Voronoi diagrams and their construction using plane sweep

Michiel Smid*

October 21, 2003

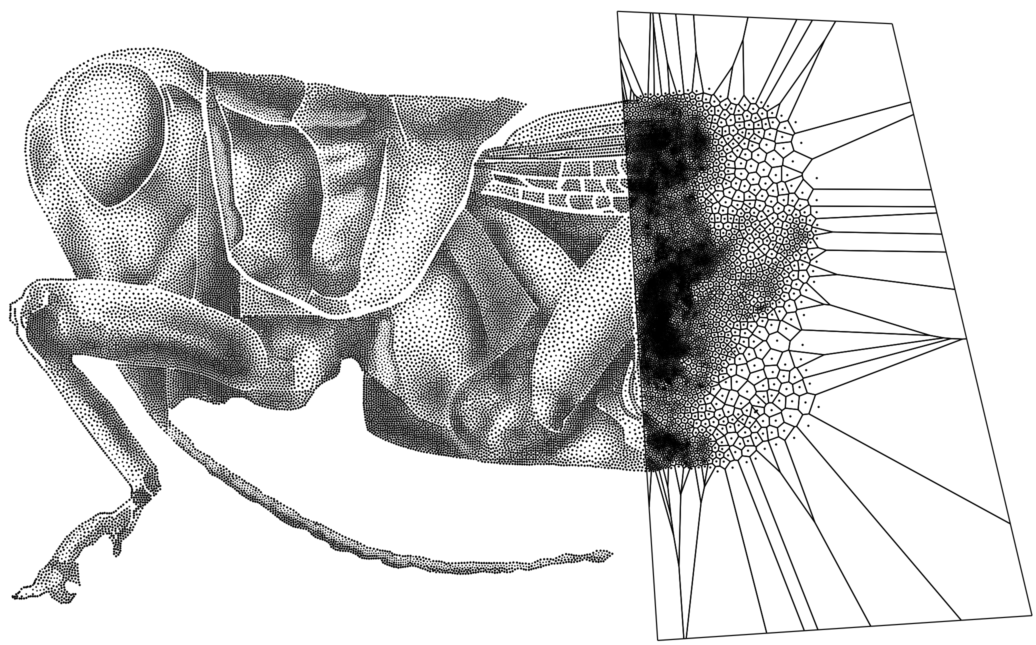

Figure 1: *A grasshopper. This image was generated using Voronoi diagrams. Source: Stefan Hiller, Relaxierte Punktverteilungen in der Computergrafik. Master's Thesis, University of Magdeburg, Germany, 1999.*

*School of Computer Science, Carleton University, Ottawa, Ontario, Canada K1S 5B6. E-mail: michiel@scs.carleton.ca.

1 Introduction

In these notes, we consider one of the most important data structures in computational geometry, the *Voronoi diagram*. These diagrams arise naturally in geometric proximity (or closest point) problems.

Throughout these notes, we will denote the Euclidean distance between two points $p = (p_1, p_2)$ and $q = (q_1, q_2)$ by $d(p, q)$, i.e.,

$$d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2}.$$

In order to show how Voronoi diagrams arise, we consider the following problem.

The post-office problem: Let S be a set of n points in the plane. We want to store these points in a data structure such that for any given query point q, we can efficiently compute a *nearest neighbor* of q in S, i.e., a point $p \in S$ that is closest to q,

$$d(p, q) = \min\{d(r, q) : r \in S\}.$$

This problem was introduced to the algorithms community in 1973 by Knuth. He imagines each point of S a post-office. Then a query specifies a point in the plane and we want to find a post-office that is closest to this point.

The complexity of any data structure solving this problem will be expressed by the following three measures, which are functions of n, the size of the set S.

- *Preprocessing time:* the time needed to build the data structure.
- *Size:* the amount of space needed to store the data structure.
- *Query time:* the time needed to answer a query, i.e., to find a nearest neighbor of the query point.

We assume that the set S is fixed and that there are a large number of queries, so that it is worth to spend time for building a data structure. (If there is only one query, then the best approach is to find a nearest neighbor using linear search. In this case, it is a waste of time to build a data structure.)

We can solve the post-office problem using the so-called *locus approach*. The basic idea of this approach is as follows.

1. We partition the plane into regions and associate with each region R a point p_R of S such that the following holds. If a query point q is contained in the region R, then p_R is a nearest neighbor of q, independent of the exact location of q in R.

2. Given this partition and given any query point q, we compute the region R that contains q and report the point p_R as its nearest neighbor.

As we will see, there is a natural partition of the plane into regions that satisfies condition 1. above. This partition is the Voronoi diagram, named after the Russian mathematician who used them in 1908 in a paper on quadratic forms. These diagrams appear outside mathematics and computer science under different names. For example, in geography they are called Thiessen polygons, in pattern recognition the name Blum transform is used, whereas metallurgists call them Wigner-Seitz zones.

In these notes, we will define Voronoi diagrams, prove some of their basic properties, and show how to construct them using the plane sweep technique, in $O(n \log n)$ time.

The problem of computing the region R containing any given query point q is called the *point location problem*. We have seen already how this problem can be solved.

2 Definition of the Voronoi diagram

Consider again the finite set S of points in the plane. The Voronoi diagram of S is a partition of the plane into *Voronoi regions*, one region $VR(p)$ for each point p of S. Such a region $VR(p)$ is defined as the set of all points $q \in \mathbb{R}^2$ that have p as a nearest neighbor. That is, for each point p of S, its Voronoi region $VR(p)$ is defined as

$$VR(p) := \{q \in \mathbb{R}^2 : d(p,q) \leq d(r,q) \text{ for all } r \in S\}.$$

To get some insight into this notion, assume that the set S contains only two points, a and b. How does the Voronoi region of a look like? Let ℓ be the *bisector* of a and b, which is the line orthogonal to the line segment with endpoints a and b, and that cuts this segment into two segments having the same length. This bisector divides the plane into two halfplanes. Let $H(a,b)$ be the halfplane containing a. Then any point q in $H(a,b)$ has a

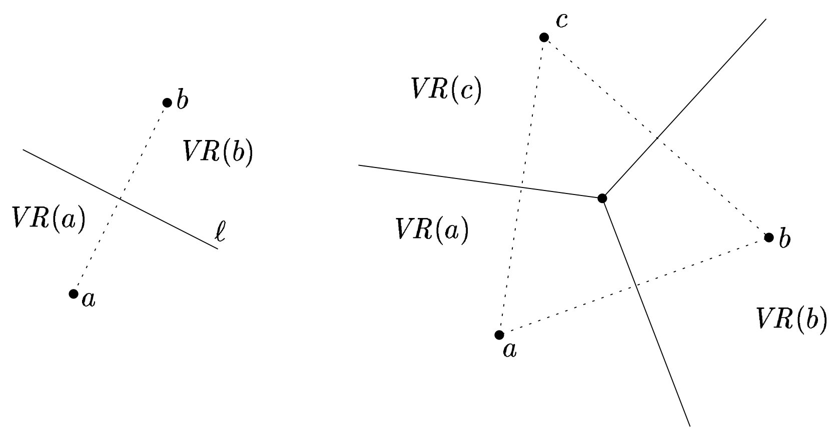

Figure 2: *The Voronoi regions for the sets $S = \{a, b\}$ and $S = \{a, b, c\}$. These regions are bounded by the solid lines.*

---

as its nearest neighbor, and we have $VR(a) = H(a, b)$. Similarly, the other halfplane $H(b, a)$, the one containing b, is the Voronoi region of b. Observe that the bisector ℓ belongs to both Voronoi regions, because each point q on ℓ has both a and b as a nearest neighbor. See Figure 2, in which also an example for a set of three points a, b, and c is given.

