: the circles touch externally to each other and the circles touch with one contained in the other. In the first case, the ellipse becomes a line segment joining the two centers. In the second case, the hyperbola branch becomes a ray from the common point and directed away from both centers.

(3) *Two nonintersecting circles, one contained in the other*: the bisector is an ellipse with foci the two centers of the circles. The ellipse separates the two circles.

(4) *A circle of radius r and a nonintersecting line* Λ: a parabola with focus the center of the circle and directrix a line Λ' parallel to Λ and at distance r from Λ. The line Λ lies between Λ' and the circle.

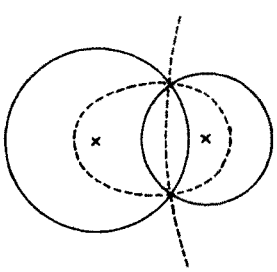

Fig. 2. Bisector of two intersecting circles.

(5) *A circle of radius r and an intersecting line* Λ: two parabolas both passing through the two intersection points of Λ and the circle. The directrices of the two parabolas are the two lines parallel to Λ and at a distance r from Λ. In the degenerate case, where the line Λ is tangent to the circle, one of the parabola becomes a ray emanating from the point where Λ touches the circle.
(6) *A circle and a point outside the circle*: a branch of hyperbola.
(7) *A circle and a point inside the circle*: an ellipse inside the circle. (Note: in the analysis, we can essentially treat cases (6) and (7) as degeneracies of (1) and (3), respectively.)

Remark. The reader familiar with [25] will note that our definition of Voronoi diagrams (when restricted to full circles) differs from Sharir's (which is the same as in [13]). Sharir works with circles and defines the distance $D(p, C)$ from a point p to a circle C of radius r centered at q as $d(p, q) - r$. So distance could be negative and the diagram for a set of circles defined by Sharir has no elliptic curves. But it is easy to see that by removing all the elliptic curves in our diagram, we obtain Sharir's. Negative distances have the disadvantage that they do not generalize easily to circular arcs.

The Voronoi diagram can be decomposed into *Voronoi edges* where each edge e is a maximal connected portion of the (s, s')-bisector determined by some pair of objects s, s'; e is called an (s, s')-edge.

Example 2. Because X is proper, each Voronoi edge is a segment of a unique conic (rather than a union of such segments). To see this, consider the set X of objects in Fig. 1. There are 15 Voronoi edges, of which eight are normals emanating from corners. Consider the Voronoi diagram of $X' = \{s_0, t_0\}$, i.e., the improper set of objects obtained by omitting the corners in X. Then $\mathrm{Vor}(X')$ consists of the curve $p_0 p_1 \cdots p_8$ (thus, $\mathrm{Vor}(X')$ coincides with the intrinsic diagram of X). The entire curve makes up one Voronoi edge, and is clearly not part of a unique conic.

The endpoints of Voronoi edges are called *Voronoi vertices.* The set of points in the plane that are neither on the $\mathrm{Vor}(X)$ nor in any object of X is partitioned into connected components called *Voronoi cells.* For example, there are eight cells in Fig. 1. Each cell is associated with an object s where for all points p in the cell, p is $*$-close to s. Conversely, a corner (resp. wall) is associated with at most one (resp. exactly two) cells. A cell associated with an object s is called an *s-cell.*

We now show that the Voronoi diagram for a set of straight and/or circular segments are portions of the curves (1)–(7) described above. To indicate why this is not immediately obvious, suppose e is an (s, s')-edge where s is a circular arc of a circle C_s. Let p be a point in e and assume that $q \in s$ such that $d(p, q) = d(p, s)$. It is conceivable that the line segment $[p, q]$ properly contains a radius of the circle C_s. If this were so, then none of the bisectors (1)–(7) fits this description and e cannot be a portion of these curves. Another possibility

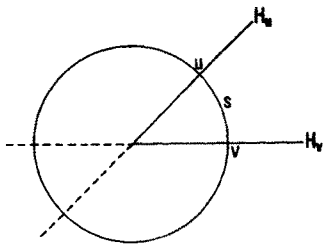

Fig. 3. Cone of Influence of s.

that does not fit (1)-(7) is when the line through p and q does not pass through the center of C_s. We will show that such possibilities do not arise. First, we state a simple geometric lemma whose proof is left to the reader.

Lemma 1. *Let s be a circular object and C_s be the circle containing s. If the endpoints of s are u and v, then let K_s be the cone bounded by the pair of rays H_u, H_v emanating from the center of C_s and passing through u and v, respectively. Then for any point p, p is not in the interior of K_s if and only if the projection of p onto s is either u or v (i.e., $d(p, s) = \min\{d(p, u), d(p, v)\}$).*

We call K_s the "cone of influence" of s (Fig. 3). We generalize this to any object s: if s is a corner then its cone of influence is the whole plane, and if s is a straight line segment, its cone of influence is the strip bounded by the pair of parallel lines normal through its endpoints. The line segment joining any point p on the Voronoi diagram to its projection q on any $*$-close object s is inside the cone of influence of s, and in fact these line segments must be normal to s.

Lemma 2. *If e is an (s, s')-edge in $\mathrm{Vor}(X)$ then e lies in $K_s \cap K_{s'}$ where K_s and $K_{s'}$ are the respective cones of influence.*

Proof. By symmetry, it is sufficient to show that e lies in K_s. If s is a circular arc, then the above lemma shows that e lies in K_s (otherwise if $p \in e$ lies outside K_s then p would be closer to an endpoint of s, contradiction). There is nothing to show if s is a point. The case where s is a straight line segment is easy. □

From this lemma, it follows that the types of Voronoi edges have been exhaustively enumerated by [CC], [CW], [WW], and (1)-(7) above. To see this, if e is an (s, s')-edge then we just have to consider the types of cones of influence for s and s', and then observe that the corresponding Voronoi edge matches one of the cases enumerated. When s is a circular wall then K_s is divided into two parts by s that must be treated differently. We omit the details. Now we prove a basic result showing that the usual linear size of Voronoi diagrams remains true in our setting.

Lemma 3. *Given a proper collection X of n objects, the number of edges in $\mathrm{Vor}(X)$ is at most $3n-6$.*

Proof. We construct an embedded graph G^* in the plane with n vertices such that embedded edges corresponds to Voronoi edges in $\mathrm{Vor}(X)$. For each object $s \in X$, we choose an arbitrary representative point $r(s) \in s$; likewise for each edge e in $\mathrm{Vor}(X)$, we choose a representative point $r(e) \in e$ (in the relative interior of s). (So if s is a corner, $r(s) = s$.) If e is an (s, s)-edge, let $p(e, s)$ and $p(e, s')$ be the projections of $r(e)$ onto s and s', respectively. (Note that $r(e)$ avoids the singularities where the projections are not well defined.) Now the embedded graph G^* has vertex set $V = \{r(s): s \in X\}$ and edge set E consisting of polygonal paths of the form

$$[r(s), p(e, s), r(e), p(e, s'), r(s')],$$

where e is an (s, s')-edge. The embedded edges to no intersect except for sharing portions of objects (to see this, notice that the property holds in each Voronoi cell). By a simple perturbation, we can ensure that these embedded edges do not intersect except at their endpoints: simply replace each point $p(e, s)$ by a point between $r(e)$ and $p(e, s)$, sufficiently close to $p(e, s)$. Hence G^* has n vertices and at most $3n - 6$ edges. □

The remaining definitions in this section have been invented for the technique in this paper. Let $m \leq n$ be the number of corners in X. As in Section 1, let us introduce $m + 1$ vertical lines called *separators* such that each corner is between a unique pair of adjacent separators. The region between any two (not necessarily adjacent) separators is called a *slab. We will assume that a circular wall has the property that an arbitrary vertical line intersects it at most once, and that no two corners are covertical.*

Let S be a slab. A wall is *long* with respect to S if it intersects both of the separators that bound S. The set of long walls of S partition the slab into closed regions that we call *quads* that *belong* to S. Quads are so-called because these have four sides when they are bounded regions. Thus if a slab has no long walls then the whole slab is the quad; otherwise, all but two of the quads are bounded. A quad is said to be *active* if there is at least one corner in it; otherwise it is *inactive.* Let Λ be a separator. A *crossing* (or *s-crossing*) is the intersection of a separator with a wall s. If Λ has $k \geq 0$ crossings, then Λ is divded in $k + 1$ segments called *windows.* A window is *active* with respect to S if it is part of the boundary of an active quad belonging to S.

