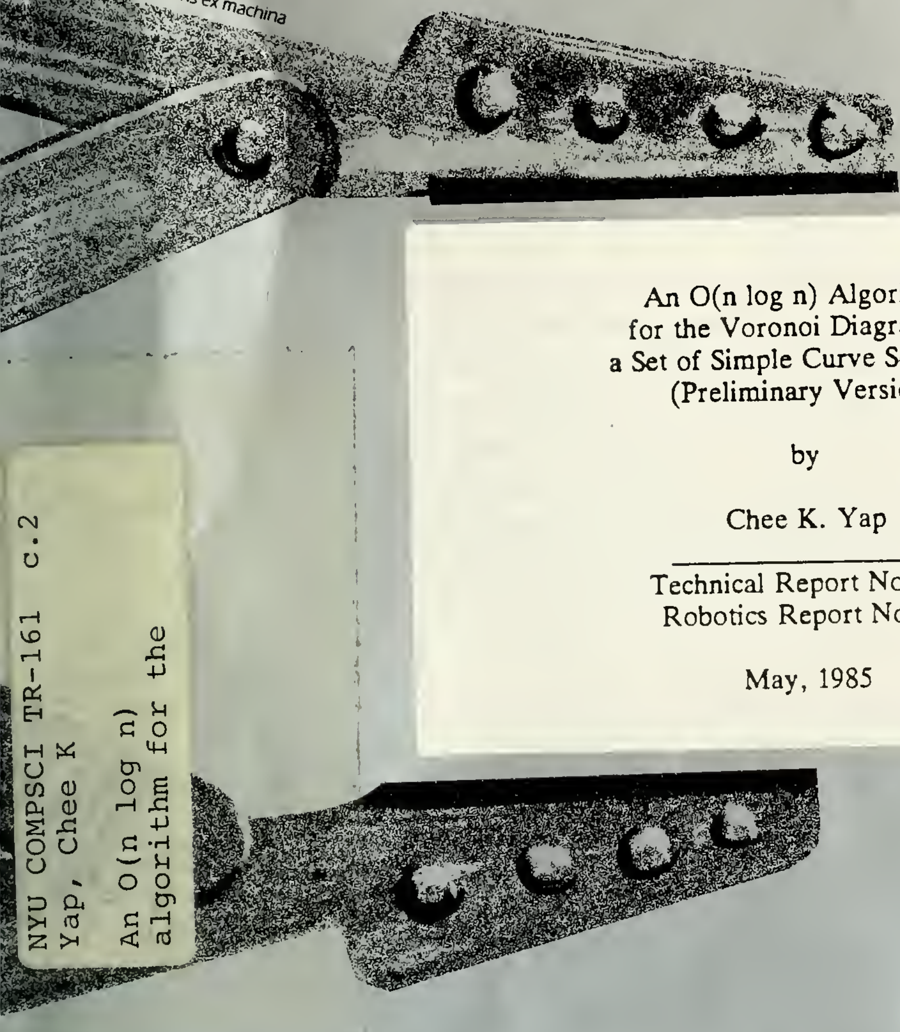
Robotics Research Technical Report

eneratorium omnis laboris ex machina



An O(n log n) Algorithm
for the Voronoi Diagram of
a Set of Simple Curve Segments
(Preliminary Version)

by

Chee K. Yap

Technical Report No. 161
Robotics Report No. 43

May, 1985

New York University
Courant Institute of Mathematical Sciences

Computer Science Division
251 Mercer Street New York, N.Y. 10012

An O(n log n) Algorithm
for the Voronoi Diagram of
a Set of Simple Curve Segments
(Preliminary Version)

by

Chee K. Yap

Technical Report No. 161
Robotics Report No. 43

May, 1985

New York University
Department of Computer Science
Courant Institute of Mathematical Sciences
251 Mercer Street
New York, N.Y. 10012

Work on this paper has been partially supported by Office of Naval Research Grant N00014-82-K-0381, and by National Science Foundation Grants DCR-83-20085, DCR-84-01898, DCR-84-01633, and by grants from the Digital Equipment Corporation, the IBM Corporation, the Sloan Foundation, and System Development Foundation.

(Preliminary Version)

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

ABSTRACT

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

October 19, 1984

This work is partially supported by NSF grants #DCR-84-01898 and #DCR-84-01633.

(Preliminary Version)

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

1. Introduction

The ubiquitous Voronoi diagram has been extensively 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 [Kir79],[Dry79], or [Lee82] for extensive references). In some literature, the alternative terminology of *Thiessen* or *Dirichlet tesselation* is more common. In this paper we are concerned with the Voronoi diagram for 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. To illustrate the range of possibilities, we mention the following:

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

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

where $\mathbf{Z}$ are the integers, and $d(p, q)$ the Euclidean distance. We may interprete 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) Assigning weights to points [AE83].

(c) Generalization to higher dimensional spaces. 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 ([OSY83],[OSY84a],[OSY84b]).

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

Our interest in the problem arose from robotics: in [OY82] we show that planning the motion of a disc amidst polygonal objects can be reduced to searching in the Voronoi diagram. 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 [Mor81] approximates itself and the obstacles by discs; an $O(n^4)$ time algorithm was implemented there.) The importance of the case of a disc is the algorithm's extreme efficiency compared to other even slightly more complicated shapes (eg. [OSY83]). Indeed, computing the Voronoi diagram can be regarded as a preprocessing cost in which case the actual path planning becomes a linear time process. Thus an algorithm for moving a disc 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 may remain mostly theoretical. The results of the present paper will hopefully remedy this situation. (See also [Sha83] for another robotics application of Voronoi diagrams: here we want to detect if any two differently colored circles intersect from a set of colored circles.)

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 will review those results that are asymptotically efficient and which rely on the techniques of computational geometry.

(a) The first algorithm in this genre is due to Shamos and Hoey [SH75] who gave an $O(n\log n)$ algorithm for the Voronoi diagram for a set of points. We might add that algorithms for the Voronoi diagram of point sets in higher dimensions follows indirectly from the work of Seidel on the convex hull [Sei81].

(b) In the thesis of Drysdale [Dry79], the problem of the Voronoi diagram for a set of disjoint polygonal and circular objects was 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.

(c) Subsequently, Drysdale and Lee improved the bound in (b) to $O(n\log^2 n)$ [LD81].

(d) At the 1979 FOCS, Kirkpatrick [Kir79] outlined an $O(n\log n)$ solution. But the technique is complicated enough that the details still remain somewhat unsettled ([Kir84],[Sha84]). Our techniques is simpler than Kirkpatrick's although his ideas ('spokes' and the use of minimum spanning tree) have independent interest.

(e) When restricted to the problem of computing the medial axis of a simple polygon, [Lee82] presented an $O(n\log n)$ algorithm (improving an earlier one in [Pre77]).

(f) Sharir [Sha83] describes an $O(n\log^2 n)$ algorithm for circles that may intersect. Note that this improves (c) for the case of circles since the solution in (c) assumes that the circles are disjoint. We remark that the connection between Sharir's problem to the setting assumed in this paper is not direct. At the end of the paper we will show how a variant of our technique improves his bound to $O(n\log n)$.

We refer to a recent review [SL85] of most of the preceding results as well as the techniques and generalizations known.

Technical discussion. All the above algorithms use the divide-and-conquer paradigm: 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 result. The merging is essentially defined by a certain 'merge curve' C. To obtain an $O(n\log n)$ algorithm, the goal is to compute C in linear time. In the Shamos-Hoey algorithm, the merge curve is a connected set (we can 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. As they reported, finding such a separability property that is computationally simple remained elusive despite considerable effort. Accepting the fact that C will have many connected components in general, the technical issue is to find at least one point (called a 'starter') in each component of C. The innovation of Kirkpatrick is to show that no notion of separability is needed (ie. 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 the starters. This idea appeared again in Sharir's work on intersection circles. In some sense, our new idea is to reintroduce the separability condition in a radical way ('by simply imposing it').

Underlying technique of this paper. We now give the basic idea of this paper. Recall from the abstract that the problem we solve is 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 $Vor(X)$.