In general, let S be any finite set of points in the plane. For any two distinct points p and r of S, we define $H(p, r)$ to be the halfplane defined by the bisector of p and r that contains p. Observe that

$$H(p, r) = \{q \in \mathbb{R}^2 : d(p, q) \leq d(r, q)\}.$$

Lemma 1 *For each point $p \in S$, we have*

$$VR(p) = \bigcap_{r \in S \setminus \{p\}} H(p, r).$$

Exercise 1 Prove Lemma 1.

Lemma 2 *For each point $p \in S$, its Voronoi region $VR(p)$ is a (possibly unbounded) non-empty convex polygon.*

Proof: Since the Voronoi region of p contains the point p, it is non-empty. By Lemma 1, $VR(p)$ is the intersection of halfplanes, which shows that it is a (possibly unbounded) polygon. Since each halfplane is convex, it also follows that $VR(p)$ is convex. ∎

Definition 1 Let S be a finite set of points in the plane.

1. The *Voronoi diagram* $VD(S)$ of S is defined as the partition of the plane induced by all Voronoi regions $VR(p)$ with $p \in S$.

2. If two Voronoi regions have an intersection of positive length, then we call this intersection a *Voronoi edge*.

3. If two Voronoi edges intersect in a single point, then we call this point a *Voronoi vertex*.

In Figure 3, an example is given. Observe that Voronoi edges can be unbounded; see also Exercise 2.

Exercise 2 Let S consist of the four corners of a square. Determine the Voronoi diagram of S. How many Voronoi vertices and edges does the diagram have? Answer the same questions for (i) a set of n points that are all on a circle, and (ii) a set of n points that are all on a straight line.

The main problem that we will consider in these notes is to design an efficient algorithm that, when given any set S of n points in the plane, constructs its Voronoi diagram. Here, constructing a Voronoi diagram means computing its Voronoi vertices and edges, together with the incidence relations among them.

We can use the characterization of Lemma 1 to construct the Voronoi diagram as follows. For each point p of S, we compute the intersection of the halfplanes $H(p,r)$, $r \in S \setminus \{p\}$. We have seen in the notes on convex hulls that this can be done in $O(n \log n)$ time for each point p. This gives us all Voronoi regions. Then, the Voronoi diagram is obtained by putting all these regions together. Overall, we get a running time of $O(n^2 \log n)$.

Later in these notes, we will give an algorithm that constructs the Voronoi diagram much faster, in $O(n \log n)$ time. This algorithm uses the plane sweep technique in a non-trivial way. Before we go to the algorithmic aspects of Voronoi diagrams, however, we prove some of its basic properties.

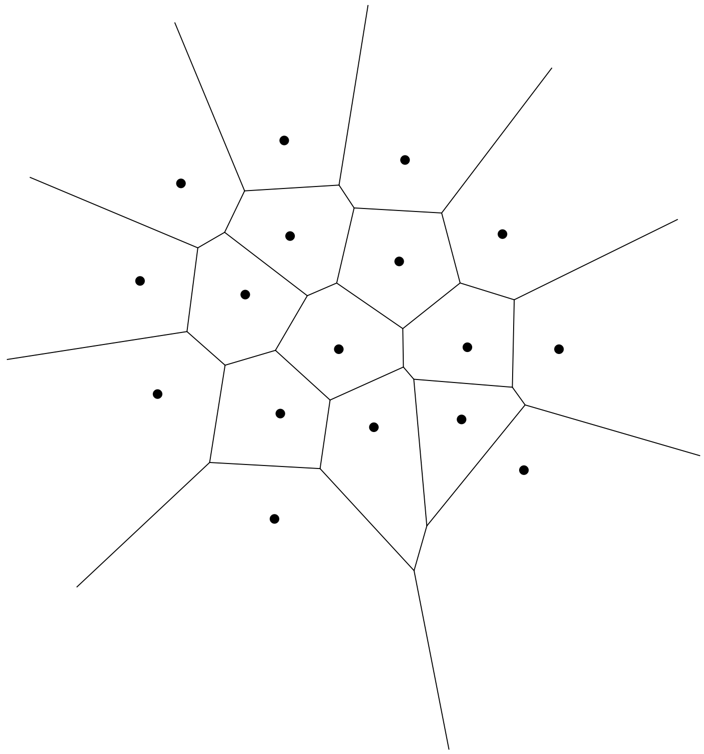

Figure 3: *An example of a Voronoi diagram.*

3 Some properties of Voronoi diagrams

Observation 1 *Each Voronoi edge is contained in the bisector of two points of S and is incident to exactly two Voronoi regions.*

Lemma 3 *The degree of each Voronoi vertex is greater than or equal to three.*

Proof: Let v be an arbitrary Voronoi vertex. It follows from Definition 1 that the degree of v is at least equal to two. Assume that this degree is equal to two. We will derive a contradiction.

Let e and e' be the two Voronoi edges that have v as endpoint. We distinguish two cases.

First assume that e and e' are not collinear; see the left part of Figure 4. By Observation 1, e is incident to exactly two Voronoi regions, and the same

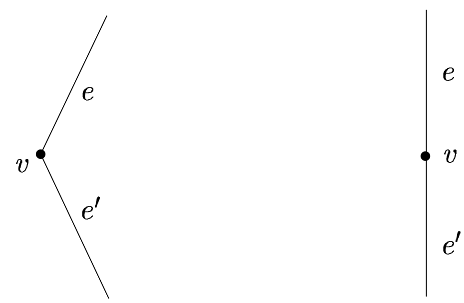

Figure 4: *Illustrating the proof of Lemma 3.*

---

is true for e'. Since v has degree two, e and e' are incident to the same Voronoi regions. But then the fact that e and e' are not collinear implies that one of these regions must be non-convex. This is a contradiction.

The second case is when e and e' are collinear; see the right part of Figure 4. Let p and q be the points of S such that e and e' are both incident to the Voronoi regions $VR(p)$ and $VR(q)$. Then e and e' are both contained in the bisector of p and q. But then e and e' together form one Voronoi edge and, therefore, the Voronoi vertex v does not exist. This is again a contradiction. ∎

Let v be any Voronoi vertex and let $p_1, p_2, \ldots, p_m$ be the points of S such that the Voronoi regions $VR(p_i)$, $1 \leq i \leq m$, have v as their common intersection. Then, m is the degree of v in $VD(S)$, which, by Lemma 3, is greater than or equal to three. The definition of Voronoi region implies that all distances $d(p_i, v)$, $1 \leq i \leq m$, are equal. Let $C(v)$ denote the circle with center v and radius $d(p_1, v)$.