Example 3. In Fig. 4 we have a slab with four (not five!) quads of which the upper two quads are active. The second quad from the top has three windows on the left boundary and two on the right.

Let Q be a quad. Roughly speaking, a *Q-object* s is obtained by restricting some object $s' \in X$ to Q. More precisely, s is one of the following:

(i) A corner corresponding to a crossing at a vertical boundary of Q. Note that each long wall contributes exactly two corners of this type, and all other walls contribute at most one.

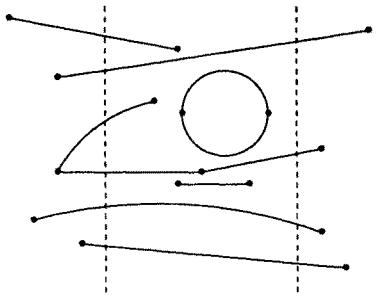

Fig. 4. A slab between two separators (the dotted lines).

(ii) An object of the form $s = s' \cap \text{int}(Q)$ where s' is an object of X and $\text{int}(Q)$ is the interior of Q.

(iii) An object of the form $s = s' \cap \text{int}(S)$ where s' is one of the long walls that bound Q, and Q belongs to the slab S. There are at most two Q-objects of this form.

Note that a Q-object s occurs in the original set X if and only if $\bar{s} \subseteq \text{int}(Q)$. Note that the set of Q-objects is proper. We also refer to a Q-object as a *Q-wall* or *Q-corner* as the case may be. For instance, if s' is a long wall that defines the upper or lower boundary of Q then s' contributes three Q-objects corresponding to the two s'-crossings and the wall $s' \cap \text{int}(S)$. The *Q-diagram*, denoted $\text{Vor}(Q)$ (by abuse of notation), is the Voronoi diagram of the Q-objects. It is important to realize that although the Q-objects are confined to Q, the Q-diagram is defined in the whole plane. Clearly, the notion of Q-objects and Q-diagrams can be extended to the case where Q is a union of several quads in a slab. If Q is quad (or a union of quads), we again abuse notation by writing $Clearance_Q(p)$ for the clearance of p with respect to the Q-objects.

In our algorithm, a slab is said to be *processed* when the Q-diagram is computed for each active quad Q belonging to the slab. We represent a Voronoi diagram as an embedded planar graph, i.e., a graph such that at each vertex the cyclic order of the incident edges are available and at each face of the embedding, the cyclic order of the bounding edges are available. For unbounded faces, we introduce fictitious edges at infinity.

3. Properties of Moving Along a Bisector

Let s and s' be objects. In our merge algorithm, we need to identify one of the two directions along the (s, s')-bisector e as being "clockwise" with respect to s. This is rather natural and can be made precise as follows:

Let p be a point in e. For our purposes we may assume that it is not the case that one of the objects s, s' is a wall and the other an incident corner; we may further assume that p is not at a transition between two segments of e corresponding to different governing equations. Then there is a well-defined tangent to e at

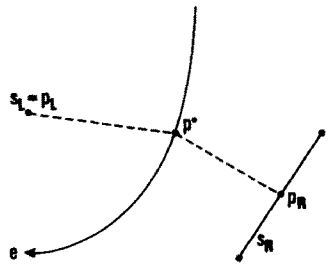

Fig. 5. Moving along a bisector.

p. Let p_s denote the projection of p onto s (so $p_s \in \bar{s}$). Let $\mathbf{u}$ be a tangent vector at p. Let $\mathbf{v}$ be the vector from p_s to p. It is easy to show that the tangent at p cannot pass through p_s, i.e., if we write $\mathbf{u}$ as $u_x\mathbf{i}+u_y\mathbf{j}+0\mathbf{k}$, then the $\mathbf{k}$-component of $\mathbf{u}\times\mathbf{v}$ is nonzero. We say $\mathbf{u}$ represents a *clockwise* (resp. *anticlockwise*) *direction about* s iff the $\mathbf{k}$-component of $\mathbf{u}\times\mathbf{v}$ is >0 (resp. <0).[3] By symmetry, it is easy to see that a direction along e is clockwise about s iff it is anticlockwise about s'.

We now state without proof some elementary properties. Let s_L and s_R be objects; e and e' be the (s_L, s_R)- and (s'_L, s_R)-bisectors, respectively. Let $p^*(t)$, for $t \geq 0$, be a parametrized curve, regarded as a moving point p^*. Initially, p^* is moving along e. Let $p_\beta(t)$ $(\beta = L, R)$ be the projection of $p^*(t)$ onto s_β, and again we regard it as a moving point p_β.

(1) See Fig. 5. Suppose that p^* moves along e in the direction that is clockwise about s_L. If s_R is a wall then the motion of p_R (reflecting the motion of p^*) either is stationary at an endpoint of s_R or is continuous and unidirectional along s_R. If p_R is stationary then the vector from p_R $(=s_R)$ to p^* is turning continuously anticlockwise about p_R.

(2) Let q be a point in $e \cap e'$ such that the moving point p^* meets q and subsequently moves along e' in the direction that is clockwise about s'_L. We claim p^* made a *left turn* at q. More precisely, if $\mathbf{u}$ (resp. $\mathbf{v}$) is the tangent to e (resp. e') at q in the direction of motion of p^* then the $\mathbf{k}$-component of $\mathbf{u}\times\mathbf{v}$ is ≥ 0.

(3) Let Λ be a separator such that s_L lies in the open half-plane to the left of Λ, and s_R lies in the closed half-plane to the right of Λ. Then there is a point on the (s_L, s_R)-bisector e beyond which the clockwise motion of p^* about s_L has a positive component in the downward (vertical) direction.

(4) The *front arc* of p^* is defined to be the arc of the clearance circle at p^* obtained by traversing the circle anticlockwise from p_L to p_R. The *back arc* of p^* is defined analogously, as the complement of the front arc with respect to the clearance circle. As p^* moves along e, this front arc is sweeping monotonically forward in a sense made precise in the following lemma:

[3] Assuming a right-hand coordinate system.

Lemma 4. *Let $s_L, s_R \in X$ and e be an (s_L, s_R)-edge in* Vor(X). *If q_0 and q_1 are the Voronoi vertices bounding e such that moving along e from q_0 and q_1 corresponds to clockwise about s_L then the clearance circle at q_0 (with respect to X) can only touch objects of X on the back arc. Similarly, the clearance circle at q_1 can only touch objects of X in the front arc. In all intermediate positions between q_0 and q_1, the clearance circle touches no other objects except s_L and s_R.*

The proof of this lemma uses a technique taken from [20]. We sketch the method here. It is now convenient to regard e as a parametrized curve $p^*(t)$, for real values t, with increasing t corresponding to the direction clockwise about s_L. We will define two closed planar sets Γ_t, Σ_t of points with the following properties:

(i) $\Gamma_t \cap \Sigma_t$ equals the clearance disk at $p^*(t)$,
(ii) the sets Γ_t and Σ_t are continuously parametrized (in the Hausdorff metric[4] on sets) by t, and
(iii) the Γ (resp. Σ) sets are monotonically *growing* (resp. *shrinking*). More precisely, for $t < u$ we have $\Gamma_t \subset \Gamma_u$ and $\Sigma_u \subset \Sigma_t$. In fact, the growth and shrinkage have the following stronger property: the front (resp. back) arc at $p^*(t)$ (resp. $p^*(u)$) lies in the interior of Γ_u (resp. Σ_t).