This formulation subsumes (generalizes) all the work explicitly mentioned above since polygons and circles can be decomposed into circular and straight segments. It will be shown in the next section that the Voronoi diagram 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.

If there are m distinct endpoints among the segments of X, we introduce $m+1$ vertical lines such that every endpoint of a segment of X lies between a unique pair of adjacent vertical lines. The region between any pair of 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' the 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 essentially have computed the 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 takes $\Omega(n^2)$ time, simply because in the first stage, 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 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. (Note: actually, k should be the number of segments with endpoints in the slab.) This implies that each stage takes linear time and our stated time bound follows.

In the rest of this paper, we first 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. Section 5 and 6, respectively, prove the termination and the correctness of the merge process.[2] After the overall algorithm is given in section 7, we analyze its complexity in section 8. In section 9, we show the techniques in this paper lead to an $O(n\log n)$ solution to Sharir's variant of the Voronoi diagram for n intersecting circles. We conclude in section 10.

2. Preliminaries

Following [Kir79] 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 [Dry79] for an extensive discussion of these issues. For simplicity, we restrict the line segments to be finite and the circular arcs to be less than a semi-circle. Typically the line segments come from a set of polygonal objects and the arcs come from circles. Motivated by robotics applications (eg. [OY82]), 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: a corner is treated as a point or a singleton set as convenient.) 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. We remark that the assumption (b) in the definition of 'properness' is a technical convenience which ensures 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 [LD81]. Note that this definition is well-defined except when s is a circular wall and p is at the center of the circle containing s. 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 $Clearance_X(p)$: we call it the *clearance circle* at p (with respect to X).

[2] As Sharir [Sha83] pointed out, the correctness and termination of the Lee-Drysdale scanning procedure (which our algorithm depends on) is not at all obvious.

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 [OY82] or [OSY84a]. 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 such that the intersection of the clearance circle at p with $\bigcup X$ is a disconnected set.[3] The definition of the intrinsic diagram serves to motivate the next definition of the Voronoi diagram. It is essentially due to [Kir79]. Say a point p is *close* to an object s in X if for all $\epsilon > 0$ 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 $\bar{s}-s$). It is not hard to observe that p is close to s iff $Clearance(p) = d(p, s)$ and either (a) the projection of p onto s is indeed in s, or (b) if s is a wall and p is on the normal through an endpoint of s.

Definition. The *Voronoi diagram* $Vor(X)$ of X is the set of points p such that there exist two objects s_1 and s_2 that are close p.

By the last observation, $Vor(X)$ is simply the intrinsic Voronoi diagram of X augmented by additional line segments lying along the normals to the endpoints of walls (case (b) above). For computational purposes and for most of this paper, we prefer to use $Vor(X)$; but we will have a few occasions to consider the intrinsic diagram of X. These are the cases where the objects in X may arbitrarily intersect, and the resulting $Vor(X)$ becomes problematic as discussed by Drysdale.

For any pair of disjoint objects s, s', the (s, s')-*bisector* is the Voronoi diagram of the set $\{s, s'\}$. In this paper, we only consider (s, s')-bisectors where s and s' are objects from X. The properness of X implies that such a 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 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.)

Let us briefly note the types of bisectors: in case the objects are corners or straight walls, the bisector is familiar from previous work. For instance, if s, s' are both corners then the bisector is a line; if s' is a straight wall and s is its endpoint then the bisector is the line through s and normal to s'; if they are both straight walls then the bisector is a curve that may be composed of up to 7 sections of straight or parabolic lines (note: this case is essentially figure 2 if we ignore the endpoints of the two walls and the related Voronoi edges). Next we illustrate the basic types of interactions involving circular arcs. It is enough to use infinite lines and full circles instead of line segments and arcs. We also allow these circles and lines to freely intersect and hence we will use the intrinsic Voronoi diagram here. It is then easy to

[3] This elegant definition of Voronoi diagrams is an alternative way to overcome the above-mentioned difficulties in defining Voronoi diagrams.

verify the following:

(1) *Two non-intersecting circles*: the bisector is a hyperbola (unless the two radii are equal, in which case we have a straight line).

(2) *Two intersecting circles, neither containing the other*: the bisector is the union of an ellipse and a hyperbola, both passing through the two intersecting points of the circles. See figure.

(3) *Two 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. In the degenerate case where the two circles touch, there is an additional ray from the center of the contained circle directed away from the other center.

(4) *A circle of radius* r and a non-intersecting line L: a parabola with focus the center of the circle and directrix a line parallel to L and at distance r from L.

(5) *A circle of radius* r and an intersecting line L: two parabolas both passing through the two intersection points of L and circle. The directrices of the two parabolas are the two lines parallel to L and at distance r from L.

(6) *A circle and a point outside the circle*: a 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.)

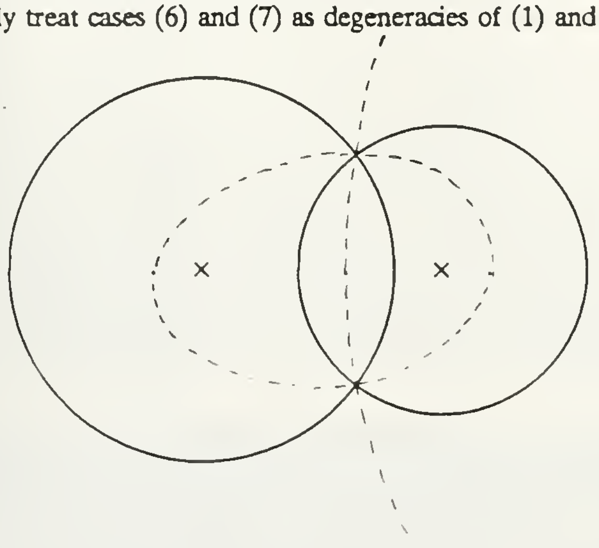


Figure 1. Bisector of two intersecting circles

Remark: The reader familiar with the work of [Sha83] will note that our definition of Voronoi diagrams (when restricted to full circles) differs from Sharir's (which is similar to that in [LD81]). 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 in our diagram, we obtain Sharir's. The disadvantage of negative distances is that it is not easy to generalize to circular arcs.

The Voronoi diagram can be decomposed into *Voronoi edges* where each edge e is determined by a pair of objects s, s'; in this case e is called an (s, s')*-edge*. It can be shown, by adapting standard proofs, that the number of Voronoi edges is linear. It is clear that the (s, s')-edge in $Vor(X)$ is a portion of the (s, s')-bisector. Because X is proper, each edge is a segment of a unique conic (rather than a union of such segments). The endpoints of Voronoi edges are called *Voronoi vertices*. The set of points in the plane that are neither on the $Vor(X)$ nor in any object of X is partitioned into connected components called *Voronoi cells*. For example, there are 8 cells in the next figure.

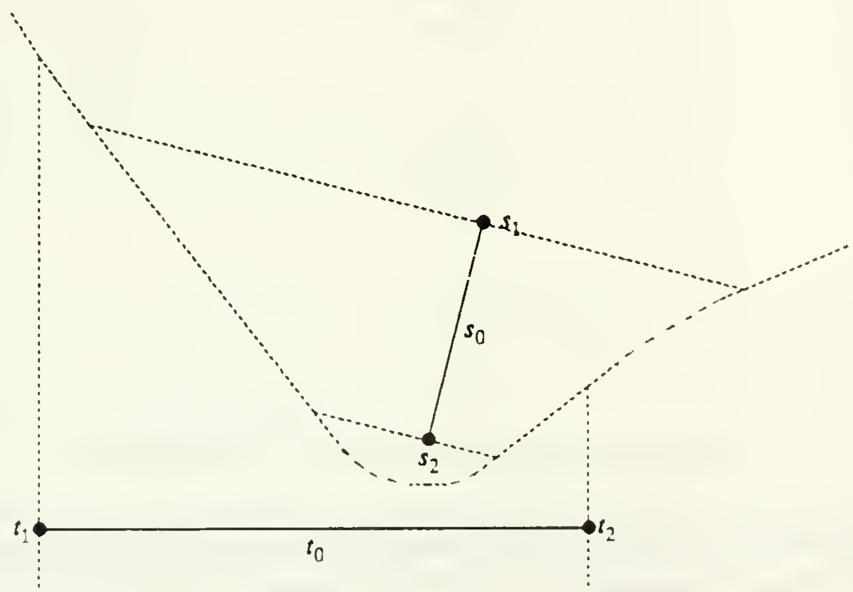


Figure 2. Voronoi cells

Each cell is associated with an object s where for all points p in the cell, p is close to s in the above sense. Conversely, a corner (resp. wall) is associated with at most one (resp. exactly two) cells. A cell associated with an object s is called a *s-cell*.