Lemma 4 *The circle $C(v)$ does not contain any point of S in its interior.*

Proof: Assume $C(v)$ contains a point x of S in its interior. Then $d(x, v) < d(p_i, v)$ for all i with $1 \leq i \leq m$, which implies that v is not contained in any of the Voronoi regions $VR(p_i)$. This is a contradiction. ∎

As we have seen already, some Voronoi edges are bounded, whereas some are unbounded. The following lemma explains the difference.

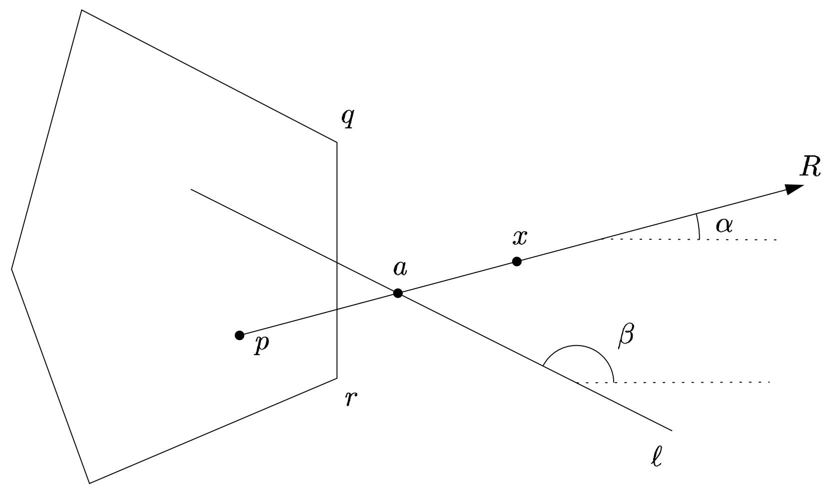

Figure 5: *Illustrating the proof of Lemma 5. We assume that $VR(p)$ is unbounded and that p is in the interior of the convex hull.*

Lemma 5 *Let S be a finite set of points in the plane and let $p \in S$. The Voronoi region of p is unbounded if and only if p is on the boundary of the convex hull of S.*

Proof: First assume that $VR(p)$ is unbounded. We will show that p is on the boundary of the convex hull of S. Assume p is not on this boundary. Since $VR(p)$ is unbounded and convex, there is a ray R starting in p that is completely contained in $VR(p)$. Also, since the convex hull of S is bounded, this ray intersects its boundary in, say, the hull edge with endpoints q and r. Observe that both q and r belong to S. Assume without loss of generality that this hull edge is vertical and that q is above r. Also, assume without loss of generality that the angle α between R and the horizontal is non-negative. Observe that $\alpha < \pi/2$; see Figure 5. We claim that there is a point x on R that is closer to q than to p. This will be a contradiction, because R and, hence, also x, is contained in $VR(p)$.

So it remains to find the point x. The bisector ℓ of p and q makes an angle β with the horizontal which satisfies $\pi/2 \leq \beta < \pi$. Therefore, R and ℓ intersect in, say, point a. Any point x on R that is to the right of a belongs to the halfplane $H(q,p)$, i.e., it is closer to q than to p. This proves one direction of the claim.

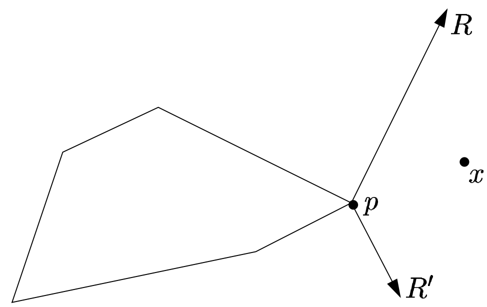

Figure 6: *Illustrating the proof of Lemma 5. We assume that p is on the boundary of the convex hull.*

To prove the converse, assume that p is on the boundary of the convex hull of S. If p is a vertex of the convex hull, then let R and R' be the rays starting in p that are orthogonal to the hull edges incident to p and that are in the exterior of the convex hull; see Figure 6. Otherwise, p is in the interior of a convex hull edge, and we let both R and R' be equal to the ray starting in p that is orthogonal to this hull edge and that is in the exterior of the convex hull.

We claim that each point x on or between R and R' is closer to p than to any other point of S. That is, each such point x is contained in $VR(p)$, which will prove that $VR(p)$ is unbounded. This claim follows from the fact that for each point q in $S \setminus \{p\}$, point x belongs to the halfplane $H(p, q)$, i.e., x is closer to p than to q. ∎

It is a priori not clear how large a Voronoi diagram can be. That is, given a set of n points, how many Voronoi vertices and edges does its Voronoi diagram have? The tool to answer this question is Euler's formula for planar graphs.

Theorem 1 (Euler) *Consider any embedding of a connected planar graph G without edge crossings. Let V, E, and F be the number of vertices, edges, and faces (including the single unbounded face) of this embedding, respectively. Then*

$$V - E + F = 2.$$

This theorem has the following important corollary.

Corollary 1 *Let G be a planar graph with V vertices and assume that $V \geq 3$. Then*

1. *G has at most $3V - 6$ edges, and*

2. *any embedding of G has at most $2V - 4$ faces (including the unbounded face).*

How can we use these results on planar graphs to give an upper bound on the size of a Voronoi diagram? Observe that in an embedding of a planar graph, each edge is *bounded* or, equivalently, has two vertices as endpoints. Euler's formula and its corollary only apply to such embeddings. But, as we have seen already, some of the edges of a Voronoi diagram are unbounded. Also, Corollary 1 bounds the number of edges and faces in terms of the number of vertices. For a Voronoi diagram, we only know at this moment that it has exactly n faces; we want to determine the number of its vertices and edges. Therefore, we proceed as follows.

Let S be a set of n points in the plane and consider its Voronoi diagram $VD(S)$. We define the *dual graph* G of $VD(S)$ as follows. This graph G has the points of S as its vertices. Any two such vertices p and q, with $p \neq q$, are connected by an edge in G if and only if the Voronoi regions $VR(p)$ and $VR(q)$ share a Voronoi edge. In Figure 7, an example is given.

By Lemma 2, the number of Voronoi regions is equal to n. Moreover, G has exactly n vertices. We show how to use this to bound the number of edges of the Voronoi diagram.

Exercise 3 Prove that G is a planar connected graph.

Let E be the number of edges of the graph G. Then, by Corollary 1, we have $E \leq 3n - 6$. Since Voronoi regions are convex, any two distinct Voronoi regions can have at most one Voronoi edge in common. Hence, there is a one-to-one correspondence between the edges of G and the Voronoi edges. It follows that the Voronoi diagram of S has E Voronoi edges, which is less than or equal to $3n - 6$.

