MAX-PLANCK-INSTITUT FÜR INFORMATIK

Randomized Incremental Construction of Abstract Voronoi Diagrams

Rolf Klein Kurt Mehlhorn Stefan Meiser

MPI–I–93–105 January 1993

INFORMATIK

Im Stadtwald
66123 Saarbrücken
Germany

Randomized Incremental Construction of Abstract Voronoi Diagrams

Rolf Klein Kurt Mehlhorn Stefan Meiser

MPI-I-93-105 January 1993

Randomized Incremental Construction of Abstract Voronoi Diagrams*

Rolf Klein
FernUniversität Hagen
Praktische Informatik VI
D-5800 Hagen, Germany

Kurt Mehlhorn and Stefan Meiser
Max-Planck-Institut
für Informatik
D-6600 Saarbrücken, Germany

January 21, 1993

Abstract

Abstract Voronoi diagrams were introduced by R. Klein [Kle89b, Kle88a, Kle88b] as an axiomatic basis of Voronoi diagrams. We show how to construct abstract Voronoi diagrams in time $O(n \log n)$ by a randomized algorithm, which is based on Clarkson and Shor's randomized incremental construction technique [CS89]. The new algorithm has the following advantages over previous algorithms:

- It can handle a much wider class of abstract Voronoi diagrams than the algorithms presented in [Kle89b, MMO91].
- It can be adapted to a concrete kind of Voronoi diagram by providing a single basic operation, namely the construction of a Voronoi diagram of five sites. Moreover, all geometric decisions are confined to the basic operation, and using this operation, abstract Voronoi diagrams can be constructed in a purely combinatorial manner.

1 Introduction

The Voronoi diagram of a set of sites in the plane partitions the plane into regions, called Voronoi regions, one to a site. The Voronoi region of a site s is the set of points in the plane for which s is the closest site among all the sites.

The Voronoi diagram has many applications in diverse fields, cf. Leven and Sharir [LS86] or Aurenhammer [Aur91] for a list of applications and a history of Voronoi diagrams. Different types of diagrams result from considering different notions of distance, e. g., Euclidean or L_p-norm or convex distance functions, and different sorts of sites, e. g., points, line segments,

*A preliminary version of this paper was presented at the SIGAL Symposium on Algorithms, Tokyo 1990. This work was supported in part by the ESPRIT Basic Research Actions Program of the EC under contract No. 7141 (ALCOM II), by the BMFT (Förderungskennzeichen ITS 9103), and by the DFG grant Kl 655/2-1.

or circles. For many types of diagrams efficient construction algorithms have been found, which are either based on the divide-and-conquer technique due to Shamos and Hoey [SH75], the sweepline technique due to Fortune [For87], geometric transforms due to Brown [Bro79] and Edelsbrunner and Seidel [ES86], or the randomized incremental construction technique due to Clarkson and Shor [CS89].

A unifying approach to Voronoi diagrams has been proposed by Klein [Kle88a, Kle88b, Kle89a, Kle89b], cf. [ES86] for a related approach. He does not use the concept of distance as the basic notion but rather the concept of bisecting curves, i. e., he assumes for each pair $\{p, q\}$ of sites the existence of a bisector $J(p, q)$, which is homeomorphic to a line and divides the plane into a p–region and a q–region. The intersection of all p–regions for different q's is then the Voronoi region of site p. He also postulates that Voronoi regions are simply-connected and partition the plane. He shows that abstract Voronoi diagrams already have many of the properties of concrete Voronoi diagrams, cf. Section 2.

At present there are two algorithms for the construction of abstract Voronoi diagrams. Both algorithms assume that certain elementary operations on bisecting curves, e. g., computation of the intersections, take $O(1)$ time, and both algorithms can handle only subclasses of abstract Voronoi diagrams.

Klein [Kle89b] presented an off-spring of the Shamos and Hoey divide-and-conquer algorithm. He has to assume that any set S of sites can be split in time $O(|S|)$ into approximately equal sized subsets L and R such that the bisector between L and R (= the common boundary of regions in L with regions in R) is acyclic and, under this assumption, constructs the Voronoi diagrams of n sites in time $O(n \log n)$. There are cases, e. g., points with additive weights in the Euclidean plane, where it is not known if such partitions exist.

Mehlhorn, Meiser and Ó' Dúnlaing [MMO91] have presented an off-spring of the Clarkson and Shor randomized incremental algorithm. They have to assume that the set of bisectors is regular, i. e., no four of them share a point and any point of intersection of two bisectors is a proper crossing of the bisectors. Under these assumptions, their algorithm runs in expected time $O(n \log n)$, the average being taken over all permutations of the input. There are cases, e. g., point sites in the Manhattan metric, where this assumption does not hold.

In this paper, we extend the randomized incremental algorithm and show that it can handle abstract Voronoi diagrams in (almost) their full generality; cf. the remark following Definition 1 in Section 2 for the minor restriction which we have to make. The algorithm runs in expected time $O(n \log n)$ and is as simple as the algorithm in [MMO91]. However, its correctness proof and running time analysis are more involved. The algorithm is uniform in the sense that only a single operation, namely the construction of a Voronoi diagram for 5 sites, depends on the specific type of Voronoi diagram and has to be programmed in order to adapt the algorithm to the type of the diagram. Moreover, all numerical operations take place within this particular operation.

In particular, comparisons only take place between objects which are related in the topology of the diagram. The incremental algorithm of Guibas and Stolfi [GS85] for Euclidean diagrams also has this property but neither the Plane-Sweep- nor the Divide-and-Conquer-algorithm do. Both algorithms need to sort the sites by x-coordinates. Moreover, the Plane-Sweep-algorithm sorts the computed events by x-coordinates; the Divide-and-

Conquer-algorithm sorts the nodes of the diagram by y-coordinates in its merge step. In both cases, objects that are not at all related in their topology are compared to each other. Therefore, it may be difficult to make geometric decisions in a consistent manner. From a programmer's point of view, concentrating the numerical computations inside a single operation may facilitate the handling of approximate arithmetic. We want to emphasize that the fact that our basic operation operates on five sites does not imply that an implementation of the basic operation must use tests which involve five sites and therefore are likely to have high algebraic degree. We show in Section 6 that four sites suffice for *simple* families of bisectors, i. e., families of bisectors where the Voronoi diagram of any three sites has at most one vertex.