The remaining definitions have been invented for the technique in this paper. Let $m \le n$ be the number of corners in X. As in the introduction, let us introduce $m+1$ vertical lines called *separators* such that each corner is between a unique pair of adjacent separators. The strip between any two separators is called a *slab*. We may assume that a circular wall has the property that any vertical line intersects it at most once.

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 partitions the slab into closed regions that we call *quads*, so-called because these have four sides when they are finite regions. Thus if a slab has no long walls then the whole slab is the quad; otherwise, all but two of the quads are finite. A quad is said to be *active* if there is at least one corner in it; otherwise it is *inactive*. Let L be a separator. A *crossing* (or *s-crossing*) is the intersection of a separator with a wall s. If L has $k \geq 0$ crossings, then L is divided 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 of S.

Example:

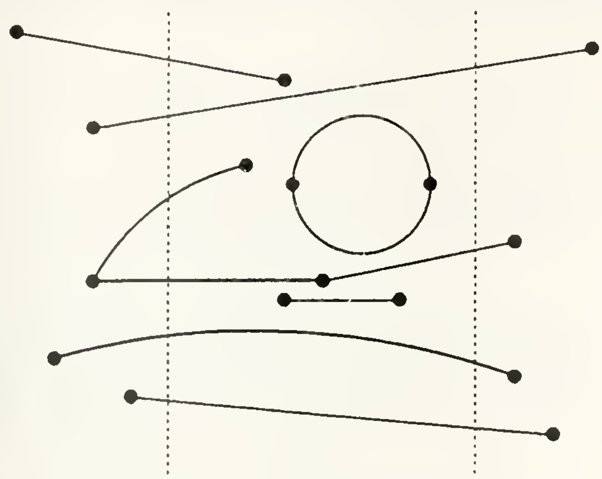


Figure 3. A slab between two separators (the dotted lines)

In the figure, we have a slab with 4 (not 5!) quads of which the top two quads are active. The second quad from the top has 3 windows on the left boundary and 2 on the right.

Let Q be a quad. A *Q-object* s is an object that is either a corner corresponding to a crossing at a vertical boundary of Q or an object of the form $s = s' \cap int(Q)$ where s' is an object of X and $int(Q)$ is the interior of Q. 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 $s' \cap int(Q)$. The *Q-diagram*, denoted $Vor(Q)$ (by abuse of notation), is the Voronoi diagram of the Q-objects. 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 a 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 in the slab. We represent a Voronoi diagram as an embedded planar graph ie., 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 each unbounded face, we introduce a fictitious edge at infinity.

3. Properties of Moving along a Bisector

Let s, 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 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 p. Let p_s denote the projection of p onto s. Let $\mathbf{u}$ be a tangent vector at p. First assume that it is not the case that one of the objects s, s' is a wall and the other an incident corner. 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, ie., $\mathbf{u} \times \mathbf{v} \neq 0$. Then we say $\mathbf{u}$ represents a *clockwise* (resp. *anticlockwise) direction about s* iff $\mathbf{u} \times \mathbf{v} > 0$ (resp. $\mathbf{u} \times \mathbf{v} < 0$). By symmetry, it is easy to see that a direction along e is clockwise about s iff it is anticlockwise about s'. Now suppose s' is a wall and s is an incident corner, and $\mathbf{v}$ is a vector from s directed away from s'. We say $\mathbf{u}$ represents a *clockwise direction about s'* iff it represents an *anticlockwise direction about s* iff $\mathbf{v} \times \mathbf{u} > 0$.

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

(1) Suppose that p^* moves along e in the direction that is clockwise about s_L. If s_R is a wall and p_R is in s_R then the motion of p_R (corresponding to that of p^*) either is stationary at an endpoint or is continuous and unidirectional along s_R. If s_R is a corner (in which case p_R is stationary) then the vector from p_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 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 $\mathbf{u} \times \mathbf{v} > 0$.

(3) Let L be a separator, s_L lie strictly to the left of L, and s_R lies on or to the right of L. Then there is a point on the (s_R, s_R)-bisector e beyond which moving clockwise about s_R has a positive downward component.

(4) The *front arc* of p^* is defined to be the arc of the clearance circle at p^* traversed by tracing 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:

Lemma 1. Let $s_L, s_R \in X$ and e be a (s_L, s_R)-edge in $Vor(X)$. If q_0, q_1 are the Voronoi vertices bounding e such that moving along e from q_0 to q_1 corresponds to clockwise about s_L then the clearance circle at q_0 (w.r.t. 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.

The proof of this lemma uses a technique taken from [OSY84a]. 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 disc bounded by the clearance circle at $p^*(t)$,

[ii] the sets Γ_t, 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 $\Gamma_t \supseteq \Gamma_u$ and $\Sigma_u \supseteq \Sigma_t$. In fact, the growth and shrinkage have the following stronger property: the front (resp. back) arc at $p^*(u)$ (resp. $p^*(t)$) lies in the interior of Γ_t (resp. Σ_u).

The preceding growth properties of Γ_t and Σ_t immediately imply the lemma. We now describe the Γ and Σ sets. It is clearest to describe these sets according to the type of conic that e conforms to. To avoid tedium, we will only treat the case of hyperbolas and ellipses: the case where e is parabolic or straight is essentially treated in [OSY84a]. (We remark that in [OSY84a], one of the cases also has a third family of sets Δ_t, but this can be avoided.)

[4] The Hausdorff metric on closed subsets of a metric space Y is as follows: for any set $S \subseteq Y$, $\epsilon > 0$, 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 $d(S, S') = inf\{\epsilon \geq 0 : S \subseteq S'_\epsilon \text{ and } S' \subseteq S_\epsilon\}$.