The preceding growth properties of Γ_t and Σ_t immediately imply the lemma. (In [20], one of the cases also has a third family of sets Δ_t, but this can be avoided.) We now describe the Γ and Σ sets. We describe these sets according to the type of conic that e conforms to. To avoid repetition, we only treat the cases of hyperbolas and ellipses: the cases where e is parabolic or straight are essentially treated in [20].

(a) *Hyperbolas.* If the bisector e is hyperbolic then both objects s_β ($\beta = L, R$) must be circular walls (possibly one of the s_β degenerated to a point). Let s_β be part of a circle C_β centered at q_β. So the hyperbola containing e is (part of) the (C_L, C_R)-bisector. We assume the centers are on the x-axis, with q_L left of q_R. There are two cases, depending on whether the two circles intersect. First suppose they do not intersect (see Fig. 6). Then they must lie external to each other. Consider the polygonal path $P(t)$ composed of the ray from q_L extending away from q_R, the ray from q_R extending in the other direction, the segments $[q_L, p^*(t)]$ and $[p^*(t), q_R]$. Clearly, $P(t)$ divides the plane into an upper and a lower part. Define Γ_t (resp. Σ_t) to be the union of the region above (resp. below) $P(t)$ with the clearance disk at $p^*(t)$. It is elementary to verify properties (i)–(iii) above. Now consider the case where C_L and C_R intersect at a pair of points r, r'. So the hyperbola passes through r and r'. Since s_L and s_R are assumed not to intersect, e must lie in one of the three sections of the hyperbola. If e

[4] The Hausdorff metric on closed subsets of a metric space Y is as follows: for any set $S \subseteq Y$, $\varepsilon > 0$, let S_ε be the union of the ε-balls about the points of S. Then the distance between two closed subsets $S, S' \subseteq Y$ is given by $d(S, S') = \inf\{\varepsilon \geq 0: S \subset S'_\varepsilon \text{ and } S' \subset S_\varepsilon\}$.

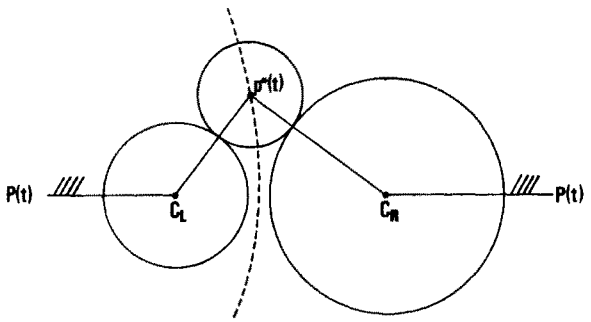

Fig. 6. The case of hyperbolas with nonintersecting circles.

were in the two infinite sections then the treatment is identical to the previous case. So assume e lies between r and r' inside the "lune" formed by intersecting the circles C_L and C_R. The front arc of $p^*(t)$ divides the lune into two parts: define Γ_t to be the upper part. Similarly, define Σ_t to be the lower part in the division of the lune by the back arc of $p^*(t)$. The reader can verify properties (i)-(iii).

(b) *Ellipses.* Again, e is elliptic implies that both objects s_β ($\beta = L, R$) are arcs of some circles C_β. Let K denote the strip bounded by the two horizontal lines through the centers of C_L and C_R. We allow the degenerate case when the two horizontal lines defining K coincide. Again we consider two cases depending on whether the circles intersect. First suppose they do not. Then one (say C_R) must lie inside the other and let J' be the interior of C_L minus the interior of C_R. So e lies inside J'. Note that, because of our assumption that a vertical line intersects an object at most once, the edge e must lie inside the strip K or entirely outside. Suppose e lies above K, since the cases where e lies below K or inside K are similarly treated. Let J be the part of J' above K. Roughly, J has a crescent shape. If D_t is the clearance circle at $p^*(t)$ then $J - D_t$ is divided into a left and a right part (one of them possibly empty). Define Γ_t to be the union of D_t and the right part. Similarly, define Σ_t to be the union of D_t and the left part. It is again easy to verify (i)-(iii) for the sets Γ_t and Σ_t. The final case, where the two circles intersect, also presents nothing new.

4. Merging

We now show how to process a slab S where S is the union of two slabs S_L and S_R separated by a separator Λ. By definition, this means we compute the Q-diagram of each active quad Q belonging to S. Note that $Q_\beta = Q \cap S_\beta$ ($\beta = L, R$) is a union of one or more quads belonging to S_β, none of which are necessarily active. The diagram of each quad Q' in Q_β is either already recursively computed (if Q' is active) or else the Q'-diagram is trivial and can be computed in constant time. Thus with $O(m)$ additional work, where m is the number of walls and

corners in $Q \cap S_\beta$, we can assume all Q'-diagrams are available. Similar to the previous well-known algorithms for Voronoi diagrams of curve segments, the Q-diagram is obtained by "merging" the set of Q'-diagrams for all Q' in Q_L and Q_R. We do this in two steps:

(1) (Vertical merge.) For $\beta = L, R$, form the Q_β-diagram by "merging" all the Q'-diagrams, for Q' in Q_β.
(2) (Horizontal merge.) Merge the Q_L- and Q_R-diagrams. This is the most important part of the algorithm.

The vertical merge is rather easy and depends on next lemma whose easy proof is omitted. Let Q_1, Q_2 be two adjacent quads belonging to the slab S_L, and let s_0 be the long wall that separates Q_1 from Q_2. So the intersection of s_0 with S_L gives rise to three objects s_1, s_2, s_3 that are simultaneously Q_1- and Q_2-objects. Without loss of generality let s_1 (resp. s_3) be the left (resp. right) endpoint of s_2.

Lemma 5. *If $C_i(s_1)$ $(i = 1, 2)$ denotes the s_1-cell in the Q_i-diagram then the horizontal ray extending leftward from s_1 is contained in $C_1(s_1) \cap C_2(s_1)$.*

Informally, this ray forms a natural boundary preventing interaction of the Q_1- and Q_2-objects. Using this lemma, it is easy to justify the following method for computing the Q_0-diagram where $Q_0 = Q_1 \cup Q_2$. For each Q_0-object s, the s-cells $C_0(s)$ in the Q_0-diagram is obtained as follows:

(1) Suppose $s = s_1$. The above lemma implies that $C_i(s_1)$ $(i = 1, 2)$ is unbounded. Assume Q_1 lies above Q_2. The boundary of $C_0(s)$ is obtained partly from the boundary of $C_1(s_1)$ starting from s_1 counterclockwise to its infinite edge, and partly from the boundary of $C_2(s_1)$ starting from s_1 clockwise to its infinite edge. Similarly for $s = s_3$.
(2) If $s = s_2$ then there are two s-cells: the $C_0(s)$ below (resp. above) s_2 is equal to the corresponding cell in the Q_2-diagram (resp. Q_1-diagram).
(3) If s is not one of the s_1, s_2, or s_3, then it is a Q_i-object for a unique $i = 1, 2$ and $C_0(s)$ is equal to the corresponding cell in the Q_i-diagram.

By repeated application of these observations, the Q_L-diagram can be obtained in $O(m)$ time. We can compute the Q_R-diagram similarly. This completes the vertical merge step.

The main part in constructing the Q-diagram is the merging of the Q_L- and Q_R-diagrams. This merging is defined by a certain "merge curve" which generally consists of several connected components. Indeed, there is a one-to-one correspondence between these components and the windows in $\Lambda \cap Q$. Therefore, we will call the component corresponding to window W the *W-contour*, and the procedure for computing the W-contour is called the *W-merge*. We mainly focus on the mechanism of the procedure at present, leaving the correctness proof to the next two sections.