It remains to bound the number of Voronoi vertices. For any Voronoi vertex v, let $deg(v)$ denote its degree, i.e., the number of Voronoi edges incident to v. If we take the sum of the degrees of all Voronoi vertices, then we count each bounded Voronoi edge twice and each unbounded Voronoi edge once. This implies that $\sum_v deg(v) \leq 2E$. Let V denote the number

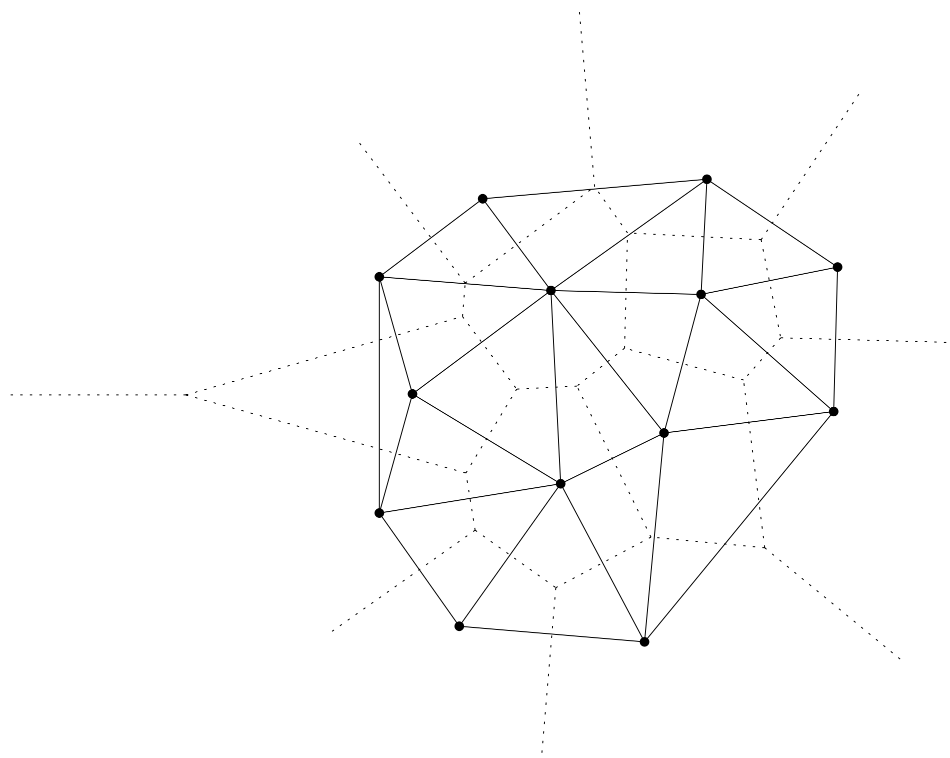

Figure 7: *The dashed lines represent the Voronoi edges, whereas the solid lines represent the edges of the dual graph G.*

---

of Voronoi vertices. We know from Lemma 3 that $deg(v) \geq 3$ for each Voronoi vertex. Therefore, the summation $\sum_v deg(v)$ is at least equal to $3V$, which implies that $3V \leq 2E$. We saw already that $E \leq 3n - 6$. Therefore, $V \leq 2E/3 \leq 2n - 4$. We have proved the following result.

Theorem 2 *The Voronoi diagram of a set S of n points in the plane consists of*

1. *exactly n Voronoi regions,*
2. *at most $2n - 4$ Voronoi vertices, and*
3. *at most $3n - 6$ Voronoi edges.*

The Voronoi diagram contains much information about the distance distribution of the point set S. By this theorem, the entire diagram can be stored in only $O(n)$ space.

We conclude this section with some remarks about the graph G. In the example in Figure 7, each bounded face of G is a triangle. This is true for any point set S having the property that (i) no four (or more) points are cocircular, and (ii) no three (or more) points are collinear. In this case, G is a *triangulation* of S; it is a partition of the convex hull of S into pairwise disjoint triangles. A set of points can have many triangulations. The triangulation G, however, has many interesting properties, and it is called the *Delaunay triangulation*, named after another Russian mathematician, who introduced it in 1934.

Exercise 4 Determine the graph G for each of the three point sets of Exercise 2.

Exercise 5 Let S be a finite set of points in the plane, such that no four (or more) points are cocircular, and no three (or more) points are collinear. Let G be the graph having S as its vertex set, and in which any three distinct vertices p, q, and r form a triangle in G if and only if the circle through p, q, and r does not contain any point of S in its interior. Prove that G is the Delaunay triangulation of S. (This is the way Delaunay defined his triangulation.)

Remark 1 In order to get a better understanding of Voronoi diagrams and Delaunay triangulations, you should surf to the following web page:

`http://wwwpi6.fernuni-hagen.de/java/anja/`

This page presents VoroGlide, an interactive program developed by the group of Rolf Klein at the FernUniversität Hagen in Germany. You can insert, delete, and move points, and watch how the Voronoi diagram and Delaunay triangulation change.

Exercise 6 Let S be a finite set of points in the plane, let $p \in S$, and let q be a nearest neighbor of p in $S \setminus \{p\}$. Prove that the Voronoi regions $VR(p)$ and $VR(q)$ share an edge.

Exercise 7 Let S be a set of n points in the plane and assume we have constructed the Voronoi diagram of S already. Prove that we can, in $O(n)$ time, solve the *all-nearest-neighbors problem*, in which we have to compute for each point p of S, a nearest neighbor of p in $S \setminus \{p\}$.

Exercise 8 Let S be a finite set of points in the plane. Prove that the Delaunay triangulation of S contains a minimum spanning tree of S.

4 Fortune's plane sweep algorithm for constructing Voronoi diagrams

In this section, we will give an algorithm that constructs the Voronoi diagram $VD(S)$ of a given set S of n points in the plane. The algorithm is due to Fortune (1987) and uses plane sweep in a non-trivial way. The main purpose of this section is to explain how the plane sweep technique can be adapted so that the Voronoi diagram can be constructed efficiently. It turns out that for "degenerate" point sets, some "technical" problems, that have nothing to do with the plane sweep method itself, have to be solved. Therefore, we will assume that the set S is in "general" position:

Assumption 1 Throughout this section, S denotes a set of n points in the plane. This set has the property that (i) no three (or more) points lie on a straight line, and (ii) no four (or more) points lie on a circle.

Exercise 9 Prove that Assumption 1 implies that the degree of every Voronoi vertex of $VD(S)$ is equal to three.

This exercise shows that each Voronoi vertex v is defined by exactly three points of S. The circle through these three points does not contain any point of S in its interior and has v as its center; see also Lemma 4.