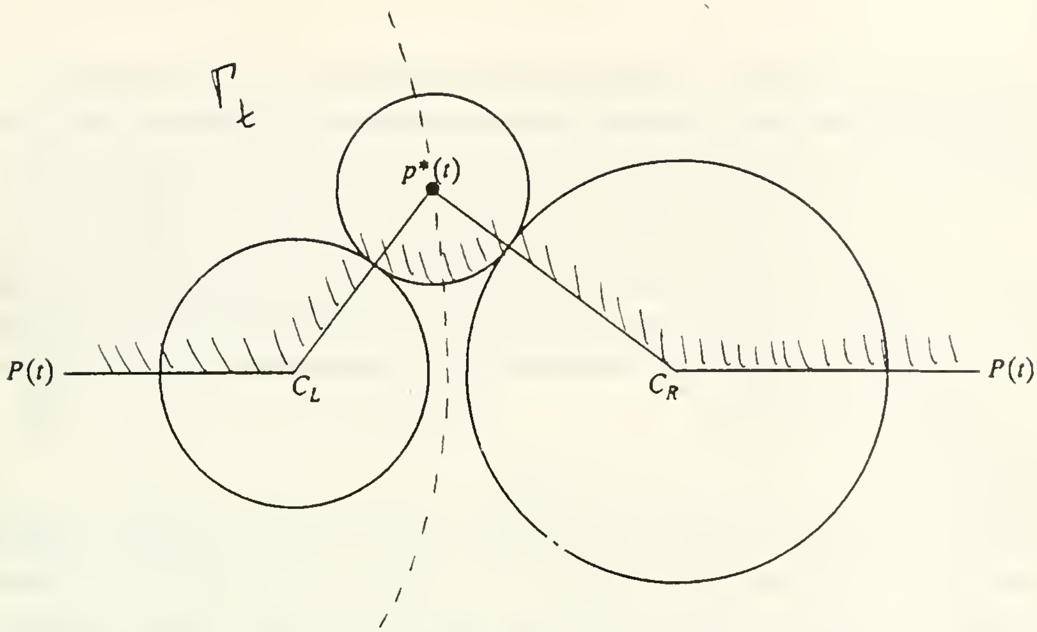


Figure 4. The case of hyperbolas with non-intersecting circles

(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. 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 disc bounded by the clearance circle at $p^*(t)$. It is easy to verify properties [i] to [iii] above. Now consider the case where C_L and C_R intersect at a pair of points r, r'. Since s_L and s_R are assumed not to intersect, e must lie in one of the three sections of the hyperbola determined by r and r'. If e 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 interior of C_L and of 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 are walls lying in some circles C_β. However, we will not assume that the centers of the circles lie on the x-axis. Let K denote the strip bounded by the two horizontal lines through the centers of C_L and C_R. Again we consider two cases depending on whether they intersect. First suppose they do not intersect. Then one (say C_R) must lie inside the other. Note that, because of our assumption that a vertical line intersects an object at most once, e must lie outside of the strip K. Without loss of generality, let e lie above K and furthermore e lies in the interior of C_L (and hence exterior to C_R). Let J' be the interior of C_L minus the interior of C_R. So J' has a crescent shape. Let J be the part of J' above K. Therefore e lies in J. In a natural way, the front arc of $p^*(t)$ divides J into a left and a right part. Define Γ_t to be the part of J to the right part of this division. Similarly define Σ_t to be the left part of the division using the back arc of $p^*(t)$. It is again easy to verify [i-iii] for the sets Γ_t and Σ_t. The other case, where the two circles intersect should be easy to defined in analogy to the preceding.

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 L_M. By definition, this means we compute the Q-diagram of each active quad Q of S. Note that $Q_L = Q \cap S_L$ is a union of one or more quads in S_L, none of which are necessarily active. The diagram of each quad Q' in Q_L 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 $Q \cap S_L$, 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-diagram. This is the most important part of the algorithm.

We can obtain the Q_L-diagram from all the Q'-diagrams (for $Q' \subsetneq Q_L$) rather easily by using the next lemma whose easy proof is omitted. Let Q_1, Q_2 be two adjacent quads of 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_0.

Lemma 2. 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 $C_i(s_1)$ $(i = 0, 1, 2)$ is unbounded. Assume Q_1 lies above Q_2. The boundary of the $C_0(s)$ 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 if $s = s_3$.

(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).

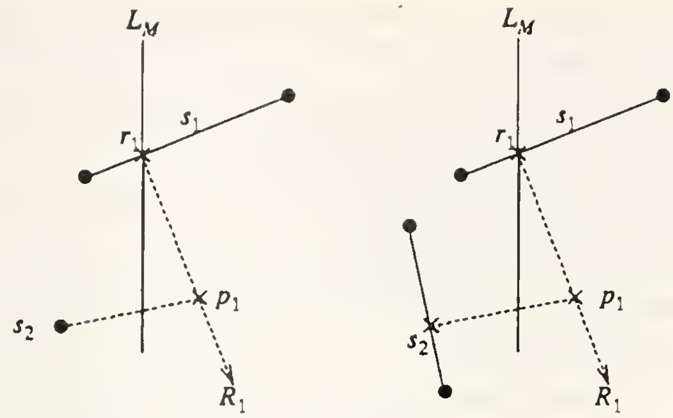
(3) If s is not one of s_1, s_2, or s_3, then it is a Q_i-object for a unique $i = 1, 2$ and $C_0(s)$ is equal to corresponding cell in the Q_i-diagram.

By repeated application of these observations, the Q_L-diagram can be obtained in $O(m)$ time. Similarly we can compute the Q_R-diagram.

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-one correspondence between these components and the windows in $L_M \cap Q$. Therefore, we will call the component corresponding to a 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 section.

Let W be fixed window in $Q \cap L_M$. 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 whose crossings at L_M determine the upper and lower endpoints of W, respectively. Let r_i ($i = 0, 1$) be the s_i-crossing. Consider the ray R_1 emanating from r_1 downwards 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- 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 starters is well-defined for a finite window because the ray R_1 has a downward component and must eventually get closer to r_0 then to r_1. Similarly, let R_0 be the ray emanating upwards 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 the top from p_1 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. Hence we may assume that this is not the case.



s_2 is a corner $\qquad$ s_2 is a wall

Figure 5. Illustrating the starter

The procedure we now describe is essentially that of [LD81]. We construct the W-contour as a sequence of curve segments

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

connecting a sequence of points

$$p_1, p_2, \ldots, p_i, \ldots,$$

satisfying the following:

(a) Each σ_i is part of the (s_i^L, s_i^R)-bisector e_i, where s_i^β ($\beta = L, R$) is a Q_β-object.

(b) The endpoints of σ_i are p_i and p_{i+1} where the direction from p_i to p_{i+1} along e_i is clockwise about s_i^L (and thus anticlockwise about s_i^R). (Henceforth, call this the *forward* direction along e_i. Note that this definition of forward is ambiguous if $\{s_i^L, s_i^R\} = \{r_0, r_1\}$ where r_i is as above. But this has been excluded.) Clearly p_i is in the intersection of the bisectors e_{i-1} and e_i, $i > 1$.

Inductively, suppose that p_i, s_i^L, and s_i^R have been computed. We now extend this to $i+1$. First we first show how p_{i+1}, s_{i+1}^β are defined, returning to their computation below. Let C_i^β be the s_i^β-cell such that $\sigma_i \subseteq C_i^L \cap C_i^R$. Consider the component of $e_i \cap C_i^\beta$ containing p_i: let q_i^β and r_i^β be the endpoints of this component. Assume that the forward direction along e_i is from q_i^β to r_i^β. Hence r_i^β comes after p_i along e_i. We define p_{i+1} to be the closer of r_i^L and r_i^R to p_i. Suppose without loss of generality that p_{i+1} is r_i^L. Then p_{i+1} is the intersection of e_i with a Voronoi edge of the Q_L-diagram. Let s_{i+1}^L be the Q_L-object such that this edge is part of the (s_i^L, s_{i+1}^L)-bisector. Under the 'general position' assumption that no clearance circle intersects more than three objects, then s_{i+1}^L is uniquely defined and s_{i+1}^R will be defined to be s_i^R. In this case, our inductive definition of p_{i+1}, s_{i+1}^β is complete. But in general, the clearance circle at p_{i+1} (w.r.t. Q-objects) may touch more than three objects. By the lemma in the previous section, the objects can only touch the clearance circle along the front arc (recall that this is the arc from the contact point with s_i^R clockwise to the contact point with s_i^L). It is also clear that if we order the objects that touch the front arc (starting with s_i^R and moving clockwise around the arc), we will encounter all the Q_R-objects before any Q_L-object. 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). This completes the conceptual description of the W-merge.

We still have to initialize the merge and specify the termination condition. For $i = 1$, p_i is just the starter. Recall that above we assume that the starter is the intersection of the ray R_1 with the (s_1, s_2)-bisector, with s_2 a Q_L-object. In this case, we let s_i^L be s_2 and s_i^R be the Q_R-wall $s_1 \cap int(Q_R)$. The condition for termination is when p_{i+1} is equal to the ender p_0.

We now attend to the computational details for obtaining the point p_{i+1}. It is sufficient to show how to obtain the point r_i^β. The following is a well-known method of obtaining r_i^β from q_i^β where we recall that q_i^β is the point where e_i last (re-)entered cell C_i^β: starting from q_i^β, scan the boundary of C_i^β clockwise (if $\beta = L$) or anticlockwise (if $\beta = R$) until the first time e_i intersects the boundary. This point is r_i^β. We shall call this the *Shamos-Hoey scan*. It is not hard to see that the Shamos-Hoey scan may examine an edge more than a constant number of times.