Let W be fixed window in $Q \cap \Lambda$. Refer to Fig. 7. We will assume that W is finite; at the end we will handle the other cases. Let s_0 and s_1 be the Q-objects

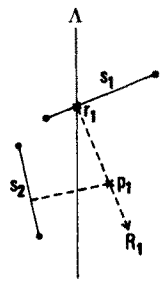

Fig. 7. Illustrating the start (s_2 is a wall).

whose crossings at Λ determine the lower and upper endpoints of W, respectively. Let r_i $(i = 0, 1)$ be the s_i-crossing. Consider the ray R_1 emanating from r_1 downward and normal to s_1. The initial part of R_1 is part of the boundary of the r_1-cell in the Q_L-diagram as well as the Q_R-diagram. Let r_1^β $(\beta = L, R)$ be the first vertex of the Q_β-diagram lying in R_1. Then define p_1 to be the closer of r_1^L or r_1^R to r_1. We call p_1 the *starter* for W. Note that the starter is well defined for a finite window because the ray R_1 has a downward component and must eventually get closer to r_0 than to r_1. Similarly, let R_0 be the ray emanating upward from r_0 and normal to s_0, and define the *ender* p_0 to be the first point along R_0 that is a vertex of the Q_L- or Q_R-diagram. The W-contour will begin at p_1 at the top and terminate at p_0.

Without loss of generality, let p_1 represent the intersection of the ray R_1 with the (s_1, s_2)-bisector where s_2 is a Q_L-object. It is possible that s_2 is the s_0-crossing r_0. In this degenerate case, the starter coincides with the ender and the W-contour is defined to be empty (we do nothing). Hence we may assume that this is not the case.

The procedure we now describe is a conceptual simplification of the Lee-Drysdale scan [13], which in turn is a modification of the Shamos-Hoey scan. To do this we follow Kirkpatrick's idea [9] of dividing each Voronoi cell into subcells by the systematic introduction of "spokes": for each Voronoi vertex v and for each point q in some object of X where $d(v, q) = Clearance_X(v)$, we call the line segment $[v, q]$ a *spoke* of X. The *augmented* Voronoi diagram of X is $\mathrm{Vor}(X)$ together with all such spokes. (Henceforth, $\mathrm{Vor}(X)$, Q-diagram, etc., refer to the augmented version.) Note that some spokes are already in $\mathrm{Vor}(X)$—these are precisely the Voronoi edges emanating normally from the endpoints of walls. The augmented Voronoi diagram is still a planar graph with at most $9n - 18$ edges. This is because each Voronoi vertex v with degree $k \geq 3$ in $\mathrm{Vor}(X)$ contributes at most k spokes. Let the connected planar regions that form the complement of the augmented Voronoi diagram be called *subcells*. Clearly, each Voronoi cell is now divided into a finite number of subcells. The most important computational property is that each subcell has at most four sides:

(a) If a bounded subcell has three sides then the subcell is incident to a corner c of X. The three sides consist of two spokes emanating radially from c and a portion of a conic connecting the free ends of the spokes. We call this a c-subcell.

(b) If the subcell has four sides then one side s of the subcell is a portion of a wall w, and the side s' opposite to s is a portion of a conic. Two spokes then connect s and s'. We call this is a w-subcell.

We describe the construction of the W-contour. The contour will be obtained as a sequence of conic segments

$$\sigma_1, \sigma_2, \ldots, \sigma_i, \ldots,$$

where each σ_i is bounded between two endpoints p_i and p_{i-1}. We call the p_i's the *breakpoints* of the W-contour. The segment σ_i is contained in the intersection of two subcells:

$$\sigma_i \subseteq C_i^{\mathrm{L}} \cap C_i^{\mathrm{R}},$$

where C_i^{β} is a subcell of the augmented Q_β-diagram. Furthermore, if C_i^{β} is an s_i^{β}-subcell then σ_i is a portion of the $(s_i^{\mathrm{L}}, s_i^{\mathrm{R}})$-bisector e_i. Here we observe the convention that moving "forward" along e_i from p_i to p_{i-1} corresponds to clockwise direction about s_i^{L}.

To initilize, σ_1 begins at the point p_1 which is the starter, and the construction halts when we reach the ender (we will prove later that this is inevitable). The subcells C_1^{L} and C_1^{R} are defined naturally. For example, if the starter p_1 is the intersection of the ray emanating from the s_1-crossing with the (s_1, s_2)-bisector where s_2 is a Q_{L}-object, then $s_1^{\mathrm{L}} = s_2$ and s_1^{R} is $s_1 \cap \mathrm{int}(S_{\mathrm{R}})$. Although s_1^{L}, s_1^{R} each has several subcells, it is easy to decide locally which should be used as C_1^{L}, C_1^{R}.

Inductively, suppose that σ_i, p_i, C_i^{L}, and C_i^{R} have been defined. We extend this to $i+1$. Now σ_i is a portion of the $(s_i^{\mathrm{L}}, s_i^{\mathrm{R}})$-bisector e_i. Moving along e_i from breakpoint p_i in the forward direction, suppose r_i^{β} is the first point on the boundary of C_i^{β} for $\beta = \mathrm{L}, \mathrm{R}$. We define the breakpoint p_{i+1} to be the first of r_i^{L} or r_i^{R} as we move from p_i forward along e_i. By symmetry, we may assume that p_{i+1} is r_i^{L}. There are two possibilities: (i) If p_{i+1} represents an intersection of e_i with a spoke (but not a segment emanating normally from endpoints of walls) then $s_{i+1}^{\mathrm{L}} = s_i^{\mathrm{L}}$ and $s_{i+1}^{\mathrm{R}} = s_i^{\mathrm{R}}$ (both unchanged), and the equation of the curve does not change. We have simply moved from one subcell of s_i to an adjacent one. (ii) If p_{i+1} represents an intersection with a Voronoi edge of the Q_{L}-diagram, then p_{i+1} is a Voronoi vertex of the Q-diagram. If we assume "general position," then there is a unique q_{L}-object s such that the clearance circle at p_{i+1} intersects s_i^{L}, s_i^{R}, and s. We then define s_{i+1}^{L} to be s and s_{i+1}^{R} to be s_i^{R}. The subcells C_{i+1}^{β} are naturally obtained. But, in general, the clearance circle at p_{i+1} (with respect to Q-objects) may touch more than three objects. By property (4) in the previous section, these objects only touch the clearance circle along the front arc (recall this is the arc from the contact point with s_i^{R} clockwise to the contact point with s_i^{L}). If we order these objects in the order of their contact with the front arc, starting with s_i^{R} and moving clockwise around the arc, we will encounter all the Q_{R}-objects before the Q_{L}-objects. Then s_{i+1}^{R} (resp. s_{i+1}^{L}) is defined to be the last (resp. first) of these Q_{R}-objects (resp. Q_{L}-objects). For later reference, we call this the *front-arc scan.*

This completes our description of the W-merge for the case of a finite window W. Note that as a consequence of going to subcells, we have removed (or trivialized) any need for analogues of the Hoey-Shamos or Lee-Drysdale scan in the literature. It remains to consider the two cases when W is infinite:

(a) Suppose W is a half-line. Without loss of generality, assume W extends infinitely upward from the s_0-crossing for some Q-object s_0. The ender p_0 for the W-contour can be defined exactly as above, except that it may not always be well defined (this happens if the ray from the s_0-crossing meets no Voronoi edges of $\mathrm{Vor}(Q_L)$ and $\mathrm{Vor}(Q_R)$). If the ender is undefined, then we say the W-contour is *empty* and we do nothing. Otherwise, we must now define the starter. Let H (resp. H_L, H_R) be the convex hull of the set of Q-objects (resp. Q_L-, Q_R-objects): the hull is composed of line segments and circular arcs. There are at most two choices for a pair p_L, p_R of points in H where the straight line segment $[p_L, p_R]$ is an edge of H that intersects Λ; this is because H as the boundary of a convex region intersects Λ at most twice. We let $[p_L, p_R]$ be the higher of the two (if any) choices, where "higher" is relative to where each segment intersects Λ.