As mentioned above, our algorithm is based on Clarkson and Shor's randomized incremental construction technique [CS89]. We make use of the refinement proposed in [GKS92, BD89, BDT90, BDS+92]; in particular, we use the notion of history graph instead of the original conflict graph.

An earlier version of the algorithm, which uses a conflict graph instead of a history graph, was implemented by N. Zimmer [Zim92]. We have used it to construct Powerdiagrams, Voronoi diagrams of line segments under the Euclidean metric, and Voronoi diagrams of points under both the Euclidean and the L_1-metric. The general, diagram-independent part of the algorithm thereby comprises circa 2700 lines of code. This should be compared to the amount of code needed to implement the diagram-specific part of the algorithm (basic operation and drawing routines). This part varies between 450 lines for points under the Euclidean metric and 3250 lines of code for line segment sites under the Euclidean metric. For Powerdiagrams and diagrams of points under the L_1-metric we needed 550 and 850 lines, respectively. Note that approximately one sixth of this is code for drawing the diagram on the screen.

The present paper is not quite in line with a popular trend in computational geometry: to use symbolic perturbation to establish general position, then to use an algorithm which can only handle inputs in general position (e. g., the algorithm of [MMO91]), and finally to produce the true output by a limit process (essentially by shrinking some edges and collapsing vertices). We cannot follow this approach for several reasons. Firstly, there is no efficient perturbation technique available for abstract Voronoi diagrams. Klein [Kle89a] showed that any admissible family of bisectors can be perturbed to general position, but his perturbation technique can require exponential time and needs to know the Voronoi diagram. Secondly, we did not want to use a perturbation technique which is outside our algorithm, e. g., one which uses properties of the particular kind of diagram under construction, because this would require programming the limit process for each particular kind of diagram. We believe that it is better to make the algorithm as uniform as possible and to confine the dependency on the particular kind of diagram to a single subroutine (here, the construction of a five sites diagram). Thirdly, perturbation and limit process are not always a trivial task. Consider for example the Voronoi diagram of a point and four open line segments touching in this point, cf. Figure 1. The perturbation is non-trivial, since it should not introduce intersections between the segments. The limit process is non-trivial, since it must collapse the point with the four endpoints of the segments. But these features are not directly linked

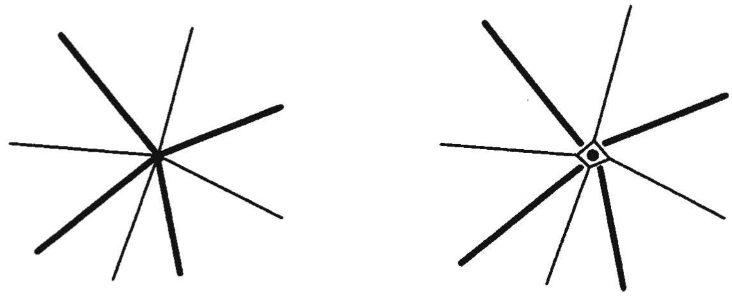


Figure 1: The Voronoi diagram for a degenerate and a perturbed input.

in a typical data structure for the perturbed diagram. Finally, perturbation might increase the running time by more than a constant factor. The expected running time of our algorithm is proportional to $\sum_{i\leq n} \frac{f_i}{i^2}(n-i)$, where f_i is the expected number of edges in a diagram for a random subset of i sites from the n given sites. Since regions in abstract Voronoi diagrams may be empty, we may have $f_i = o(i)$. In such a situation, the running time of the algorithm can be $o(n \log n)$. Perturbation creates general position and may increase f_i to $\Theta(i)$. Finally, we believe that despite the handling of degenerate cases the algorithm presented in this paper is still very simple. Degenerate cases complicate the discussion of correctness and running time, but affect the algorithm itself only to a small extent.

The paper is organized as follows: In Section 2 we introduce abstract Voronoi diagrams; we give the relevant definitions and state some properties. In Section 3 we investigate the Voronoi diagram of five sites and present the basic operation of our algorithm. The algorithm is then given in Section 4. Section 5 contains the analysis of the algorithm's running time and space requirements. In Section 6 we inspect the basic operation for a subclass of abstract Voronoi diagrams in more detail.

Throughout the paper, we use the following notation: For a subset $X \subseteq \mathbb{R}^2$ the closure, boundary and interior of X are denoted by $\overline{X}$, *bd* X and *int* X, respectively.

2 Abstract Voronoi Diagrams

Let $n \in \mathbb{N}$, and for every pair of integers p, q such that $1 \leq p \neq q < n$ let $D(p,q)$ be either empty or an open unbounded subset of $\mathbb{R}^2$ and let $J(p,q)$ be the boundary of $D(p,q)$. We postulate:

1. $J(p,q) = J(q,p)$ and for each p,q such that $p \neq q$ the regions $D(p,q)$, $J(p,q)$ and $D(q,p)$ form a partition of $\mathbb{R}^2$ into three disjoint sets.
2. If $\emptyset \neq D(p,q) \neq \mathbb{R}^2$ then $J(p,q)$ is homeomorphic to the open interval $(0,1)$.

We call $J(p,q)$ the bisecting curve for sites p and q and $D(p,q)$ the region of dominance of p over q. Following [Kle89b], the abstract Voronoi diagram is now defined as follows:

Definition 1 *Let* $S = \{1, \ldots, n-1\}$ *and*

$$R(p,q) := \begin{cases} D(p,q) \cup J(p,q) & \text{if } p < q \\ D(p,q) & \text{if } p > q \end{cases}$$

$$EVR(p,S) := \bigcap_{\substack{q \in S \\ q \neq p}} R(p,q)$$

$$VR(p,S) := \mathit{int}\ EVR(p,S)$$

$$V(S) := \bigcup_{p \in S} \mathit{bd}\ EVR(p,S)$$

$VR(p,S)$ is called the Voronoi region *of p or p–*region *w.r.t. to S, $EVR(p,S)$ is called the* extended Voronoi region *of p w.r.t. S, and $V(S)$ is called the* Voronoi diagram *of S. The elements of S are referred to as* sites.