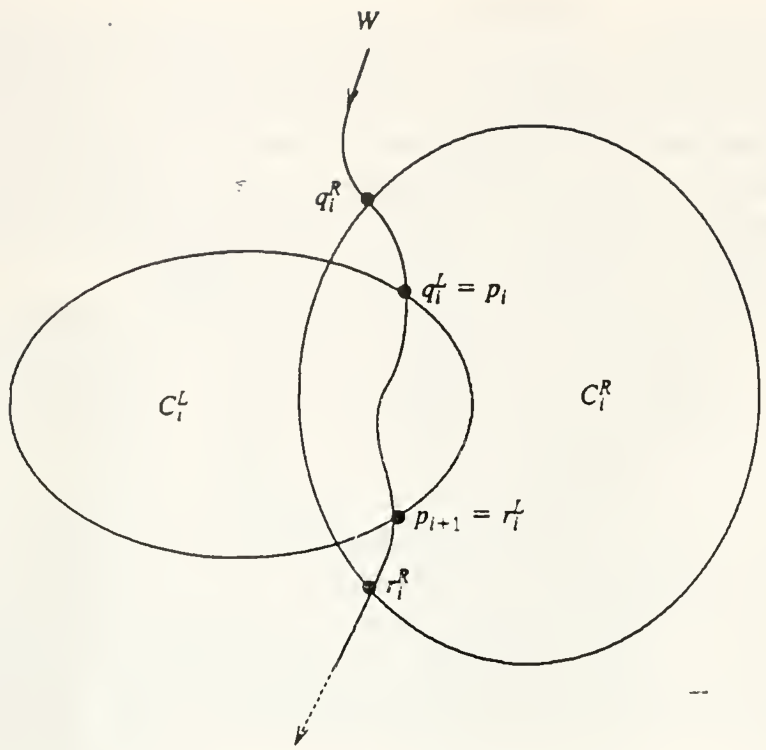


Figure 6. Schematic illustrating the construction of the W-contour

To improve on this simple scheme, we must analyze the properties of the points p_i. Our goal is to avoid rescanning edges of a cell C_i^β if the same cell is encountered for a succession of values of i. Suppose that $C_i^R = C_{i+1}^R$. The clearance circle at p_i is divided into the front arc and back arc by the radius joining p_i to the closest points in the objects s_i^L and s_i^R. The (s_i^L, s_i^R)-bisector e_i continues beyond p_i to meet the boundary of C_i^L at r_i^L. On the other hand the W-contour continues beyond p_i by following the (s_{i+1}^L, s_{i+1}^R)-bisector e_{i+1}.

Definition. The *truncated version* $\hat{C}_i^R$ of C_i^R is defined to be the part of C_i^R to the right of the curve comprising the W-contour from q_i^R to p_i and the part of e_i from p_i to r_i^R. $\hat{r}_{i+1}^R$ is defined as the first intersection of e_{i+1} with the boundary of $\hat{C}_i^R$ scanning anticlockwise from p_i.

To see that $\hat{C}_i^R$ is well-defined, we must show that the curve in the definition divides C_i^R into a left and right hand side; but this follows from the fact that for any pair of points $p \neq p'$ on this curve, the shortest open line segments from p and p' (respectively) to s_i^R are disjoint.

Lemma 3. In the definition of the point p_{i+1}, we could have replaced the cell C^R_{i+1} with the truncated version $\hat{C}^R_i$, and replace the point r^R_{i+1} by $\hat{r}^R_{i+1}$.

Proof. There are two cases to consider: (i) Suppose that after p_i the bisector e_{i+1} next rejoins e_i at a point q before it reaches r^R_{i+1}. By definition, $\hat{r}^R_{i+1} = q$. But q is in the (s^L_i, s^L_{i+1})-bisector. Hence q cannot be in the interior of C^L_{i+1}. This implies that the point r^L_{i+1} (where e^L_{i+1} first meets the boundary of C^L_{i+1}) lies at or before q. Thus p_{i+1}, if taken as the earlier of $\hat{r}^R_{i+1}$ or r^L_{i+1}, would agree with the original definition. (ii) Suppose case (i) does not hold. So either e_{i+1} does not intersect e_i beyond p_i, or e_{i+1} reaches r^R_{i+1} before reintersecting e_i. The W-contour, in making the transition from e_i to e_{i+1} at p_i makes a left-ward turn, by remarks in section 3. Hence the point r^R_{i+1} lies after r^R_i (scanning anticlockwise from p_i). In this case $\hat{r}^R_{i+1}$ is equal to r^R_{i+1}, so the lemma is again true. □

It is important to note that this lemma can be iterated for each value of i and (by symmetry) we could similarly used $\hat{C}^L_i$ in place of C^L_i. Thus we can successively truncate the cells C^β_i by replacing those edges that have been scanned by an appropriate section of e_i. We now summarize the scanning procedure for computing p_{i+1}, assuming that p_i, s^β_i, and the truncated cells $\hat{C}^\beta_i$ are available:

[a] We are at p_i. Compute the values s^L_{i+1} and s^R_{i+1} in the manner outlined above (this involves checking all Voronoi edges that intersect p_i). This gives us $\hat{C}^\beta_{i+1}$ and e_{i+1}.

[b] For each $\beta = L, R$, search for the first intersection $\hat{r}^\beta_{i+1}$ of e_{i+1} with the boundary of $\hat{C}^\beta_{i+1}$. Do the search starting from p_i, scanning clockwise iff $\beta = L$.

[c] Choose p_{i+1} to be the earlier of $\hat{r}^L_{i+1}$ and $\hat{r}^R_{i+1}$.

[d] Update $\hat{C}^\beta_{i+1}$. Replace the part of its boundary that is examined in [b] from p_i to $\hat{r}^\beta_{i+1}$ by the segment of e_{i+1} between the same pair of points.

This scan will be called the *Lee-Drysdale scan*. This completes our discussion for the case of a finite window W. 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 downwards from the s_1-crossing for some Q-object s_1. The starter p_1 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_1-crossing meets no Voronoi vertices of $Vor(Q_L)$ and $Vor(Q_R)$. If the starter is undefined, then we say the W-contour is empty. If it is defined, the merge process just described can be applied. Although there is no ender we can use the following to give us a criterion for termination: 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. Clearly there are exactly two choices for a pair p_L, p_R of adjacent vertices in H where p_β is a Q_β-corner. Choose the pair that is the lower of the two, where 'lower' makes sense since the segment $[p_L, p_R]$ intersects L_M.

Lemma 4. If the W-contour is defined then the W-contour eventually coincides with the (p_L, p_R)-bisector.

We omit the easy proof. 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. Indeed, this amounts to computing the points p_L and p_R from H_L and H_R. After checking that the starter is defined we can do the usual merge, looking for the (p_L, p_R)-bisector as given by the lemma.

(b) Suppose W is the entire separator L_M. The W-contour is always defined in this case. It is clear how the previous lemma leads to a criteria for initializing and terminating the W-merge. Note that this case is just the 'separable' situation that arises in the original algorithm Hoey and Shamos.

We have completed the description of the W-merge. It is now fairly easy to construct the Q-diagram from the set of W-contours, $W \subseteq Q \cap L_M$. This amounts to computing the cells of each Q-object: For each crossing r at L_M, discard the r-cells in $Vor(Q_L)$ and $Vor(Q_R)$; if r is the s-crossing where s is a Q-wall, we can form the two s-cells by merging the appropriate pair of cells from $Vor(Q_L)$ and $Vor(Q_R)$. For the other cases, if s' is a Q-object, then s' is a Q_β-object for a unique β. In this case, the s'-cells in $Vor(Q)$ are directly taken from $Vor(Q_\beta)$. Note that if s' is adjacent to a W-contour then the W-merge would have truncated the original s'-cell to the desired output.