Lemma 6. *Let $[p_L, p_R]$ be as above, and let $p_\beta \in s_\beta$ where s_β $(\beta = L, R)$ is a Q_β-object. Consider the infinite strip S of region bounded by $[p_L, p_R]$ and the two parallel rays emanating upward from p_L, p_R and normal to $[p_L, p_R]$. Let $Y \subseteq X$ consist of s_L and s_R together with those wall objects incident on s_L or s_R. Choose $D_1 > 0$ so that for each $s \in Y$ and any point $p \in S$ at distance greater than D_1 from $[p_L, p_R]$, the point q in $\bar{s}$ nearest to p is at p_L or p_R. Define $\varepsilon > 0$ to be the minimum distance of any object $s \in X - Y$ from the line through p_L, p_R. Choose $D_0 > D_1$ such that*

$$D_0 \varepsilon > d(p_L, p_R)^2.$$

Then $\mathrm{Vor}(X)$ confined to the strip S is equal to the (s_L, s_R)-bisector at all points further than distance D_0 from $[p_L, p_R]$.

Proof. We see that any point p in the strip S at distance $D > D_0$ from $[p_L, p_R]$ satisfies

$$d(p, s_\beta) - D < \varepsilon.$$

But for all object $s \in X - Y$, $d(p, s) - D \geq \varepsilon$. In particular, p is in the (s_L, s_R)-bisector if and only if p is in $\mathrm{Vor}(X)$. □

We define the starter to be the point on the (s_L, s_R)-bisector at distance D_0 from $[p_L, p_R]$.

The W-merge is done as follows: inductively assume the availability of the convex hulls H_L and H_R. We can compute H in time linear in the number of corners in Q: this is a simple modification of the standard algorithm of Hong and Preparata for the case of the convex hull of points. Indeed, this amounts to computing the points p_L and p_R from H_L and H_R. After checking that the ender is defined we can do the usual merge.

(b) Suppose W is the entire separator Λ. The W-contour is always defined in this case. We use the previous lemma to define a starter and ender. Note that this case is analogous to the "separable" situation that arises in the original algorithm of Hoey and Shamos.

This completes the description of the W-merge. In Section 6 we will describe how to use the W-contours obtained here to piece together the Q-diagram from the Q_L- and Q_R-diagrams.

5. Termination

The notations relative to a window W from the previous section are retained. We first consider the case where W is finite. It is now convenient to regard the W-contour as a parametrized curve $p^*(t)$, and p^* as a moving point. Let $q^*(t)$ be the horizontal projection of $p^*(t)$ to the separator Λ. We first prove the $q^*(t)$ is monotonic in t:

Lemma 7. *For all $t \neq t'$, $q^*(t) \neq q^*(t')$.*

Proof. For the sake of contradiction, suppose $q^*(t) = q^*(t')$. Without loss of generality, assume that $p^*(t) = p_L$ is strictly to the left of $p^*(t') = p_R$. Let C_β ($\beta = L, R$) be the clearance circle at p_β (with respect to Q-objects). Since C_β must touch both a Q_L- and a Q_R-object, it follows that Λ intersects C_β. It is impossible for any clearance disk centered at a point in the W-contour to be fully contained in another clearance disk. This implies C_L and C_R must intersect. Let Λ' be the vertical line through the two intersection points u, v of $C_L \cap C_R$. If Λ is strictly right of Λ' then note that C_L must touch some Q_R-object at a point in the interior of C_R, a contradiction. By another contradiction in the symmetrical case, we conclude Λ coincides with Λ'. But then since C_L must touch a Q_R-object, C_L must touch the Q_R-object at one of the two intersection points, say u. Since u lies in Λ, u must be either r_0 or r_1. If $u = r_i$ ($i = 0, 1$) then we see that the wall s_i must intersect the interior of C_L or C_R, contradiction. □

Corollary. *The horizontal projection $q^*(t)$ of the W-contour $p^*(t)$ into Λ is monotonic along Λ. In particular, the W-contour does not self-intersect.*

Proof. If the projection $q^*(t)$ is not monotonic then we contradict the previous lemma. □

Lemma 8. *Let the horizontal projections of the starter p_1 and ender p_0 on Λ be q_1 and q_0, respectively. Then q_1 lies above q_0.*

Proof. There is nothing to prove if $p_0 = p_1$. Otherwise, we show a contradiction by assuming that q_0 lies above q_1. By the corollary, q_0 lies strictly above q_1. By symmetry, let us assume $d(q_0, r_0) \geq d(q_1, r_1)$. Then the clearance circle at p_0 contains r_1 in its interior, contradiction. □

The preceding lemmas imply that, in tracing out the W-contour $p^*(t)$, the projection $q^*(t)$ is monotonically moving downward, from $q^*(t) = q_1$ until eventually $q^*(t) = q_0$.

Lemma 9. *When the horizontal projection of the W-contour reaches $q^*(t) = q_0$ then the W-contour reaches the ender, $p^*(t) = p_0$.*

Proof. Let C_0 be the clearance circle at p_0 and C^* be the clearance circle at $p^*(t)$ when $q^*(t) = q_0$. If $C_0 = C^*$ then the desired result holds. Otherwise, neither circle can be contained in the other and they must intersect in exactly two points. As in the proof of Lemma 7, let Λ' be the vertical line through the intersection points of C^* and C_0. A similar argument shows a contradiction (there are three cases depending on the relative positions of Λ' and Λ). □

The preceding three lemmas are not quite sufficient for concluding that our algorithm terminates: although it is true that the projection of the W-contour is monotonically descending, it logically possible to take infinitely many steps, hence not reaching the ender in finite time.

Recall that the W-contour is divided into curve segments $\sigma_1, \sigma_2, \ldots$ by breakpoints $p_1, p_2, \ldots$. The next two lemmas bound the number of breakpoints.

Lemma 10. *Let W be a window in $\Lambda \cap Q$ and let σ be any spoke in the Q_L- or Q_R-diagram.*

(a) *The W-contour is a portion of the Q-diagram.*
(b) *Each W-contour meets σ at most once.*
(c) *If W and W' are two windows in $Q \cap \Lambda$ then the W-contour and the W'-contour do not both intersect σ.*

Proof. (a) By construction, the clearance circle at points p of the W-contour touches at least two Q-objects, and thus p is in the Q-diagram.

(b) A spoke σ is a straight line segment emanating from a point p in some object s. Suppose the W-contour intersects σ at distinct points x and y. Then the clearance circles at x and y must both touch p. This implies that the clearance disk at y contains or is contained in the clearance disk at x. This is impossible for points x and y in the Q-diagram. The same argument applies in (c). □

Lemma 11. *If the number of Voronoi edges in the Q-diagrams (resp. Q_L- and Q_R-diagrams) is m_Q (resp. m_L and m_R) then the number of distinct breakpoints made by all the W-contours is $m_Q + m_L + m_R$ where W ranges over the windows in $Q \cap \Lambda$.*

Proof. The breakpoints corresponding to vertices of the Q-diagram is at most m_Q. The remaining breakpoints correspond to intersections with spokes in Q'-diagrams (Q' in Q_L or Q_R), and these number at most $m_L + m_R$. (This is because each Voronoi vertex of degree k has k spokes emanating from it, and the number

of spokes in the Q_L- or Q_R-diagram is at most the sum of the number of spokes in the Q'-diagrams, Q' in Q_L or Q_R.) By the previous lemma, the number of such breakpoints is bounded by $m_L + m_R$. □

Theorem 12 (Termination)

(a) *The W-merge algorithm terminates.*
(b) *If m is the number of Q-objects then the total number of steps for the W-merges is $O(m)$ when summed over all W in $Q \cap \Lambda$. Here a "step" corresponds to advancing from one breakpoint to the next.*
(c) *The total work done summed over all W-merges, W in $Q \cap \Lambda$, is $O(m)$.*

Proof. (a) Consider the case of a finite window W. The case of infinite windows is similar. By the previous lemma, the number of distinct breakpoints in a W-contour is bounded. Since the W-contour does not self-intersect, this number is also the number of steps in a W-merge. So the algorithm terminates.