We require the Voronoi regions and the bisecting curves to satisfy the following two conditions:

3. Any two bisecting curves intersect in only a finite number of connected components.
4. For all non-empty subsets S' of S
   (a) for all $p \in S'$ for which $EVR(p,S')$ is non-empty: $VR(p,S')$ is non-empty and $EVR(p,S')$ and $VR(p,S')$ are path-connected,
   (b) $\mathbb{R}^2 = \bigcup_{p \in S'} EVR(p,S')$

Abstract Voronoi diagrams include a large number of concrete Voronoi diagrams, e.g., Voronoi diagrams for point sites under any L_p-metric, $1 \leq p \leq \infty$, or under any convex distance function, whose unit circle is semi-algebraic. They furthermore comprise Power-diagrams, and Voronoi diagrams for line segments or circles under the Euclidean metric. The line segments may even touch at their endpoints, thus possibly forming polygons, and the circles are allowed to intersect. Voronoi diagrams for disjoint convex figures under a convex distance function are also included, provided their bisectors satisfy our Condition 3. Of course, there are also negative examples: Euclidean Voronoi diagrams for point sets with multiplicative weights or Euclidean Voronoi diagrams for non-convex figures, e.g., circular arcs. In both cases there may be circular bisecting curves violating our Condition 2. Figures 2 and 3 show two abstract Voronoi diagrams.

Abstract Voronoi diagrams are defined by means of bisecting curves. Depending on the concrete Voronoi diagram, the complexity of the bisectors may vary considerably. For the sake of simplicity we assume however that bisectors are computationally simple (see Section 3). We will show that under these assumptions abstract Voronoi diagrams can be constructed in time $O(n \log n)$ by a randomized algorithm.

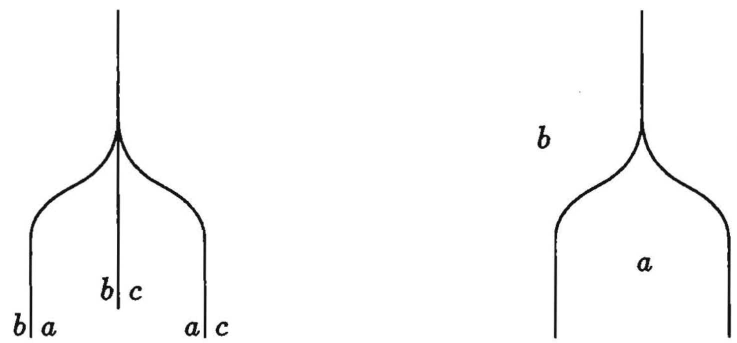


Figure 2: A family of three bisectors for sites a, b and c, and the induced Voronoi diagram. For each bisector the two sites separated by the bisector are indicated near to the bisector.

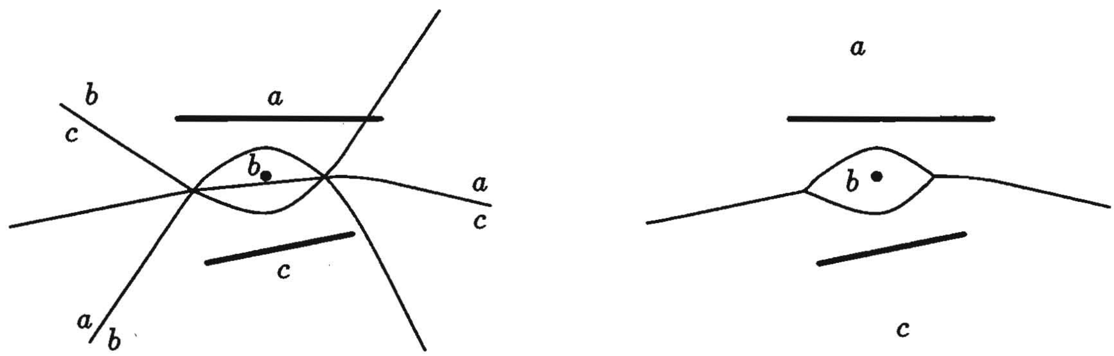


Figure 3: A family of three bisectors arising from two line segment sites a and c and one point site b under the Euclidean metric. The bisectors are drawn as thin curves, the segments are fat.

Remarks:

1. In [Kle89b] Condition 4b is shown to be equivalent to the following transitivity property: $R(p,q) \cap R(q,r) \subseteq R(p,r)$ for any three pairwise distinct sites $p, q, r \in S$.

2. The union in 4b is disjoint by the definition of Voronoi regions.

3. Our Definition of an abstract Voronoi diagram differs in two respects from Klein's original definition in [Kle89b]. Firstly, we also allow empty Voronoi regions which does not harm Klein's theory. Secondly, our Condition 4a is slightly more restrictive than the one in [Kle89b]. There, only the extended Voronoi regions are required to be path-connected, but not their interior. Figure 4 shows a system of bisectors for three sites p, q, r, which satisfies Klein's assumptions if $p < q$ and $p < r$. Our assumptions, however, exclude this example since its p–region is disconnected.

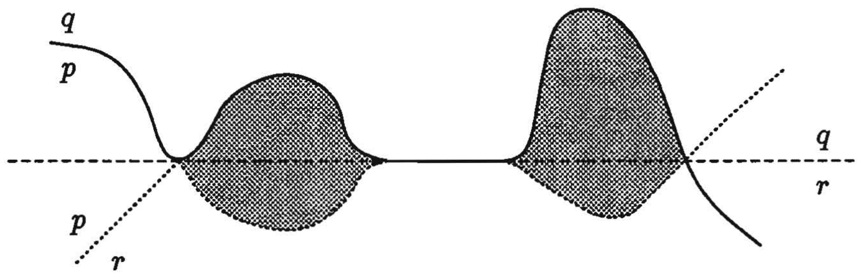


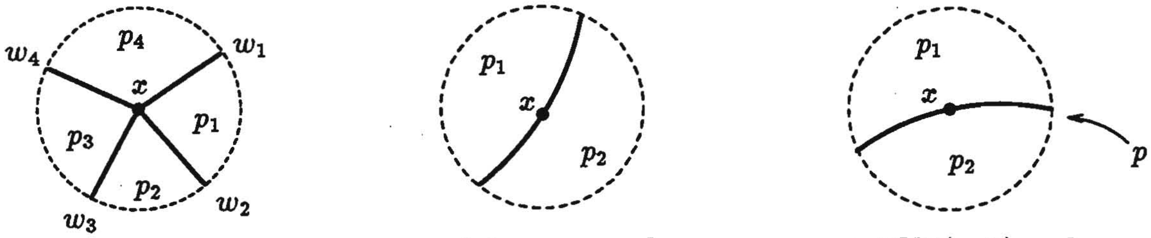
Figure 4: A degenerate case.