5. Termination

The notations relative to a window W from the previous section is retained in this section. We now show that the W-contour eventually reaches the ender p_0 if W is finite. It is convenient in this section to regard the W-contour as a parametrized curve $p^*(t)$, and p^* as a moving point. Suppose p^* is currently moving along σ_i in the (s_i^L, s_i^R)-bisector, for some i. If p^* is not at an endpoint (p_i or p_{i+1}) then the projection of p^* onto the object s_i^β ($\beta = L, R$) is well-defined and will be denoted $p^*_\beta = p^*_\beta(t)$. If $p^*(t) = p_{i+1}$, we (arbitrarily) define p^*_β to be the projection of p^* onto s_i^β (rather than s_{i+1}^β). Clearly $p^*_\beta(t)$ moves in a piecewise continuous manner. It is natural to regard the path consisting of the two segments $[p^*_L, p^*]$ and $[p^*, p^*_R]$ as a 'moving front'. We propose to study the motion of the point $x^* = x^*(t)$ defined by the intersection of L_M with the moving front:

$$\{x^*\} = L_M \cap ([p^*_L, p^*] \cup [p^*, p^*_R]) \qquad (*)$$

Note that x^* is well-defined if p^*_L and p^*_R lie on opposite sides of the separator L_M: this will be shown below. Recall that the ender p_0 lies on the ray R_0 originating from the s_0-crossing r_0.

Lemma 5. Let the W-contour intersect the ray R_0 at some point q. If

$Clearance_Q(q) = d(q, r_0)$ then $q = p_0$.

Proof. Suppose the clearance circle C at q (w.r.t. Q-objects) touches r_0. Let C_0 be the clearance circle at p_0 (w.r.t. Q-objects). Note that both circles are tangent to the Q-wall passing through r_0. If either circle properly contains the other, we derive an immediate contradiction. Hence $C_0 = C$ and $q = p_0$. □

Lemma 6. The W-contour does not self-intersect.

Proof. If the W-contour self-intersects then let p be a point in the interior of one of the finite regions bounded by the W-contour. Let p_β be the first point on the W-contour met by a horizontal ray originating in the 'β-ward' direction. Let C_β be the clearance circle at p_β (w.r.t. Q-objects). Since C_β must touch both a Q_L- and a Q_R-object, it follows that L_M intersects C_β. This implies that C_L and C_R intersect. Let L be the vertical line through the two intersection points of $C_L \cap C_R$. If L_M is strictly left of L 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 L_M coincides with L. But in this case, we easily see that r_1 and r_0 are the two points of the intersection $C_L \cap C_R$. This implies that p_R is both the ender and the starter, contradiction. □

Lemma 7. If $p^*(t)$ is not equal to p_1 or p_0 then $p^*_L(t)$ and $p^*_R(t)$ lie strictly to the left and right (resp.) of L_M.

Proof. It follows from the two previous lemmas that for each $\beta = L, R$, none of the objects s_i^β $(i = 1, 2, \ldots,)$ can be equal to r_0 or r_1 (for if $s_i^\beta = r_0$, say, then $p^* = p_0$, but this would imply that the contour self-intersect). Hence each s_i^β lies strictly to the 'β-side' of L_M. Now $p^*_\beta(t)$ lies in the closure of s_i^β and X is proper implies $p^*_\beta(t)$ in fact lies in s_i^β. Hence the lemma. □

Therefore the point $x^*(t)$ is well-defined for $t > 0$. We may extend this definition by defining $x^*(0)$ to be $x^*(0^+)$. Our goal is to prove that $x^*(t)$ is monotonically moving downwards along L_M. Suppose the ender p_0 is the intersection of the ray R_0 with the (s, s_0)-bisector where r_0 is the s_0-crossing and s is some Q_L- or Q_R-object. Let p' be the projection of p_0 onto s (see next figure). Define the point x_0 to be the intersection of L_M with the segment $[p_0, p']$. The next lemma shows that the W-contour terminates at p_0 iff $x^*(t)$ terminates at x_0.

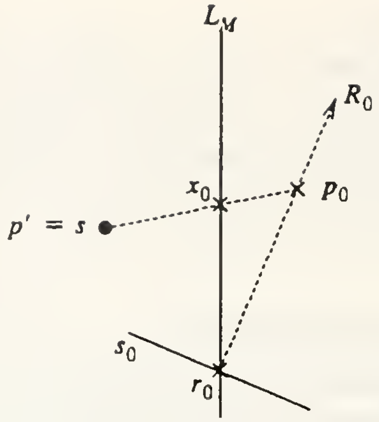


Figure 7. The point x_0

Lemma 8. (a) For all $t \geq 0$ corresponding to a point $p^*(t)$ on the W-contour, $x^*(t)$ is above or equal to x_0. (b) If $x^*(t) = x_0$ then $p^*(t) = p_0$.

Proof. (a) Let C_0 be the clearance circle at p_0 (w.r.t. Q-objects). Let $C^* = C^*(t)$ be the clearance circle at $p^*(t)$ (w.r.t. Q-objects). If $C^* = C_0$ then $p_0 = p^*$ and $x^* = x_0$. Hence assume $C^* \neq C_0$. Let q_h and q_d be the higher and lower (resp.) of the two points of intersection of L_M with C^*. Similarly let r_h be the point such that C_0 intersects L_M at r_h and r_0. Note that $C^*(t)$ changes continuously with t and hence $q_d(t)$ moves continuously along L_M. It is therefore clear that $q_d(t)$ cannot pass below r_0. We also claim that q_h must be higher than r_h: suppose otherwise. Then, since C^* cannot be properly contained in C_0, it is easy to see that both points of the intersection of C_0 and C^* lie to one side of L_M, say, to the left side.

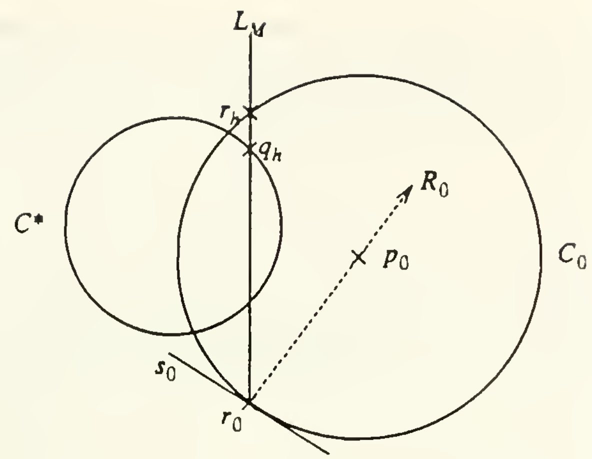


Figure 8. Case where q_h lies below r_h

This implies that the part of C^* lying right of L_M is fully contained in C_0. But this part touches some Q_R-object, giving a contradiction. Assuming then q_h is above r_h, we finally consider two possibilities:

(i) Suppose the segments $[r_0, r_h]$ and $[q_d, q_h]$ are disjoint. Then clearly x^* is above x_0 since $x^* \in [q_d, q_h]$ and $x_0 \in [r_0, r_h]$.

(ii) Suppose $[r_0, r_h]$ and $[q_d, q_h]$ intersect. Let y_L and y_R be the two intersection points of C_0 and C^*. Clearly y_L and y_R lie on opposite sides of L_M. Suppose without loss of generality, $[p_0, y_L]$ intersects L_M (hence p_0 must be on the right side of L_M). Note that x_0 lies below the intersection z of L_M with the segment $[p_0, y_L]$. Similarly, x^* lies above the intersection z' of L_M with the union of segments $[y_L, p^*] \cup [p^*, y_R]$. But it is easy to verify that z is below z' and hence x_0 lies below x^*.

This completes the proof of part (a). Part (b) follows from the fact that $x^*(t)$ is strictly moving downwards with t: if $x^*(t) = x_0$ and $p^*(t) \neq p_0$ then $x^*(t+\epsilon)$ is strictly below x_0, contradicting (a). □

We are now ready to prove the main result of this section:

Lemma 9. *(Termination)* The W-contour for a finite W must terminate at the ender p_0.