Our goal is to design a plane sweep algorithm that constructs the Voronoi diagram $VD(S)$. Let us recall what this means. We move a vertical sweep line SL from left to right over the set S. During the sweep, we would like to maintain the invariant that the part of $VD(S)$ that is to the left of SL has been constructed already. Also, we would like to maintain this invariant by only using information obtained from points that are to the left of SL; we do not want to use any information that is implied by the points to the right of SL. Unfortunately, there is a problem here: The Voronoi region $VR(p)$ of any point p of S starts to the left of p. Hence, the sweep line reaches the left boundary of $VR(p)$ before it reaches p. That is, some points that are to the right of the sweep line contribute to the part of the Voronoi diagram that is to the left of this line.

This shows that we cannot apply the plane sweep technique in the standard way. Instead, let us try to maintain a weaker invariant. Let VD_{SL} denote the part of the Voronoi diagram that is to the left of the sweep line SL.

Question 1 *Is there a part VD'_{SL} of VD_{SL} that is uniquely determined by those points of S that are to the left of SL?*

If this question has a positive answer, then we could try to maintain the invariant that at any moment during the sweep, the part VD'_{SL} has been constructed already. The following lemma states that there is indeed a positive answer. Before we state this lemma, we introduce the following notation.

If p is a point in the plane, then $d(p, SL)$ denotes the shortest distance from p to any point on the sweep line SL. This is of course the horizontal distance between p and SL. The set of all points of S that are to the left of SL will be denoted by S_{SL}.

Lemma 6 *Let*

$$L := \{x \in \mathbb{R}^2 \,|\, \exists q \in S_{SL} : d(x,q) < d(x, SL)\}.$$

The set L is completely to the left of SL. The part of the Voronoi diagram of S that is in the region L does not depend on any point of $S \setminus S_{SL}$. In other words, this part of the Voronoi diagram is completely determined by the points of S_{SL}.

Proof: It is clear that L lies completely to the left of SL. To prove the other claim, let $x \in L$ and let p be a point of S such that $x \in VR(p)$. We have to show that $p \in S_{SL}$. If we have shown this, then it follows that $VD(S) \cap L$ is completely determined by the points of S_{SL}.

Assume that $p \in S \setminus S_{SL}$. By the definition of L, there is a point $q \in S_{SL}$ such that $d(x,q) < d(x, SL)$. Since $x \in VR(p)$, we have $d(x,p) \leq d(x,q)$. Also, since x is to the left of SL and p is on or to the right of SL, we have

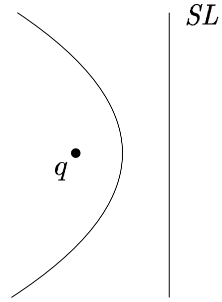

Figure 8: *The set of all points x such that $d(x, q) = d(x, SL)$ is a parabola.*

$d(x, SL) \leq d(x, p)$. Combining these inequalities shows that $d(x, q) < d(x, q)$, which is a contradiction. Therefore, $p \in S_{SL}$. ∎

Let us find out how the set L looks like. Assume that the set S_{SL} contains only one point, say q. Then

$$L = \{x \in \mathbb{R}^2 | d(x, q) < d(x, SL)\}.$$

The boundary of L consists of all points x such that $d(x, q) = d(x, SL)$. We claim that this boundary is a parabola. To prove this, assume that the sweep line has equation $x_1 = s$ and let q be given by $q = (a, b)$. Observe that $a < s$. Then $d(x, q) = ((x_1 - a)^2 + (x_2 - b)^2)^{1/2}$ and $d(x, SL) = |s - x_1|$. Hence, a point x is on the boundary of L if and only if $(x_1 - a)^2 + (x_2 - b)^2 = (s - x_1)^2$, which can be rewritten as $2(s - a)x_1 = -x_2^2 + 2bx_2 - a^2 - b^2 + s^2$. The latter equation is clearly that of a parabola in the x_1x_2-plane. Its axis is the horizontal line through q and its extremum is on this axis, in the middle of the horizontal segment connecting q with SL; see Figure 8. The set L itself is to the left of this parabola, i.e.,

$$L = \{x \in \mathbb{R}^2 | 2(s - a)x_1 < -x_2^2 + 2bx_2 - a^2 - b^2 + s^2\}.$$

We call this set the *interior* of the parabola.

In general, if the set S_{SL} is non-empty, the region L is the union of the interiors of $|S_{SL}|$ such parabolas, one parabola for each $q \in S_{SL}$. The boundary of L, which we call *beach line* (aka *wave front*), consists of a sequence of parabola segments; see Figure 9.

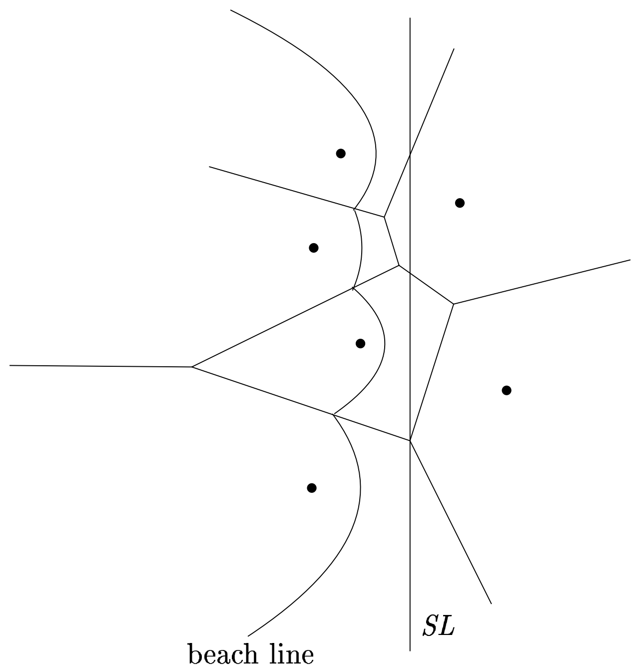

Figure 9: *The boundary of the region L is called beach line.*

Exercise 10 Prove that the beach line is connected.

Our algorithm will maintain this beach line of parabola segments in the Y-structure, which stores the segments sorted from bottom to top. Of course, these segments are not stored explicitly, because they change as soon as the sweep line moves. Instead, we store them implicitly; the details will be given later. During the sweep, we maintain the following invariant.

Invariant: For any position of the sweep line SL, the part of the Voronoi diagram $VD(S)$ that belongs to the region L has been constructed already. More precisely, we have computed all Voronoi vertices of this part of $VD(S)$, and for each such vertex, we have computed its three defining points of S. (See Exercise 9 and the paragraph following it.)

When the sweep line SL moves from left to right, it "pulls" the beach line with it. If SL is at position $x_1 = \infty$, the invariant implies that we have computed all Voronoi vertices, together with their defining points. Using this information, the complete Voronoi diagram can easily be constructed.