Definition 2 *An* edge e *of* $V(S)$ *is a maximal connected subset of* $V(S)$ *such that every point* $x \in e$ *lies on bd* $VR(p,S)$ *for exactly two sites* p *of* S. *The edge is said to* separate *the regions of these two sites. A* vertex v *of* $V(S)$ *is a point* $x \in V(S)$ *which lies on bd* $VR(p,S)$ *for at least three sites* p *of* S.

Fact 1 (piece of pie fact)

1. *All but finitely many points of* $V(S)$ *belong to an edge of* $V(S)$.

2. *For each point* $x \in V(S)$ *there are arbitrarily small neighborhoods* U *of* x *having the following properties:* $V(S) \cap$ *bd* U *consists of finitely many points. Let* $w_1, \ldots, w_h$ *denote these points as encountered in a clockwise traversal of bd* U. *Then* $h \geq 2$ *and* $V(S) \cap U$ *is the union of curve segments* $\beta_1, \ldots, \beta_h$ *where* β_i *connects* x *to* w_i *and the* β_i*'s are disjoint except at their common endpoint* x. *For each* i, $1 \leq i \leq h$, *there is a site* $p_i \in S$ *such that the open "piece of pie" bordered by* β_i, β_{i+1} *(read indices mod h) is contained in* $VR(p_i, S)$. *Then* $p_i \neq p_j$ *for* $i \neq j$. *For each* β_i *there is a site* $q_i \in S$, *such that* $\beta_i - x \subseteq EVR(q_i, S)$. *We have* $q_i \leq \min\{p_{i-1}, p_i\}$. *The point* x *belongs to* $EVR(p,S)$, *where* $p = \min\{p_1, \ldots, p_h, q_1, \ldots, q_h\}$. *Also, only the extended Voronoi region of site* p *can be encountered more than once on the march around bd* U.



a vertex x | a point x on an edge $x \in EVR(\min\{p_1, p_2\}, S)$ | $x \in EVR(p, S)$ and $p < \min\{p_1, p_2\}$

Figure 5: Illustration of Fact 1.

Figure 5 illustrates Fact 1. Fact 1 is an immediate derivative of Theorem 2.3.5 of [Kle89b]. There only $p_{i-1} \neq p_i$ was claimed. The claim $p_i \neq p_j$ for $i \neq j$ made here follows from our strengthened Condition 4a and Lemma 2.2.4 of [Kle89b].

For the sequel, it is helpful to restrict attention to the "finite part" of $V(S)$. Let Γ be a simple closed curve such that in the outer domain of Γ the curve segments of any two bisectors are either disjoint or identical. We add a site ∞ to S, define $J(p, \infty) = J(\infty, p) = \Gamma$ for all p, $1 \leq p < n$, and $D(\infty, p)$ to be the outer domain of Γ for each p, $1 \leq p < n$.

Fact 2 *$V(S)$ is connected. The extended Voronoi region of a site $p \in S - \{\infty\}$ is simply-connected, each non-empty Voronoi region $VR(p, S)$, $p \in S - \{\infty\}$, is homeomorphic to an open disc and its boundary is a simple closed curve. The Voronoi region of site ∞ is not simply connected but it has only one hole being the inner domain of Γ. A Voronoi diagram can be represented as a planar graph in a natural way. The vertices and edges of the graph are the vertices and edges of $V(S)$, respectively; the faces of the graph correspond to the non-empty Voronoi regions. We use $V(S)$ also to denote this graph.*

For a proof of Fact 2 see Lemma 2.2.4 and Theorems 2.3.5 and 2.5.5 [Kle89b].

The extended Voronoi region $EVR(p, S)$ for a site $p \in S - \{\infty\}$, consists of its Voronoi region, some vertices and edges on the boundary of $VR(p, S)$, and some other vertices and edges of $V(S)$. The other edges and vertices form trees rooted at *bd* $VR(p, S)$, cf. Figure 6.

We will next return to the example in Figure 2 in order to illustrate the concepts introduced so far. We will use this example as our running example throughout the paper.

Example: Figure 7 shows $V(\{a, b, \infty\})$ and $V(\{a, b, c, \infty\})$ for the bisectors defined in Figure 2. Assume $a < b < c < \infty$. Then edges e_2 and e_3 belong to $EVR(a, \{a, b, \infty\})$ and edges e_4, e_5, e_7 and e_8 belong to $EVR(a, \{a, b, c, \infty\})$.

Inserting a new site

The algorithm presented in Section 4 constructs the Voronoi diagram $V(S)$ by adding one site after the other. In this section we investigate the part of a Voronoi diagram that is "cut

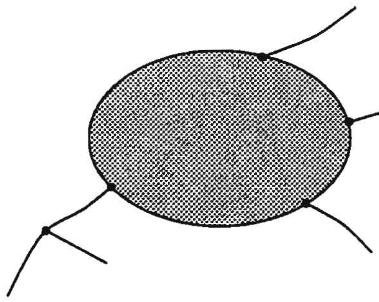


Figure 6: An extended Voronoi region $EVR(p, S)$.

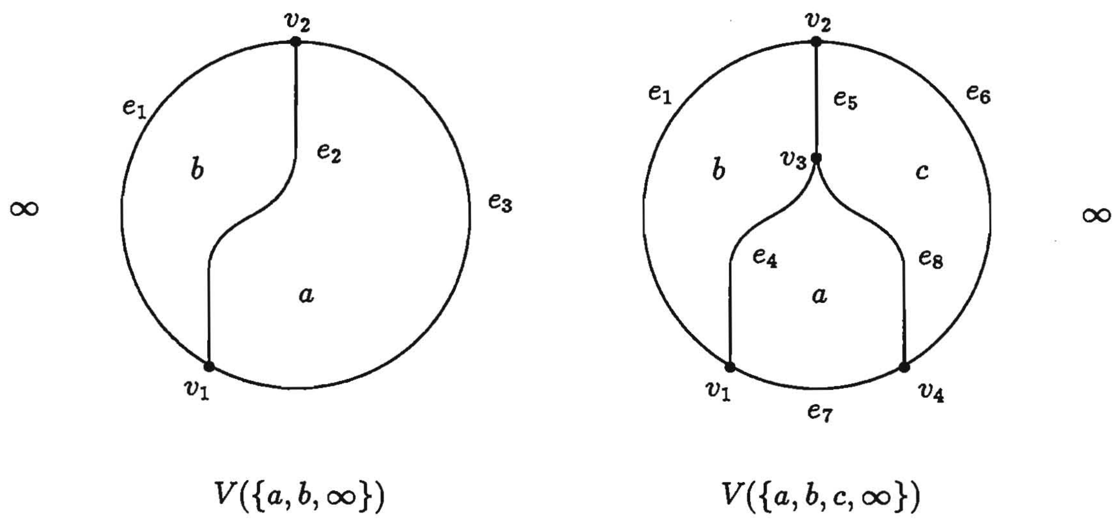


Figure 7: The enclosing circle represents Γ. Edges e_1 and e_3 are part of $J(b, \infty) = \Gamma$ and $J(a, \infty) = \Gamma$, respectively.