Proof. Suppose first that $p^*(t)$ is not equal to any p_i. If $x^*(t)$ is the intersection of L_M with $[p^*_L(t), p^*(t)]$ then the fact that $p^*(t)$ is clockwise about s_L implies $x^*(t)$ is moving downwards. Similarly if $x^*(t)$ is the intersection of L_M with $[p^*(t), p^*_R(t)]$. So suppose $p^*(t)$ is equal to some p_i. Then x^* may suffer a discontinuity at t. If t is not a discontinuity for x^* then the preceding arguments show that x^* is moving downwards at t. So suppose there is a discontinuity. Then $x^*(t^-) = x^*(t) \neq x^*(t^+)$. But then lemma 1 implies that $x^*(t^+)$ lies strictly below $x^*(t)$. Finally, lemma 8 implies that $x^*(t)$ is never strictly below x_0. So $x^*(t)$ eventually reach x_0 and the same lemma asserts that this is equivalent to $p^*(t) = p_0$. □

6. Correctness

For each window W, the notations relative to W from the two previous sections will be retained. From the preceding development, we know that the Lee-Drysdale scan terminates and is correct (ie., equivalent to the Shamos-Hoey scan) when applied to a *single* window W. Note that the Lee-Drysdale scan is assumed whenever we refer to any 'W-merge' below. Now we prove that there is no interference between the W-merges for different windows W. In particular, we need to be sure that the modifications to the cells of $Vor(Q_\beta)$ made during a W-merge will not affect subsequent merges.

Lemma 10. Let W and W' be two windows of $Q \cap L_M$. Assume W lies above W' and r_0 be the lower endpoint of W. Let L be the horizontal line through r_0.

(i) The W- and W'-contours are disjoint. In fact they lie on opposite sides of L.

(ii) No Voronoi edge is examined in both the W-merge and the W'-merge.

(iii) The order of doing the W-merge and the W'-merge is immaterial.

Proof. Let R be the ray from r_0 extending horizontally eastward. As in lemma 2, we know that R is contained in the r_0-cell in $Vor(Q_L)$. Clearly (i) follows by symmetrical considerations if we show that the W-contour does not intersect R: this in turn follows from following fact (recall the termination proof) that for any point p on the W-contour, if p is right of L_M then the shortest segment joining p to a Q_L-object intersects L_M above the point x_0 (recall x_0 from lemma 8). To see (ii), note that any portion e of an edge in $Vor(Q_R)$ scanned in the W-merge has the following property: it intersects or lies to the left of the W-contour. By 'lying to the left' we mean that for each point p in e, the shortest segment joining p to a closest Q_R-object must intersect the W-contour. Suppose that e is the portion of an edge of $Vor(Q_R)$ that is examined in both merges. If p is any point in e then by the preceding observation, there is a shortest segment from p to a closest Q_R-object which intersects the W-contour. Similarly, there is a shortest segment that intersects the W'-contour. But these two intersections lie on opposite sides of the horizontal line L. It follows from the triangular inequality that p must in fact be closest to the Q_R-object r_0. But the W- and W'-merges do not examine any edge on the boundary of the r_0-cell, contradiction. Part (iii) follows from (ii) and the fact that Voronoi edges are first examined before being modified (discarded or truncated). □

We finally show that the Q-diagram constructed is indeed the Voronoi diagram of the set of Q-objects. 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 non-empty, the extended W-contour can be defined analogously. If W is empty, then W corresponds to a half-line determined by s-crossing (for some Q-wall s). In this case, define the extended W-wall to be the half-line originating from the s-crossing in a direction normal to s and away from the Q-objects. Define C to be the union of all the extended W-contour where W ranges over those windows in L_M that intersects Q. We claim that C divides the plane into two infinite regions that can be naturally distinguished as the left and right sides of C. This follows from the following two facts:

(a) C is a doubly infinite line.

(b) C does not self-intersect. This follows from the fact that the individual W-contours do not self-intersect and distinct W-contours are disjoint.

Lemma 11. All the Q_R-objects lie to the right of C and similarly for the Q_L-objects. Only the objects corresponding to crossings at L_M lie in C itself.

Proof. Let s be a Q_R-object to the left of C, and assume that s does not lie in L_M. We will derive a contradiction. Let p be the first point on C horizontally to the right of some point q in s. Clearly p is to the right of L_M.

(a) Suppose p is in the W-contour for some W. The clearance circle C at p (w.r.t. 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 L_M. Since the shortest line segment from p to L_M 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. Then p is near to a corner r in L_M. 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 $Vor(Q_R)$ lies between s and p, contradiction. □

We now prove the main result of this section:

Lemma 12. *(Correctness)* The Q-diagram $Vor(Q)$ is the union of all the W-contours together with the portion of $Vor(Q_L)$ to the left of C and the portion of $Vor(Q_R)$ to the right of C.

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

$$Clearance_{Q_L}(p) < Clearance_{Q_R}(p).$$

Suppose the claim is false and for some Q_R-object s, $Clearance_{Q_L}(p) \geq d(p, s)$. Let $q \in s$ such that $d(p, s) = d(p, q)$. Then C intersects the half-open segment $(p, q]$. Let r be a point in the intersection of 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 (w.r.t. Q-objects). The interior of C_0 does not intersect any Q_L-object. Clearly C_1 is contained in C_0 (otherwise q would be in the interior of C_1). Consider two cases:

(a) q is not a corner in L_M. 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 L_M. Let q be the s'-crossing where s' is a Q-wall. Thus p is on the line through q and normal to s'. Thus p lies in C, contradiction. □

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.

(1) *Pre-sort:* Sort the set of corners according to their vertical projection onto the x-axis.

(2) *Pre-scan:* 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-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 pre-sorting. During the pre-scan we also introduce the set of separators. Let $left(s)$ and $right(s)$ indicate the two separators adjacent to s. (Note: to handle the case of more than one corner in any vertical line, only trivial modifications are necessary in our entire preceding development.)

(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 $Vor(X)$.

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 of the pre-scanning, is available to the procedure via global variables. The recursive procedure returns

(i) a list of the active quads of S and a list of windows for each quad, (ii) a list of the Voronoi diagrams of these quads, and (iii) the convex hull of the Q-objects for the topmost and bottommost active quads Q in the slab.

Recursive Procedure

Input: (m, S) where S is a slab and m the number of corners in S.

(1) *Basis:* If $m = 1$, then use the pre-scanning information to determine the unique corner p in S and also the quad Q containing p. Compute the diagram of Q and return. (Note that the division of walls into subobjects at their crossings is done at this point in the algorithm.)

(2) *Divide:* Divide the slab S into 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 S_L and S_R. Let L_M denote the separator between S_L and S_R.

(3) *Conquer:*

(3.1) *Vertical Merge.* It is actually quite interesting to determine the list of active quads Q in S. In order to preserve the continuity of the main description, we defer this to the end of this section. For now we just note that for each active $Q \subseteq S$ with k objects, we can gather together the set of active and inactive quads of S_β ($\beta = L, R$) whose union forms Q. Furthermore, if S has k active objects then this takes time $O(k)$ over all the active Q. We then merge the diagrams of the quads in $Q_\beta = Q \cap S_\beta$ to obtain the diagram of Q_β, using the method described in section 4.

(3.2) *Horizontal Merge.* We assume that a list of the active quads Q of S and also a list of the windows of each Q is available from the previous step. We apply the W-merge to each window $W \subseteq L_M \cap Q$, for each Q. Finally, we construct the Q-diagram from these. Note that in the process of doing the W-merge for the topmose and bottommost quads, we have also computed their convex hulls.

End Recursive Procedure.

We are now done except for the details on determining the active quads of S from the corresponding lists for S_L and S_R. Let us introduce some terminology relative to any slab S and separator L: an *S-interval* I in L is one of the form $I = L \cap Q$ for some quad Q in S. An *S-active* interval is similarly defined except that the quad under consideration must be active. Two intervals in L *overlap* if their interiors intersect. Given a set I of intervals in L, the *I-equivalence* relation on I is defined as the reflexive, symmetric, and transitive closure of the overlap relation on I. Let the slab S be divided into slabs S_L and S_R by the separator L_M. Let I_β be the set of S_β-active intervals in L_M.