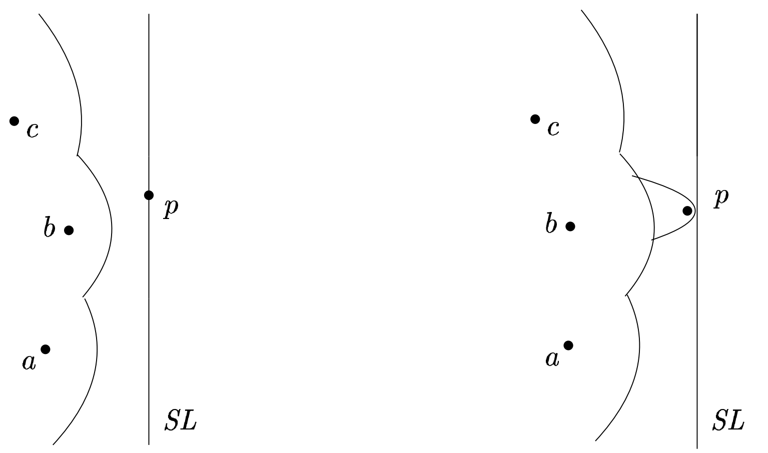

Figure 10: *If the sweep line reaches point p, then a new parabola segment appears on the beach line.*

4.1 The transition points of the sweep

At this moment, we know that the Y-structure (implicitly) stores the parabola segments of the beach line. Recall that the *transition points* of the sweep are those positions of the sweep line at which the combinatorial structure of the beach line changes. This happens if a new parabola segment becomes part of it or if a parabola segment disappears. We maintain (a subset of) the transition points in the X-structure. It turns out that there are two types of transition points. We consider each type separately.

Type 1: The points of S are transition points. We call these *site events*, because the points of S are often called sites.

Let p be a point of S. Consider what happens when the sweep line SL reaches p. Refer to Figure 10. Assume that the horizontal line through p intersects b's parabola segment in its interior.

If the sweep line is at point p, then p's complete parabola is the set $\{x | x_1 \leq p_1 \text{ and } d(x,p) = d(x, SL)\}$, which is that part of the horizontal line through p that is to the left of p. If SL moves to the right, this degenerate parabola opens itself, becomes wider and wider, and its extremum moves to the right. We see that a new parabola segment, determined by p, appears on the beach line. Also, the "middle" part of the parabola segment determined by point b disappears from the beach line.

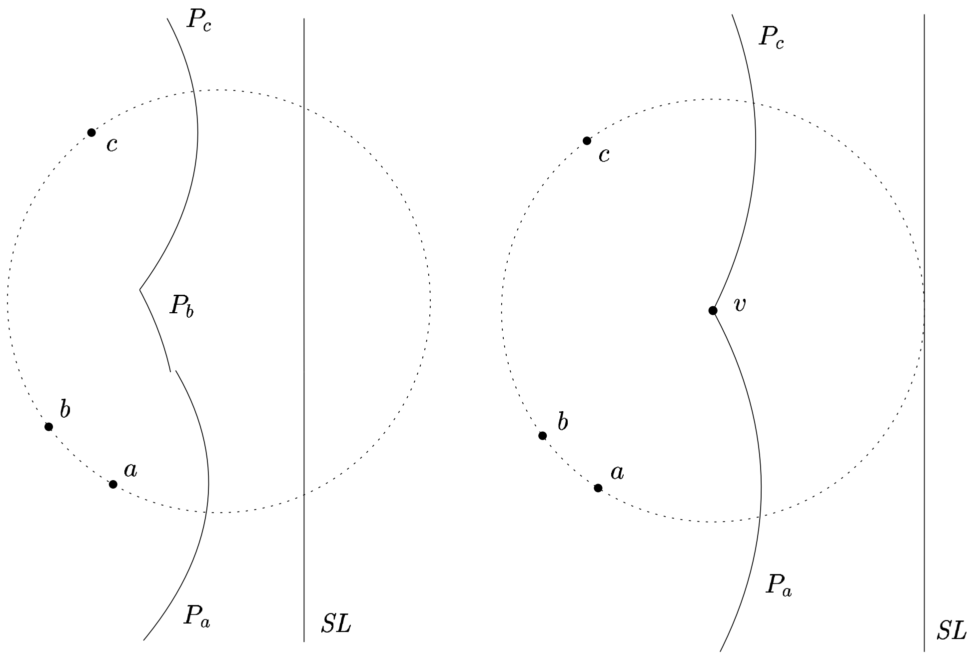

Figure 11: *The parabola segment P_b disappears from the beach line if the sweep line reaches the rightmost point of the circle through a, b, and c.*

Type 2: The positions where a parabola segment disappears from the beach line are transition points. We call these *circle events*.

To illustrate this, refer to Figure 11. Consider three consecutive parabola segments P_a, P_b, and P_c, as indicated in the left figure. Here, a, b, and c are three points of S. If the sweep line moves to the right, the three parabola segments open themselves and their extrema move to the right. The segment P_b disappears from the beach line at the moment when P_a, P_b, and P_c have one point in common. Let us see when this happens. For any point v on the beach line, and any $z \in \{a, b, c\}$, we have

$$v \text{ is on } P_z \text{ if and only if } d(v, z) = d(v, SL). \tag{1}$$

Therefore, P_a, P_b, and P_c have a common intersection in, say v, if and only if

$$d(v, a) = d(v, b) = d(v, c) = d(v, SL). \tag{2}$$

Let $C(a, b, c)$ be the circle through a, b, and c. Then (2) holds if the sweep line SL contains the rightmost point of this circle. Hence, if SL is at this position, the parabola segment P_b disappears from the beach line. It follows from the proof of Lemma 8 below that $C(a, b, c)$ does not contain any point of S in its interior. Therefore, at this position of SL, we have found a new Voronoi vertex: the center v of $C(a, b, c)$.

It follows that circle events are characterized as follows. They are the (x-coordinates of the) rightmost points of circles $C(a, b, c)$, for any triple P_a, P_b, and P_c of parabola segments that are consecutive on the beach line.

The basic structure of the sweep algorithm should be clear. The sweep line moves from one transition point to the next one. At each such point, the beach line and the set of transition points are updated. Before we give more details, let us prove two lemmas that show that the algorithm indeed constructs the Voronoi diagram. These lemmas have the following interpretation. Assume we sweep SL from left to right and during the sweep, draw the trajectories of the common endpoints of all pairs of parabola segments that are neighbors on the beach line. Lemma 7 states that in this way, we draw the complete Voronoi diagram of the points of S, whereas Lemma 8 states that we only draw this diagram.

Lemma 7 *Let e be an edge of the Voronoi diagram $VD(S)$ and let u be an arbitrary point on e. (u may be an endpoint of e, in which case it is a Voronoi vertex.) There is a position of the sweep line such that u is the common endpoint of two parabola segments that are neighbors on the beach line.*

Proof: Let p and q be the points of S such that the Voronoi regions $VR(p)$ and $VR(q)$ share the edge e. Then, by the definition of Voronoi region, we have $d(u, p) = d(u, q)$. Consider the circle C centered at u that contains p and q. Since u belongs to both $VR(p)$ and $VR(q)$, this circle does not contain any points of S in its interior; see Figure 12.