off" by the insertion of a new site. For the remainder of the section, let $R \subseteq S$, $\infty \in R$, $s \in S - R$, $\mathcal{F} = VR(s, R \cup \{s\})$, and $\mathcal{E} = V(R) \cap \overline{\mathcal{F}}$. Then, according to Fact 2, *bd* $\mathcal{F}$ = *bd* $\overline{\mathcal{F}}$ is a simple closed curve.

Lemma 1 *If $\mathcal{F} \neq \emptyset$ then $\mathcal{E}$ is a non-empty connected set which intersects bd $\mathcal{F}$. Moreover, $\mathcal{E}$ is not just a single point.*

Proof: If $\mathcal{E}$ were empty, then $\overline{\mathcal{F}} \subseteq VR(p, R)$ for some $p \in R - \{\infty\}$. Consequently, $VR(p, R \cup \{s\})$ would not be simply connected. Now let $\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_k$ be the connected components of $\mathcal{E}$ for some k. Observe that no $\mathcal{E}_j$ can be entirely contained inside $\mathcal{F}$ because otherwise $V(R)$ would not be connected, a contradiction.

Assume $k \geq 2$. Then a path $\mathcal{P} \subseteq \overline{\mathcal{F}} - \mathcal{E}$ exists, connecting two points x and y on the boundary of $\mathcal{F}$ and separating $\mathcal{E}_1$ from $\mathcal{E}_2$, cf. Figure 8. From $\mathcal{P} \cap \mathcal{E} = \emptyset$ we have

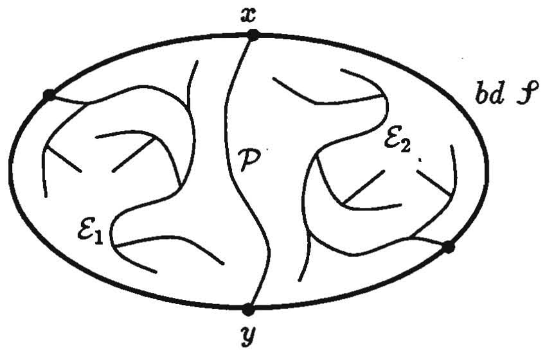


Figure 8: Path $\mathcal{P}$ and two connected components of $\mathcal{E}$.

$\mathcal{P} \cap V(R) = \emptyset$ and thus $\mathcal{P} \subseteq VR(r, R)$ for a site $r \in R - \{\infty\}$. $x, y \in \mathcal{P}$ implies that all sufficiently small neighborhoods $U(x)$ and $U(y)$ are entirely contained in $VR(r, R)$. The points in the intersection of these neighborhoods with the complement of $\overline{\mathcal{F}}$ thus lie in $VR(r, R \cup \{s\})$ and can be connected by a path $\mathcal{Q} \subseteq VR(r, R \cup \{s\}) \subseteq VR(r, R)$. The cycle $\mathcal{P} \circ \mathcal{Q}$ is therefore entirely contained in $VR(r, R)$ and contains $\mathcal{E}_1$ or $\mathcal{E}_2$ in its interior. This is a contradiction.

At this point we have shown that $\mathcal{E}$ is a non-empty connected set which intersects *bd* $\mathcal{F}$. Assume now that $\mathcal{E}$ is a single point. This point, say v, is either a vertex of $V(R)$ or lies on an edge of $V(R)$. In either case, one of the regions of $V(R)$ incident to v in $V(R)$ is split by $\mathcal{F}$ in a neighborhood of v and hence represented twice at v in $V(R \cup \{s\})$, cf. Figure 9, a contradiction to the piece of pie fact. □

Note that Lemma 1 implies in particular, that if $\mathcal{F} \neq \emptyset$, then $\overline{\mathcal{F}}$ intersects an edge of $V(R)$. Lemma 2 discusses the various forms which an intersection between $\overline{\mathcal{F}}$ and an edge e of $V(R)$ can have.

Lemma 2 *Let e be an edge of $V(R)$. If $e \cap \overline{\mathcal{F}} \neq \emptyset$, then either $e \cap \overline{\mathcal{F}} = V(R) \cap \overline{\mathcal{F}}$ and $e \cap \overline{\mathcal{F}}$ is a single component or $e - \overline{\mathcal{F}}$ is a single component (possibly empty).*

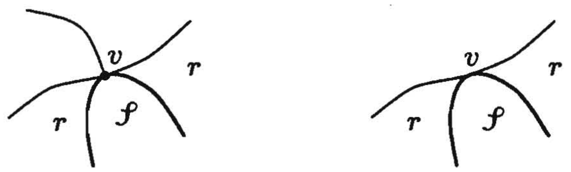


Figure 9: r–region is represented twice at v, a contradiction to Fact 1.

Proof: Assume first that $e \cap \overline{\mathcal{F}} = V(R) \cap \overline{\mathcal{F}}$. Since $V(R) \cap \overline{\mathcal{F}}$ is connected according to Lemma 1, $e \cap \overline{\mathcal{F}}$ is also connected. Assume next that $e \cap \overline{\mathcal{F}} \neq V(R) \cap \overline{\mathcal{F}}$. Then for every point $x \in e \cap \overline{\mathcal{F}}$ one of the subpaths of e connecting x to an endpoint of e must be contained in $\overline{\mathcal{F}}$, since $V(R) \cap \mathcal{F}$ is a connected set. Hence $e - \overline{\mathcal{F}}$ is a single component. □

Figure 10 shows some possible and impossible configurations for $e \cap \overline{\mathcal{F}}$ according to Lemma 2. Figure 11, which also illustrates the following Definition 3, shows that cases (a) and (d) of Figure 10 arise even for Euclidean diagrams of line segment sites.