Lemma 13. There is a bijective correspondence between the set of S-active intervals in L_M and the set of $(I_L \cup I_R)$-equivalence classes such that if I is an S-active 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 S-active interval. The set

$$G_I = \{J \in I_L \cup I_R : J \subseteq I\}$$

of S_L- and S_R-active intervals inside I must be non-empty. Note that an interval not in G_I cannot be $(I_L \cup I_R)$-equivalent to any interval in G_I. Hence G_I is a union of equivalent classes. Claim: 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 (ie., $1 < i < k$) 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 S_R-active 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 L_R, making the S_R-interval containing x_2 active. So let x_i $(i > 2)$ be the lower endpoint of J (the upper endpoint is x_1). A similar argument shows that x_i is in the interior of some S_L-active interval I_2. Furthermore, I_1 and I_2 overlaps. Continuing this way, we eventually get to an interval whose lower endpoint is x_k. □

Using this lemma, we can compute the list of the S-active intervals at each of the two separators bounding S: assume that inductively we have a list of the S_β-active intervals at each of the separators bounding S_β. We can now 'merge' the list of S_L and S_R at L_M in the straightforward way. When done, we can easily convert this list into the corresponding S-active list of intervals at the two boundary separators of S.

8. Complexity

We first analyse the complexity of the W-merge.

Lemma 14. The W-merge takes time $O(k+l)$ where k is the number of segments $\sigma_1, \sigma_2, \ldots, \sigma_k$ comprising the W-contour and l is the number of examined edges from $Vor(Q_L)$ and $Vor(Q_R)$.

Proof. The work to construct σ_i is proportional to the number of edges of the cells $\hat{C}_i^\beta$ $(\beta = L, R)$ examined in the scan. Refer to section 4, steps [a-d] of the scanning procedure:

Step [a]:

Here we determine all those Voronoi edges that terminates at p_i. It is not hard to show that none of these edges, except for the edges that bound the cells C_{i+1}^L, C_{i+1}^R, will ever be examined again. Hence the work done in this step, summed over the entire W-merge, is $O(k)$ for those edges that will not be examined again and $O(l)$ for the others.

Step [b]:

Among the edges of $\hat{C}^{\beta}_{i+1}$ that are scanned in step [b], all except possibly the first and second edges are scanned for the first time. We can charge the cost of scanning the first two edges to σ_{i+1}, and the rest are charged to the edges themselves. The overall charges for these are $O(l)$ and $O(k)$, respectively.

Steps [c,d]:

These can be made $O(1)$ for each segment σ_i, and hence $O(l)$ when summed over the entire W-merge. □

Lemma 15. The construction of a Q-diagram is $O(m)$ if there are m Q-objects.

Proof. Recall that before the 'horizontal' merging is applied to $Vor(Q_L)$ and $Vor(Q_R)$, the 'vertical' merging is done. The time bound is $O(m)$. To bound the time for horizontal merging, recall that in the last section we have essentially showed that the separate W-merges do not interact. From the previous lemma the work done in the W-merge is $O(k_W + l_W)$ where k_W is the number of segments comprising the W-contour and l_W is the number of examined edges. The sum $\sum_W k_W$ (over all windows W of Q) is order of the number of segments in the curve C, and the sum $\sum_W l_W$ is order of the number of Voronoi edges in $Vor(Q_L)$ and $Vor(Q_R)$. Since $Vor(Q)$ has no more edges than the combined number of edges in $Vor(Q_L)$, $Vor(Q_R)$ and C, it is clear that the sum $\sum_W (k_W + l_W)$ over all windows W is $O(m)$ where m is the number of Q-objects. □

Theorem 16. The Voronoi diagram of a set X of n pairwise disjoint objects cab be computed in $O(n\log n)$.

Proof. As in the introduction, we can divide the work done in the various recursive calls into *stages*, where each stage corresponds to processing a collection of slabs $S_1, S_2, \ldots, S_p$, where these slabs are disjoint and the union of the slabs contains all the objects in X. Since each object occurs in at most two active quads in these slabs, and the processing of each quad is linear in the number of objects in it, we conclude that a stage takes linear time. But there are $O(\log n)$ stages. □

9. Diagram for intersecting circles

[to be filled in]

10. Conclusion

This paper solves the open problem of an $O(n\log n)$ algorithm for the problem of computing the Voronoi diagram of a set of points, line segments and circular arcs. The algorithm is simple enough to have an impact on practical applications, and in particular, in motion-planning for robots. The technique clearly works for more general algebraic curves, provided we take care to break up each curve into a number of suitably small sections. It also seems that the technique can be generalized to handle the Voronoi diagram of a set of polyhedral objects. This is a subject of further research.

Acknowledgement. It is a pleasure to thank Sally Howe of the National Bureau of Standards for rekindling my interest in the problem.

REFERENCES

[AE83]. Aurenhammer, F. and Edelsbrunner, H., "An optimal algorithm for constructing the weighted Voronoi diagram in the plane," Institute fur Informationsverarbeitung, Technical University of Graz (Jan. 1983).

[Dry79]. Drysdale, R. L. III, "Generalized Voronoi Diagrams and Geometric Searching," STAN-CS-79-705, Computer Science Tech. Report, Stanford Univ. (1979). (PhD Thesis)

[Kir79]. Kirkpatrick, D. G., "Efficient Computation of Continuous Skeletons," *20th FOCS*, pp. 18-27 (1979).

[Kir84]. Kirkpatrick, D. G., *Private communication*, October 1984.

[LD81]. Lee, D. T. and Drysdale, R. L. III, "Generalization of Voronoi diagrams in the plane," *SIAM J. Comp.* **10** pp. 73-87 (1981).

[Lee82]. Lee, D. T., "Medial Axis Transformation of a Planar Shape," *IEEE Trans. Pattern Analysis and Machine Intelligence* **Vol. PAMI-4**(No. 4) pp. 363-369 (1982).

[Mor81]. Moravec, H., *Robot Rover Visual Navigation*, UMI Research Press (1981). (PhD Thesis, Stanford)

[OY82]. Ó 'Dúnlaing, C. and Yap, C. K., "A retraction method for planning the motion of a disc," *J. Algorithms (to appear)*, (1982).

[OSY83]. Ó 'Dúnlaing, C., Sharir, M., and Yap, C. K., "Retraction: A new approach to motion planning," *15th FOCS*, pp. 207-220 (1983).

[OSY84a]. Ó 'Dúnlaing, C., Sharir, M., and Yap, C. K., "Generalized Voronoi diagrams I: topological analysis of the diagram used in motion-planning for a ladder," Courant Institute - Robotics Laboratory Tech. Report (1984).

[OSY84b]. Ó 'Dúnlaing, C., Sharir, M., and Yap, C. K., "Generalized Voronoi diagrams II: Efficient construction of the diagram," Courant Institute - Robotics Laboratory Tech. Report (1984).

[Pre77]. Preparata, F. P., "The Medial Axis of a Simple Polygon," *Proc. 6th Symp. Math. Foundations of Comp. Sci.*, pp. 443-450 (1977).

[Sei81]. Seidel, R., "A Convex Hull Algorithm Optimal for Point Sets in Even Dimensions," University of British Columbia (1981). (M.Sc. Thesis)

[SH75]. Shamos, M. I. and Hoey, D., "Closest Point Problems," *16th FOCS*, pp. 151-162 (1975).

[Sha83]. Sharir, M., "Intersection and Closest-pair Problems for a set of Circular Discs," Courant Institute - NYU Technical Report No. 67 (Feb. 1983). (to appear SIAM J. Computing)

[Sha84]. Sharir, M., *Private communication*, October 1984.

[SL85]. Sharir, M. and Leven, D., "Intersection Problems and Applications of Voronoi Diagrams," in *Advances in Robotics, Vol. 1: Algorithmic and Geometric Aspects of Robotics*, ed. J. Schwartz and C. K. Yap, Lawrence Erlbaum Associates Inc. (to appear).

[WPB]. Widlund, O., Peskin, C., and Borgers, C., *Private communication*, (1984).

NYU COMPSCI TR-161 c.2
Yap, Chee K
An O(n log n)
algorithm for the

DATE DUE	BORROWER'S NAME
8·12·88	ILL – U of Mich
7·25·93	ILL – U of Delaware