(b) The $O(m)$ bound follows immediately from the preceding lemma, since $m_Q + m_L + m_R = O(m)$.

(c) There is only one subtle point here: it is not true that a single "step" takes constant time since, without assumptions of nondegeneracy, the clearance circle at a breakpoint may have an arbitrary number k of objects suddenly appearing on its front arc. In doing the "front-arc scan" we take $O(k)$ time. However, whenever the breakpoint has this property, we have "met" k Voronoi edges e of the Q_L- or Q_R-diagram. We can meet each Voronoi edge e at most twice this way. If the W-contour meet e more than twice, then the intersections are in the relative interior of e, and new Voronoi vertices are generated in the Q-diagram. We can thus charge the work done by the front-arc scan to either Voronoi vertices of the Q-diagram or to the Voronoi edges of the Q_L- and Q_R-diagram. The charge to each Voronoi vertex and to each Voronoi edge is constant. Therefore the $O(m)$ bound holds for both types of charges. □

6. Correctness

We will show how to construct the Q-diagram from the Q_L- and Q_R-diagrams, and the different W-contours. We first show that there is no interference between the W-contours for different windows W.

Lemma 13. *Let W and W' be two windows of $Q \cap \Lambda$. Assume W lies above W' and r_0 is the lower endpoint of W. The W- and W'-contours are disjoint. In fact, if Λ is the horizontal line through r_0 then the two contours lie on opposite sides of Λ.*

Proof. The W-contour projects horizontally onto Λ in a monotonic fashion. The desired result now follows since the ender for W is above Λ while the starter for W' is below Λ. □

If W is finite, define the *extended* W-contour to be the union of the W-contour with the two segments $[r_0, p_0]$ and $[r_1, p_1]$, where p_i are the starter and ender and r_i are the two crossings bounding W. Suppose W is infinite. If the W-contour is nonempty, the extended W-contour can be defined analogously. If the W-contour is empty, then W corresponds to a half-line determined by an s-crossing (for some Q-wall s). In this case, define the extended W-contour to be the half-line originating from the s-crossing in a direction normal to s and away from the Q-objects. Define $\mathscr{C}$ to be the union of all the extended W-contours where W ranges over those windows in Λ that intersect Q. Observe that $\mathscr{C}$ divides the plane into two infinite regions that can be naturally distinguished as the left and right sides of $\mathscr{C}$. This follows from the fact that the horizontal projection of $\mathscr{C}$ onto Λ is a bijection.

Lemma 14. *All the Q_R-objects lie to the right of $\mathscr{C}$ and, similarly, all the Q_L-objects lie to the left of $\mathscr{C}$. Only the objects corresponding to crossings at Λ lie on $\mathscr{C}$ itself.*

Proof. Let s be a Q_R-object to the left of $\mathscr{C}$, and assume that s does not lie in Λ. We will derive a contradiction. Pick any point q in s and let p be the (unique) point on $\mathscr{C}$ horizontally on to the right of q. Note that s and hence p is to the right of Λ.

(a) Suppose p is in the W-contour for some W. The clearance circle C at p (with respect to Q-objects) must touch a point q' in some Q_L-object. Hence the radius of C is at least the distance from p to Λ. Since the shortest line segment from p to Λ is the horizontal one, it follows that q lies on this shortest segment. So q is in the interior of C, contradiction.

(b) Suppose for some W, p is in the extended W-contour but not in the W-contour. The clearance circle at p touches a crossing r in Λ. This means the starter (similarly if it were the ender) for W lies beyond p in the ray R_1 emanating from r. But it is easy to see that the presence of q implies that the first vertex of $\mathrm{Vor}(Q_R)$ lies between r and p, contradiction. □

We come to the main result of this section:

Theorem 15 (Correctness). *The Q-diagram* $\mathrm{Vor}(Q)$ *is the union of all the W-contours (where W bounds Q) together with the portion of* $\mathrm{Vor}(Q_L)$ *to the left of $\mathscr{C}$ and the portion of* $\mathrm{Vor}(Q_R)$ *to the right of $\mathscr{C}$.*

Proof. For each point p in $\mathscr{C}$, it is easy to see that p is in some W-contour iff p is in $\mathrm{Vor}(Q)$. So assume p is strictly left of $\mathscr{C}$. The lemma then follows from the following claim:

$$\mathit{Clearance}_{Q_L}(p) < \mathit{Clearance}_{Q_R}(p).$$

Suppose the claim is false and for some Q_R-object s, $\mathit{Clearance}_{Q_L}(p) \geq d(p, s) = \mathit{Clearance}_{Q_R}(p)$. Let $q \in s$ such that $d(p, s) = d(p, q)$. Then $\mathscr{C}$ intersects the half-open segment $(p, q]$. Let r be any point in the intersection of $\mathscr{C}$ with $(p, q]$.

Let C_0 be the circle centered at p with radius $d(p, q)$ and let C_1 be the clearance circle at r (with respect to Q-objects). By assumption, the interior of C_0 does not intersect any Q_L- or Q_R-object. Clearly, C_1 is contained in C_0 (otherwise q would be in the interior of C_1) and also C_0 is a clearance circle of Q. Consider two cases:

(a) q is not a corner (i.e., crossing) in Λ. Then s is not a Q_L-object. Since C_1 must touch some Q_L-object s', it follows that the interior of C_0 intersects s', contradiction.

(b) q is a corner in Λ. Let q be the s'-crossing where s' is a Q-wall. But s' would have to be tangent to C_1 (and hence to C_0) at q. Thus s' is vertical, contradicting our assumption that no two vertices are covertical. □

Determination of Subcells. As a corollary, we see that the subcells of the Q-diagram can be obtained as follows. Fix any window $W \subseteq Q \cap \Lambda$:

(i) The new subcells created by the W-merge have the following form: if $p_i, \ldots, p_j$ $(i<j)$ form a contiguous sequence of breakpoints such that p_i and p_j are Voronoi vertices in the Q-diagram and $p_{i+1}, \ldots, p_{j-1}$ represent intersection with spokes, then the W-contour between p_i and p_j is a portion of the (s_i^L, s_i^R)-bisector. (Of course, $s_i^\beta = s_k^\beta$ for $k = i+1, \ldots, j$, $\beta = L, R$.) If the projection of p_i onto s_i^β $(\beta = L, R)$ is q_i^β then we obtain the new subcell D_i^β bounded by σ_i, the line segments $[p_i, q_i^\beta]$ and $[p_j, q_j^\beta]$, and the portion of s_i^β between q_i^β and q_j^β. Note that if $i<j-1$ then the formation of D_i^β is the result of merging two or more subcells in the Q_β-diagram. It is not hard to see how to create these new subcells on-the-fly during the W-merge.

(ii) For each crossing r at $Q \cap \Lambda$, discard the r-subcells in $\mathrm{Vor}(Q_L)$ and $\mathrm{Vor}(Q_R)$.

(iii) For each Q-wall s that intersects Λ, we form two new s-subcells. Note that there may be many s-subcells but we are interested in the two (one on each side of the wall) adjacent to the s-crossing $s \cap \Lambda$. Each new subcell is obtained by merging an appropriate subcell from $\mathrm{Vor}(Q_L)$ with a subcell from $\mathrm{Vor}(Q_R)$.

(iv) All other subcells of Q_L- and Q_R-diagrams unaffected by the preceding steps remain intact in the Q-diagram.

This procedure is correct because there is no interaction objects on different sides of the W-contour. For instance, D_i^L in step (i) is an s_i^L-subcell of the Q-diagram because of two properties: (a) since D_i^L is left of the curve $\mathscr{C}$, D_i^L is closer to Q_L-objects than to any Q_R-objects and (b) by construction, D_i^L is a subset of an s_i^L-cell in the Q_L-diagram.