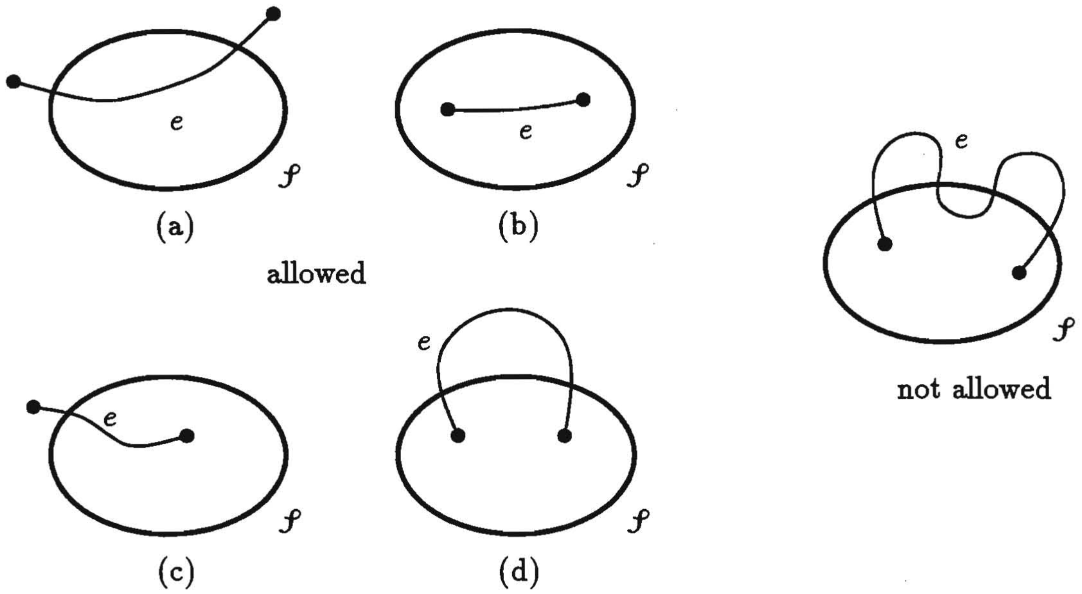


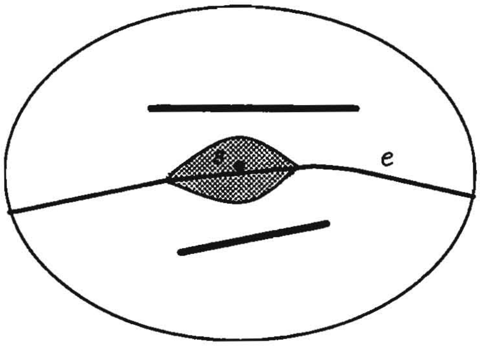
Figure 10: Four situations as allowed by Lemma 2 and an impossible one.

We close this section with some notations that we need in the forthcoming sections.

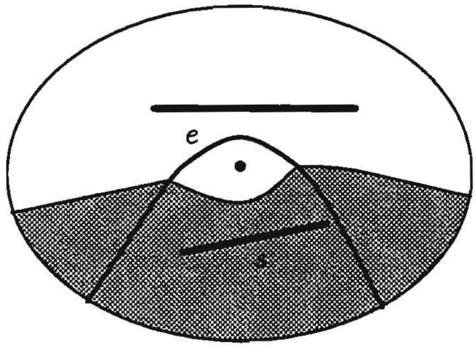
Definition 3 *Let e be an edge of $V(R)$ and let v be an endpoint of e. Then*

1. s intersects e with respect to R *iff* $e \cap \overline{\mathcal{F}} \neq \emptyset$.

2. s clips e at v with respect to R *iff* $e \cap \overline{\mathcal{F}}$ *contains a component incident to* v.



site s intersects edge e without clipping



site s clips e at both endpoints

Figure 11: The Euclidean Voronoi diagram of two line segment sites and one point site. Two sites form an edge e. The third site s is inserted into the diagram of the other two sites and thereby intersects edge e. The Voronoi region of site s is shown shaded.

3 The Basic Operation

Computing the intersection between an edge and the region of a new site is the fundamental operation in our algorithm. We have already seen in Lemma 2 that there are only a few types of such intersections. In this section we show that a particular type of intersection can be extracted from the Voronoi diagram of only five sites and therefore computed in constant time. The five sites involved are the newly added site and four sites "defining" the edge. We first specify how sites "define" edges. As above let $R \subseteq S$ and $\infty \in R$.

Definition 4 *Let p, q, r and t be sites in R.*

1. *A vertex v of $V(R)$ is called a pqr–vertex, if v is incident to the p-, q-, and r–regions, and there is a clockwise traversal of the regions incident to v which encounters p–region before q–region before r–region before p–region.*
2. *An edge e of $V(R)$ is called a $prqt$–edge, if e separates p- and q–region, and its endpoints are prq- and qtp–vertices.*

Example (continued): We continue our running example. Let v_1, v_2, e_1, e_2, and e_3 be defined as in the left picture of Figure 7. Then the vertex v_2 is an $ab\infty$–vertex and the vertex v_1 is a $ba\infty$–vertex. Edge e_1 is a $ba\infty a$–edge, e_2 is a $b\infty a\infty$–edge and e_3 is an $ab\infty b$–edge.

Lemma 3 *Let $R \subseteq S$ and let p, q, r and t be sites in R. Then $V(R)$ contains at most one pqr–vertex and at most one $prqt$–edge.*

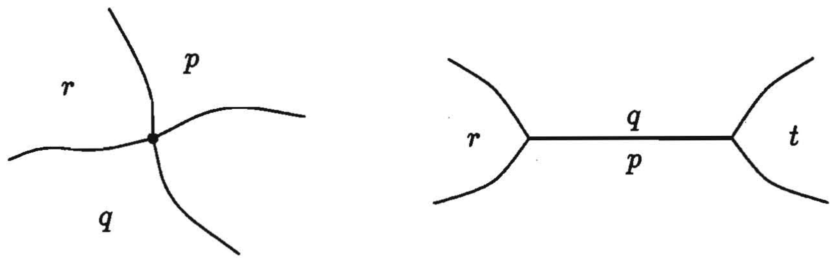


Figure 12: A pqr–vertex and a $prqt$–edge.