Let s be the x-coordinate of the rightmost point of C. We claim that s is the position of the sweep line we are looking for. To prove this, consider the moment when the sweep line is at position s. Then $d(u, p) = d(u, SL)$ and $d(u, q) = d(u, SL)$. Hence, point u either belongs to our set L or is on the boundary of L.

We first prove by contradiction that u is on L's boundary, i.e., it is on the beach line. So, assume u is not on the beach line. Then $u \in L$ and, by the

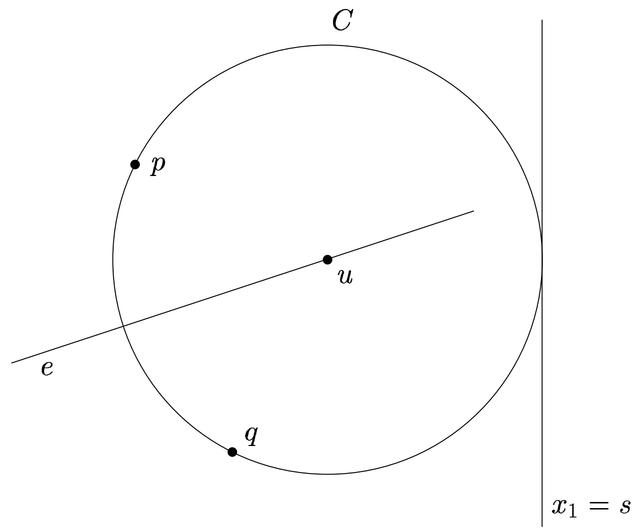

Figure 12: *Illustrating the proof of Lemma 7. When the sweep line is at position s, u is the common endpoint of the parabola segments of p and q.*

---

definition of L, there is a point r in S_{SL} such that $d(u, r) < d(u, SL)$. But then, r is in the interior of the circle C, which is a contradiction.

Hence, we know that u is on the beach line. Since $d(u, p) = d(u, q) = d(u, SL)$, it follows from (1) that u is on the parabola segments of both p and q. These two segments must be neighbors on the beach line. ■

Lemma 8 *Consider any position of the sweep line. The common endpoint of any two parabola segments that are neighbors on the beach line lies on some Voronoi edge.*

Proof: Let x be the common endpoint of two neighboring parabola segments, corresponding to, say points p and q, as in the left part of Figure 13. Then $d(x, p) = d(x, q) = d(x, SL)$. Let C be the circle with center x and going through p and q. We claim that x belongs to the intersection $VR(p) \cap VR(q)$, which implies that x lies on a Voronoi edge. (x may be an endpoint of a Voronoi edge, in which case it is a Voronoi vertex.) To prove this claim, it suffices to show that C does not contain any point of S in its interior. This is easily proved by contradiction. Assume point r of S is in the interior of C. Then $d(x, r) < d(x, p) = d(x, SL)$ and r is to the left of the sweep line.

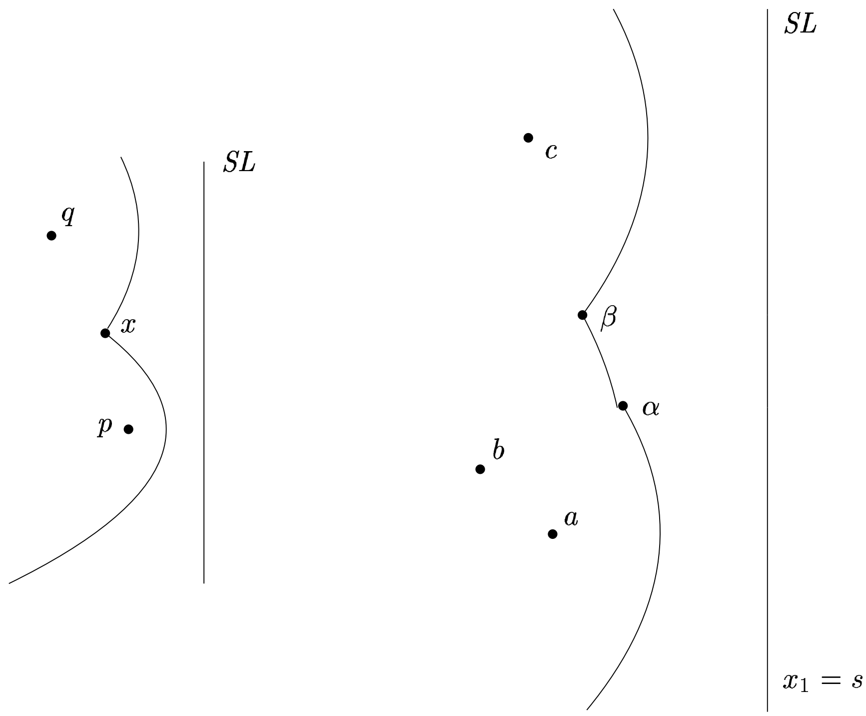

Figure 13: *Illustrating the proof of Lemma 8 (left figure), and how to compute the parabola segment $P(a, b, c)$ (right figure).*

Therefore, by the definition of L, we have $x \in L$. This is a contradiction, because x is on the boundary of L, which does not belong to L itself. ∎

4.2 Some implementation details

We have mentioned already that the Y-structure will contain an implicit representation of the beach line. How does this representation look like?

A parabola segment of the beach line is defined by three points, say a, b, and c, of S, and the position of the sweep line. Let s be the x-coordinate of this position. Assume that a's parabola segment is below b's segment and b's segment is below c's segment. Observe that the points a, b, and c can have a different ordering in the y-direction. In fact, a can be equal to c. The complete parabola of b has equation $2(s - b_2)x_1 = -x_2^2 + 2b_2x_2 - b_1^2 - b_2^2 + s^2$; the parabolas for a and c have similar equations.

The parabola segment of b can be computed from a, b, c, and s, as follows. (Refer to Figure 13.)

1. Write down the equations for the parabolas of a, b, and c.
2. Compute the appropriate intersection α of the parabolas of a and b.
3. Compute the appropriate intersection β of the parabolas of b and c.
4. The parabola segment of b is the part of b's parabola that is between α and β.

We denote the resulting parabola segment of b by $P(a, b, c)$.

The Y-structure is a balanced binary search tree which stores, for any position of the sweep line, the parabola segments of the beach line, sorted from bottom to top. To store a parabola segment $P(a, b, c)$ at a node of this tree, we store the triple of points (a, b, c). We just saw that this information suffices to compute $P(a, b, c)$, for any given position of the sweep line, in constant time.

The X-structure is also a balanced binary search tree and it stores the following information. First, it stores (the x-coordinates of) all points of S that are to the right of the sweep line. Second, for each parabola segment $P(a, b, c)$ that is on the beach line and for which $a \neq c$, we store its circle event in the X-structure, provided this event is to the right of the sweep line. Recall that this circle event is the (x-coordinate of the) rightmost point of the circle through a, b, and c. Of course, the elements of the X-structure are stored in sorted order from left to right.

Now we can describe in more detail what happens if the sweep line moves from one transition point to the next one.