7. Putting It Together

The main procedure consists of two preprocessing steps followed by a call to a recursive procedure:

Main Procedure
Input: a proper set X of objects (points, open line segments, open circular arcs).
Output: a representation of the augmented Voronoi diagram Vor(X).

(1) *Presort:* Sort the set of corners according to their vertical projection onto the x-axis. We introduce the set of separators in this step. Let *left*(s) and *right*(s) indicate the two separators adjacent to a corner s. (Note: to handle the case of more than one corner in any vertical line, only trivial modifications are necessary in our entire development.)

(2) *Prescan:* Do a scan-line sweep of the line segments to determine for each corner s the walls that are immediately (vertically) above and below s. This is essentially the algorithm of Hoey and Shamos for detecting line segment intersections. Use *above*(s) and *below*(s) to represent these walls. If s has no walls above or below it then this is given a special indicator. This step uses the information gathered in the presorting. Also at this step we determine at corner s the circular list of walls incident on s.

(3) *Recursion:* Call a recursive procedure to process the slab S bounded by the leftmost and rightmost separators. Note that the entire slab S constitutes an active quad, so the diagram of this quad is the desired Vor(X).

End Main Procedure.

We now present the recursive procedure for processing an arbitrary slab S (represented by a pair of separators). We assume that X, as well as the other information from the presort and prescan steps, is available to the procedure via global variables. The recursive procedure returns (i) a list of the active quads of S and two lists of windows for each quad where these lists are sorted in the natural top-to-bottom order; (ii) a list of the Voronoi diagrams of these quads; and (iii) the convex hull of the entire set of S-objects (i.e., the union of the Q-objects for all $Q \subseteq S$).

Recursive Procedure
Input: (m, S) where S is a slab and m the number of corners in S.
Output: The active quads, window lists, and convex hulls as described above.

(1) *Basis:* If $m = 1$, then use the prescanning information to determine the unique corner p in S and also the quad Q containing p. Compute the Q-diagram and return. Observe that if p has $k \geq 0$ incident walls then the Q-diagram takes time $O(k)$. (Note: the subdivision of walls into subobjects at their crossings is done at this point. This ensures that we do not introduce $\Omega(n^2)$ crossings.)

(2) *Divide:* Divide the slab S into two two slabs S_L and S_R with $\lceil m/2 \rceil$ and $\lfloor m/2 \rfloor$ vertices, respectively. (This is easy to do assuming that the separators for S are just indices into an array.) Recurse on $(\lceil m/2 \rceil, S_L)$ and $(\lfloor m/2 \rfloor, S_R)$. Let Λ denote the separator between S_L and S_R.

(3) *Conquer:*

(3.1) *Determination of active quads.* On return from recursion, we have the two lists of active quads belonging to S_L and S_R, respectively. It is

actually quite interesting to determine the list of active quads Q belonging to S. In order to preserve the continuity of the main description, we defer this to the end of the section. We just note here that for each active Q belonging to S we also determine (a) two lists $T_\beta(Q)$ ($\nu = \mathrm{L}, \mathrm{R}$) consisting of the quads (both active and inactive) belonging S_β whose union constitutes $Q_\beta = Q \cap S_\beta$, and (b) two lists of the windows on each side of Q. Recursively we have the diagrams of active quads in T_β, so here we only need to compute the diagrams of the inactive quads. Furthermore, if S has a total of m corners and has k edges incident on these corners then this substep takes time $O(k+m)$.

(3.2) *Vertical Merge.* For each list $T_\beta(Q)$ of quads, we do the "vertical" merge of their diagrams, resulting in the Q_β-diagram. The method is described in Section 4.

(3.3) *Horizontal Merge.* For each active Q belonging to S, we apply the W-merge to each window $W \subseteq \Lambda \cap Q$. Here, before doing the W-merge for the topmost and bottommost quads, we first compute the convex hull of the set of all Q-objects, using the recursively computed convex hulls of the Q_β-objects.

(3.4) *Determination of the subcells.* Finally, we construct the Q-diagram for each active Q. This is basically the computing of all the subcells of the Q-diagram described at the end of the previous section. Most of the work here may be done as part of the horizontal merge.

End of Recursive Procedure.

We are now done except for the details (step (3.1)) for determining the active quads of S. Let us introduce some terminology relative to any slab S and separator Λ: an *S-interval* I in Λ has the form $I = \Lambda \cap Q$ for some quad Q in S. If Q is active, then we call I an *active* S-interval. So an S-interval is a union of windows. (We will use this definition in the case where Λ is part of the boundary of S as well as when Λ is in the interior of S.) Two intervals in Λ *overlap* if their interiors intersect. Given a set $\mathcal{I}$ of intervals in Λ, the *$\mathcal{I}$-equivalence* relation on $\mathcal{I}$ is defined as the reflexive, symmetric, and transitive closure of the overlap relation on $\mathcal{I}$.

Lemma 16. *Let the slab S be divided into slabs S_L and S_R by the separator Λ. Let $\mathcal{I}_\beta$ be the set of active S_β-intervals in Λ. There is a bijective correspondence between the set of active S-intervals in Λ and the set of $(\mathcal{I}_\mathrm{L} \cup \mathcal{I}_\mathrm{R})$-equivalence classes such that if I is an active S-interval corresponding to an equivalence class then I is equal to the union of the intervals in that equivalence class.*

Proof. Suppose I is an active S-interval. The set

$$G_I = \{J \in \mathcal{I}_\mathrm{L} \cup I_\mathrm{R} : J \subseteq I\}$$

of S_L- and active S_R-intervals inside I must be nonempty. Note that an interval not in G_I cannot be $(\mathcal{I}_\mathrm{L} \cup \mathcal{I}_\mathrm{R})$-equivalent to any interval G_I. Hence G_I is a union

of equivalent classes. We must show that there is only one equivalent class in G_I. Let E_I be the set of endpoints of intervals in G_I. Sort E_I as $x_1, x_2, \ldots, x_k$ ($k \geq 2$) according to their height. If $k = 2$ then the claim is immediate, so let $k > 2$. Note that if x_i is not one of the endpoints of I (i.e., $1 < i < k$) then x_i cannot be the endpoint of both an S_L- and an S_R-interval. Suppose x_2 is the endpoint of some S_L-interval. Then we see that x_2 is in the interior of some active S_R-interval I_1: this is because x_2 represents an s-crossing for some Q-object s and one of the endpoints of s is in S_R, making the S_R-interval containing x_2 active. So let x_i ($i > 2$) be the lower endpoint of I_1 (the upper endpoint is seen to be x_1). If $i < k$ then a similar argument shows that x_i is in the interior of some active S_L-interval I_2. Furthermore, I_1 and I_2 overlap. Continuing this way, we eventually get to an interval I_m ($m \geq 2$) whose lower endpoint is x_k. Let $I_1, I_2, \ldots, I_m$ be the sequence of intervals so obtained. Since members of this list are equivalent and I is covered by these intervals, every $J \in G_I$ is $(\mathcal{I}_L \cup I_R)$-equivalent, as we wanted to show. □

Using this lemma, we can compute the list of the active S-intervals at each of the two separators bounding S: assume that inductively we have a list of the active S_β-intervals at each of the separators bounding S_β. We can now "merge" the list of S_L-intervals at Λ and the list of S_R-intervals at Λ. This is done in linear time in the standard way, only making sure that we record certain additional information. For instance, the crossing of a long Q-wall is detected as the coincidence of endpoints of intervals in both lists.

Once the list of active S-intervals is compiled, we can easily construct the list of active quads belonging to S. Similarly, we can obtain the corresponding list of active S-intervals at each of the two separators bounding S (we need to do this when we next merge S with an adjacent slab S').