Proof: We first show that there is at most one pqr–vertex. Assume otherwise that, say, v and w are two distinct pqr–vertices. Since $VR(p, R)$ and $VR(q, R)$ are path-connected, there are paths $\mathcal{P}$ and $\mathcal{Q}$ connecting v and w and running completely (except at their endpoints) inside p- and q–region respectively. The cycle $\mathcal{P} \circ \mathcal{Q}$ then contains r–region in its interior and its exterior, a contradiction to the fact that $VR(r, R)$ is homeomorphic to a disc. Thus there is at most one pqr–vertex.
The existence of two $prqt$–edges clearly contradicts the existence of at most one prq–vertex. $\square$

Four-tuples of sites not only allow us to distinguish between different edges of the same diagram, they furthermore capture all information necessary to compute the intersection of an edge with a new region:

Lemma 4 (basic operation lemma) *Let e be a $prqt$–edge of $V(R)$. Then the point set e also constitutes a $prqt$–edge of $V(R')$ for all R' with $\{p, r, q, t\} \subseteq R' \subseteq R$. Moreover, $e \cap \overline{VR(s, R \cup \{s\})} = e \cap \overline{VR(s, R' \cup \{s\})}$ for any $s \notin R$.*

Proof: Since $VR(o, R') \supseteq VR(o, R)$ for $o \in R'$, the point set e is incident to the Voronoi regions of p, q, r and t w.r.t. R', too. In particular the ordering of these Voronoi regions around e does not change. Thus e is a $prqt$–edge in $V(R')$ as well. Let $s \in S - R$ be arbitrary. Observe first that $e \cap \overline{VR(s, R \cup \{s\})} \subseteq e \cap \overline{VR(s, R' \cup \{s\})}$ follows from $VR(s, R \cup \{s\}) \subseteq VR(s, R' \cup \{s\})$. For the converse, let $x \in e \cap \overline{VR(s, R' \cup \{s\})}$ be arbitrary. Since e is an edge of $V(R)$ separating p- and q–region with respect to R, there are arbitrarily small neighborhoods U of x such that $U - e \subseteq VR(p, R) \cup VR(q, R)$. Since $x \in \overline{VR(s, R' \cup \{s\})}$, for each such U there is a point $y \in U - e$ for which $y \in VR(s, R' \cup \{s\})$. On the other hand $y \in U - e$ implies $y \in VR(p, R') \cup VR(q, R')$. We conclude $y \in D(s, p)$ or $y \in D(s, q)$. Since $y \in VR(p, R) \cup VR(q, R)$ this implies $y \in VR(s, R \cup \{s\})$. The claim $x \in \overline{VR(s, R \cup \{s\})}$ follows because we can assume U to be arbitrarily small. $\square$

By Lemmas 3 and 4 a $prqt$–edge e of $V(R)$ is also the unique $prqt$–edge of $V(\{p, r, q, t\})$ and the intersection between e and the region of site s is the same in $V(R \cup \{s\})$ as in $V(\{p, r, q, t, s\})$. We therefore define the following operation as the basic operation of our algorithm:

Basic Operation

Input: a five-tuple (p, r, q, t, s) such that
1) $V(\{p, r, q, t\})$ contains a *prqt*–edge e, and
2) $s \notin \{p, r, q, t\}$.

Output: The combinatorial structure of $e \cap \overline{VR(s, \{p, r, q, t, s\})}$, i. e., one of the following:
1) intersection is empty
2) intersection is non-empty and consists of a single component:
   a) e itself
   b) a segment of e adjacent to the *prq*-endpoint
   c) a segment of e adjacent to the *qtp*-endpoint
   d) a segment not adjacent to any endpoint of e
3) intersection is non-empty and consists of exactly two components

Each call of *basic_op* will be charged one time unit. Note that the input to the basic operation is a combinatorial object, namely the 5-tuple (p, r, q, t, s), and that the output is a combinatorial object, namely a symbol in $\{1, 2a, 2b, 2c, 2d, 3\}$. Also note that the six cases specified exhaust all possible cases by Lemma 2 and that in case 3 the two components are incident to one endpoint of e each. We use $basic\_op(p, r, q, t, s)$ to denote the output of the basic operation on input (p, r, q, t, s).

It is clear that the basic operation can also be used to decide whether an edge is intersected or clipped by a site. Let e be a *prqt*–edge of $V(R)$ and $s \in S - R$. Then s intersects e iff $basic\_op(p, r, q, t, s) \in \{2a, 2b, 2c, 2d, 3\}$ and s clips e at the *prq*–endpoint iff $basic\_op(p, r, q, t, s) \in \{2a, 2b, 3\}$.

Example (continued): In $V(R)$, $R = \{a, b, \infty\}$, of Figure 7 we have $basic\_op(b, \infty, a, \infty, c) = 2b$, $basic\_op(a, b, \infty, b, c) = 2c$ and $basic\_op(b, a, \infty, a, c) = 1$. Thus site c intersects edges e_2 and e_3 and clips edges e_2 and e_3 at their endpoint v_2 with respect to $R = \{a, b, \infty\}$.

We have seen that four sites uniquely define an edge in the sense that there is no other edge defined by the same tuple of sites. However, an edge may in this way be defined by several different four-tuples of sites. In the analysis of our algorithm in Section 5 and for the presentation of the algorithm we need a stronger combinatorial characterization of edges.

Definition 5 *1. Let e be an edge of $V(R)$ separating p- and q–region. Let f_p and g_p be the edges preceding and following e in a clockwise traversal of the boundary of $VR(p, R)$, and let f_q and g_q be the edges preceding and following e in a counter-clockwise traversal of the boundary of $VR(q, R)$, cf. Figure 13. Assume further that f_p separates p- and r_p–region, g_p separates p- and t_p–region, and g_q separates q- and t_q–region, and f_q separates q- and r_q–region. Then $D_R(e) = \{(r_q, q, p, r_p), (t_p, p, q, t_q)\}$ is called the* description of e with respect to R. *By $set(D_R(e))$ we denote the set $\{p, q, r_p, r_q, t_p, t_q\}$.*

2. Let D be the description of an $\underline{\text{edge } e \text{ of } V(R)}$, and let $s \in S - set(D)$. Then site s intersects description D *iff* $e \cap \overline{VR(s, set(D) \cup \{s\})} \neq \emptyset$.