Case 1: The next transition point is a site event, i.e., the sweep line encounters a point, say p, of S.

We do the following; refer to Figure 10. First, we search in the Y-structure for the parabola segment $P(a, b, c)$ that is intersected by the horizontal line ℓ through p. This search starts in the root of the Y-structure and follows a path down the tree. At each node on this path, we compute its parabola segment P. Given P, we decide if (i) ℓ intersects P, (ii) ℓ is below P, or (iii) ℓ is above P. If (i) holds, then $P = P(a, b, c)$, and the search terminates. If (ii) holds, then the search proceeds to the left child of the current node. Otherwise, if (iii) holds, the search proceeds to the right child. (Here, we assume for simplicity that the line ℓ intersects the parabola segment $P(a,b,c)$ in its interior; see Exercise 11 below.)

Having found $P(a,b,c)$, we delete it from the Y-structure, and, if $a \neq c$, delete its circle event from the X-structure. Then we insert the new parabola segments $P(a,b,p)$, $P(b,p,b)$, and $P(p,b,c)$ into the Y-structure. Finally, we compute the circle events of $P(a,b,p)$ and $P(p,b,c)$, and insert those that are to the right of the sweep line into the X-structure.

The circle event of, say $P(a,b,p)$, is computed as follows. Compute the circle C through a, b, and p. Let $m = (m_1, m_2)$ and δ be the center and radius of C, respectively. Then the circle event of $P(a,b,p)$ has x-coordinate $m_1 + \delta$.

Case 2: The next transition point is a circle event, i.e., the sweep line encounters the rightmost point of the circle through three points, say a, b, and c. In this case, the parabola segment $P(a,b,c)$ disappears from the beach line.

We do the following; refer to Figure 14. First, we output the center v of the circle through a, b, and c as a Voronoi vertex. With v, we also output the points a, b, and c, because they define the three Voronoi edges that are incident to v. Then we search in the Y-structure for the successor parabola segment $P(b,c,d)$ of $P(a,b,c)$. Similarly, we search for the predecessor $P(e,a,b)$ of $P(a,b,c)$. We assume for simplicity that both the successor and the predecessor exist.

We delete $P(a,b,c)$ from the Y-structure and, if $b \neq d$ resp. $e \neq b$, delete the circle events of $P(b,c,d)$ and $P(e,a,b)$ from the X-structure. The node of the Y-structure representing $P(b,c,d)$ stores the triple (b,c,d). We replace this triple by (a,c,d), because $P(a,c,d)$ is a new parabola segment on the beach line. Similarly, in the node representing $P(e,a,b)$, we replace the triple (e,a,b) by (e,a,c).

Finally, if $a \neq d$, we compute the circle event of $P(a,c,d)$. If it is to the right of the sweep line, then we insert this event into the X-structure. If $e \neq c$, then we do the same for $P(e,a,c)$.

Exercise 11 Work out the details for Case 1 if the line ℓ intersects $P(a,b,c)$ in one of its endpoints. (*Hint:* In this case, we have found a new Voronoi vertex.)

The complete algorithm for computing the Voronoi vertices has a form that is similar to the Bentley-Ottmann algorithm for computing intersections

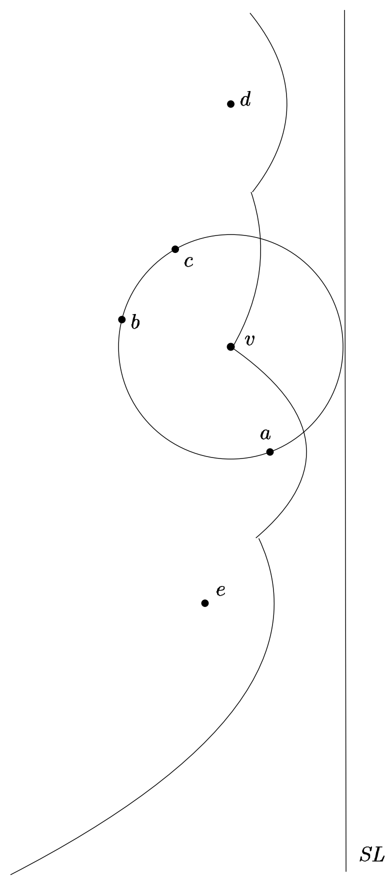

Figure 14: *How to handle a circle event.*

of line segments.

To initialize the algorithm, we store all points of S in the X-structure, sorted from left to right. At this moment, the Y-structure is empty, and the X-structure does not contain any circle event. After the initialization, there is a while-loop. In each iteration, we take the leftmost element in the X-structure, delete it, and process it as described above.

We analyze the complexity of the algorithm. The initialization takes $O(n \log n)$ time. The number of iterations is $O(n)$, because of the following reason. Clearly, there are n site events. For each circle event for which the sweep line halts, we find a new Voronoi vertex. Then Theorem 2 implies that the number of these circle events is at most $2n - 4$.

During one iteration, we insert at most a constant number of elements into the X- and Y-structures. Hence, at any moment, these structures store $O(n)$ elements and, therefore, we can search and update them in $O(\log n)$ time per operation. During one iteration, we perform at most a constant number of operations in the X- and Y-structures. Therefore, the complete while-loop takes $O(n \log n)$ time. This shows that the entire algorithm has running time $O(n \log n)$.

At this point, we still do not have the complete Voronoi diagram. Our algorithm only computes the Voronoi vertices and for each such vertex, the three points of S that define it. (Observe that this implies that we have basically constructed the Delaunay triangulation of S.) It is, however, not difficult to compute the Voronoi edges from this information.

Theorem 3 *The Voronoi diagram of any set of n points in the plane can be constructed in $O(n \log n)$ time using $O(n)$ space.*

4.3 Some final remarks about the algorithm

The analysis of our sweep algorithm is not complete. We used the fact that a new parabola segment can only appear on the beach line if the sweep line encounters a new point of S. Although intuitively this should be true, it has to be proved. To be more precise, we have to show that a new parabola segment cannot enter the beach line through the "back door". By considering the derivatives of the parabola equations (which depend on the distances to the sweep line), it can be shown that this indeed cannot happen.

We also used the fact that a parabola segment P only disappears from the beach line if the two adjacent segments have a point of P in common. To prove this fact, we must show that a parabola segment cannot disappear because of a parabola that comes through the back door. Again, this can be proved by considering the derivatives of the parabola equations.

We have described the algorithm for constructing the Voronoi diagram for point sets that satisfy Assumption 1. This was crucial for two reasons. First, if no three points of S are collinear, then the circle events are always well-defined. Second, if no four points are cocircular, then all Voronoi vertices have degree three. The algorithm can be adapted so that it computes the Voronoi diagram of any point set. Then, however, more cases must be considered and the analysis becomes more complicated. The treatment of special cases is a general problem that has to be dealt with when implementing almost any geometric algorithm. Another problem that arises is that of finite precision arithmetic. We assumed that all computations are done exactly, which is of course not realistic.