The inactive quads belonging to S_β which will now be incorporated into the active quads of S can be determined at this point. These quads together with their diagrams are created at this point. If S has m corners and incident edges, then it is not hard to see that the overall work done is still $O(m)$. We conclude:

Lemma 17 *The work done in step* (3.1) *is linear in the number of active objects in the slab and the number of walls incident on these slabs.*

8. Complexity

We first analyze the complexity of the Recursive Procedure of the last section.

Lemma 18. *If a slab S contains m corners of the original input set X and the total number of edges incident on these m corners is k, then the nonrecursive part of the Recursive Procedure (steps* (3.1)-(3.4)) *takes $O(m+k)$ time.*

Proof. Step (3.1). We have already determined that this step takes $O(m+k)$ time.

Step (3.2). This is bounded by the number of subcells in the Q-diagrams over all active Q. This number is $O(m+k)$.

Step (3.3). In our termination proof, we showed that the work for tracing all the W-contours is bounded by the number of breakpoints encountered, and there is a linear number of these breakpoints. The other work done in this step is computing the convex hull of all the S-objects, assuming the convex hulls in S_L and S_R, respectively. This work is also linear by standard techniques.

Step (3.4) This is bounded by the number of cells that eventually appear in all the active Q-diagrams. □

Theorem 19. *The Voronoi diagram of a set X of n pairwise disjoint straight and circular arcs can be computed in time $O(n \log n)$.*

Proof. The presort and prescan steps of the Main Procedure takes $O(n \log n)$ time by standard methods. The Recursive Procedure also takes $O(n \log n)$ time because of the preceding lemma. □

9. Conclusion

This paper solves the open problem of an $O(n \log n)$ algorithm for computing the Voronoi diagram of a set of points, line segments, and circular arcs. The algorithm is simple enough that we think it can have an impact on practical applications such as in robotics.

The following simple observation is useful for implementing Voronoi diagram algorithms when the input has line segments only: although the Voronoi edges here consist of straight and parabolic segments, there is never a need to compute Voronoi vertices by intersecting pairs of parabolas. This is because every Voronoi vertex arises as the intersection of the pairwise bisectors among three (and possibly more) objects s, s', s''. Now note that either two objects are corners or two are walls. The bisector of two corners or two walls is a straight line.

The technique introduced in this paper seems to extend to more general algebraic curves, provided we take care to break up each curve into a number of suitably small sections. The technique may also be extendible (with additional ideas) to computing the Voronoi diagram of a set of polyhedral objects. This is a subject of further research.

Another direction which we have undertaken (jointly with Ó'Dúnlaing and Goodrich) is to parallelize our algorithm to run in $O(\log^2 n)$ steps using a linear number of processors. In contrast, it is not known at present whether the plane-sweep algorithm of Fortune can be parallelized to run in poly-logarithmic parallel time.

Acknowledgments

It is a pleasure to thank Sally Howe of the National Bureau of Standards for rekindling my interest in the problem. The clarity of this work has benefited

greatly from the comments of Micha Sharir, Denis Fortin, Neil Stewart, Colm Ó'Dúnlaing, and Boris Aronov. Denis and Neil at the University of Montreal have implemented a version of this algorithm.

References

1. F. Aurenhammer and H. Edelsbrunner, An optimal algorithm for constructing the weighted Voronoi diagram in the plane, *Pattern Recognition* **17** (1984), 251-257.
2. B. S. Baker, S. J. Fortune, and E. H. Grosse, Stable prehension with three fingers, *Proceedings of the 17th ACM Symposium on Theory of Computing*, 1984.
3. C. Borgers, C. Peskin, and O. Widlund, Private communication, 1984.
4. L. P. Chew and R. L. Drysdale, Voronoi diagrams based on convex distance functions, *Proceedings of the ACM Symposium on Computational Geometry*, 235-244, 1985.
5. R. L. Drysdale III, Generalized Voronoi Diagrams and Geometric Searching, Computer Science Technical Report STAN-CS-79-705, Stanford University, Stanford, California, 1979.
6. S. Fortune, A fast algorithm for polygon containment by translation, *Proceedings of the 12th International Colloquium on Automata, Language, and Programming*, 189-198, 1985.
7. S. Fortune, A sweepline algorithm for Voronoi diagrams, *Proceedings of the Second ACM Symposium on Computational Geometry*, 313-322, 1986.
8. H. Imai, M. Iri, and K. Murota, Voronoi diagram in the Laguerre geometry and its applications, *SIAM J. Comput.* **14** (1985), 93-105.
9. D. G. Kirkpatrick, Efficient computation of continuous skeletons, *Proceedings of the 14th IEEE Symposium on Foundations of Computer Science* 18-27, 1979.
10. D. G. Kirkpatrick, Private communication, 1984.
11. D. T. Lee, Two-dimensional Voronoi diagram in the L_p-metric, *J. Assoc. Comput. Mach.* **27** (1980), 604-618.
12. D. T. Lee, Medial axis transformation of a planar shape, *IEEE Trans. Pattern Anal. Mach. Intel.* **4** (1982), 363-369.
13. D. T. Lee and R. L. Drysdale III, Generalization of Voronoi diagrams in the plane, *SIAM J. Comput.* **10** (1981), 73-87.
14. D. T. Lee and C. K. Wong, Voronoi diagrams in $L_1(L_\infty)$ metrics with two-dimensional storagè applications, *SIAM J. Comput.* **9** (1980), 200-211.
15. D. Leven and M. Sharir, Planning a purely translational motion for a convex object in two-dimensional space using generalized Voronoi diagrams, *J. Discrete Comput. Geom.* **2** (1987), 9-31.
16. D. Leven and M. Sharir, Intersection problems and applications of Voronoi diagrams, in *Advances in Robotics*, Vol. 1 (J. Schwartz and C. K. Yap, eds.), Erlbaum, 1987.
17. H. P. Moravec, *Robot Rover Visual Navigation*, UMI Research Press, 1981. (Ph.D. thesis, Stanford.)
18. C. Ó'Dúnlaing and C. K. Yap, A retraction method for planning the motion of a disc, *J. Algorithms* **6** (1985) 104-111. Also in *Planning, Geometry, and Complexity* (J. Hopcroft, J. Schwartz and M. Sharir, eds.), Ablex, Norwood, NJ, 1987).
19. C. Ó'Dúnlaing, M. Sharir, and C. K. Yap, Retraction: a new approach to motion planning, *Proceedings of the 15th IEEE Symposium on Foundations of Computer Science*, 207-220, 1983. Also in *Planning, Geometry, and Complexity* (J. Hopcroft, J. Schwartz and M. Sharir, eds.), Ablex, Norwood, NJ, 1987).
20. C. Ó'Dúnlaing, M. Sharir, and C. K. Yap, Generalized Voronoi diagrams for moving a ladder: I. Topological analysis, *Comm. Pure Appl. Math.* **XXXIX** (1986), 423-483.
21. C. Ó'Dúnlaing, M. Sharir, and C. K. Yap, Generalized Voronoi diagrams for moving a ladder: II. Efficient construction of the diagram, *Algorithmica* **2** (1987), 27-59.
22. F. P. Preparata, The medial axis of a simple polygon, *Proceedings of the Sixth Symposiums on Mathematical Foundations of Computer Science*, 443-450, 1977.
23. R. Seidel, A Convex Hull Algorithm Optimal for Point Sets in Even Dimensions, M. Sc. thesis, University of British Columbia, 1981.

24. M. I. Shamos and D. Hoey, Closest point problems, *Proceedings of the 16th IEEE Symposium on Foundations of Computer Science*, 151-162, 1975.
25. M. Sharir, Intersection and closest-pair problems for a set of circular discs, *SIAM J. Comput.* **14** (1985), 448-468.
26. C. K. Yap, An $O(n \log n)$ Algorithm for the Voronoi Diagram of a Set of Simple Curve Segments, Robotics Laboratory Report No. 43, Courant Institute, New York University, New York, 1985.

Received December 29, 1986, and in revised form May 10, 1987.