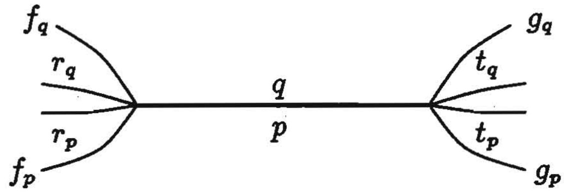


Figure 13: An edge with description $\{(r_q, q, p, r_p), (t_p, p, q, t_q)\}$.

Example (continued): In $V(R)$, $R = \{a, b, \infty\}$, of Figure 7 the edges have the following descriptions: $D_R(e_1) = \{(a, b, \infty, a), (a, \infty, b, a)\}$, $D_R(e_2) = \{(\infty, b, a, \infty), (\infty, a, b, \infty)\}$, and $D_R(e_3) = \{(b, a, \infty, b), (b, \infty, a, b)\}$.

Remarks:

1. The basic operation lemma has the following consequence: An edge e of $V(R)$ with description D is also an edge of $V(set(D))$, and site $s \in S - R$ intersects edge e with respect to R iff s intersects e with respect to $set(D)$ iff s intersects D. Moreover, using our basic operation we can decide in constant time whether or not a site $s \in S - set(D)$ intersects description D.

2. In the case of general position, i. e., if Voronoi vertices have degree 3, the four-tuple "defining" an edge and the description of the edge contain the same set of sites. In fact, in that case, the whole analysis could also be done with four-tuples as descriptions. Descriptions are only introduced for the handling of degenerate cases, especially in the analysis of the algorithm in Section 5.

4 The Incremental Algorithm

In this section, we describe the incremental construction algorithm. The algorithm starts with the set $R_3 = \{\infty, p, q\}$, where p and q are chosen uniformly at random from $S - \{\infty\}$, and then adds the remaining sites in random order, i. e., $R_{k+1} = R_k \cup \{s\}$, where s is chosen uniformly at random from $S - R_k$. The following data structures are maintained for the current set $R = R_k$ of sites:

1. The Voronoi diagram $V(R)$: It is stored as a planar map; with every face of $V(R)$ the corresponding site in R is stored.

2. The history graph $\mathcal{H}(R)$: It is a directed acyclic graph with a single source. Its vertex set is given by

$$\{source\} \cup \bigcup_{3 \leq i \leq k} \{D_{R_i}(e) \mid e \text{ is an edge of } V(R_i)\}.$$

The following history-graph invariants are maintained:

1. Every vertex of $\mathcal{H}(R)$ has outdegree at most 5 and the vertices in $\{D_R(e) \mid e \text{ edge of } V(R)\}$ have outdegree 0, i. e., are leaves of the graph.
2. Every edge e of $V(R)$ is linked to its corresponding description $D_R(e)$ of $\mathcal{H}(R)$ and vice versa.
3. For every site $s \in S - R$ and every leaf D of $\mathcal{H}(R)$ that is intersected by s there is a path from *source* to D whose vertices are all intersected by s.

We now discuss how to construct $V(R \cup \{s\})$ and $\mathcal{H}(R \cup \{s\})$ from $V(R)$ and $\mathcal{H}(R)$. To this aim let $E_s = \{e \mid e \text{ is an edge of } V(R) \text{ and } e \text{ is intersected by } s\}$. We first show how to construct E_s (Step 1), from $\mathcal{H}(R)$ and $V(R)$ in time proportional to the number c of vertices of $\mathcal{H}(R)$ which are intersected by s. Given E_s, it is then easy to construct $V(R \cup \{s\})$ (Step 2), and $\mathcal{H}(R \cup \{s\})$ (Step 3) in time $O(|E_s|)$.

Step 1: Construction of E_s

Starting at the source of $\mathcal{H}(R)$ we explore all descriptions in $\mathcal{H}(R)$ which are intersected by s. Since the outdegree of $\mathcal{H}(R)$ is bounded by 5, the number of visited vertices is proportional to c. Note that we can decide in constant time whether or not a description is intersected by a site by using our basic operation. Thus the search takes time $O(c)$. Also, by the third history-graph invariant, it identifies all leaves of $\mathcal{H}(R)$ intersected by s. By the second history-graph invariant this set immediately gives the set of edges of $V(R)$ whose descriptions are intersected by s. By the basic operation lemma this is set E_s. We conclude:

Lemma 5 *The set E_s can be computed in time $O(c)$.*

Step 2: Construction of $V(R \cup \{s\})$

As above, let $\mathcal{J} = VR(s, R \cup \{s\})$. We know from Lemma 1 that $\mathcal{J} \neq \emptyset$ iff $E_s \neq \emptyset$. So, $V(R \cup \{s\}) = V(R)$ and $\mathcal{H}(R \cup \{s\}) = \mathcal{H}(R)$ if $E_s = \emptyset$. We therefore assume from now on that $E_s \neq \emptyset$. For an edge $e \in E_s$, $e - \overline{\mathcal{J}}$ consists of at most two subsegments of e. Also, if e is a *prqt*–edge of $V(R)$ *basic_op*(p, r, q, t, s) tells us the structure of $e - \overline{\mathcal{J}}$. We call a point v an endpoint of $e - \overline{\mathcal{J}}$ if it is an endpoint of one of the subsegments of e. In this way, $e - \overline{\mathcal{J}}$ may have 0, 2, or 4 endpoints. These endpoints are distinct by Lemma 2. We first characterize the vertices of $V(R \cup \{s\})$. Let V be the set of vertices of $V(R)$ and let

$V_{del} = \{v \mid v \in V \text{ and all edges incident to } v \text{ are clipped at } v \text{ by } s\}$
$V_{unch} = \{v \mid v \in V \text{ and no edge incident to } v \text{ is clipped at } v \text{ by } s\}$
$V_{chang} = \{v \mid v \in V \text{ and some but not all edges incident to } v \text{ are clipped at } v \text{ by } s\}$
$V_{new} = \{v \mid v \notin V \text{ and } v \text{ is endpoint of } e - \overline{\mathcal{J}} \text{ for some } e \in E_s\}$

Example (continued): Let $R = \{a, b, \infty\}$ and $s = c$. Then $V_{del} = \emptyset$, $V_{unch} = \{v_1\}$, $V_{chang} = \{v_2\}$, and $V_{new} = \{v_3, v_4\}$. Note that our basic operation tells us that $e_2 - \overline{\mathcal{J}}$ connects v_1 and v_3 and $e_3 - \overline{\mathcal{J}}$ connects v_1 and v_4, cf. Figure 14.

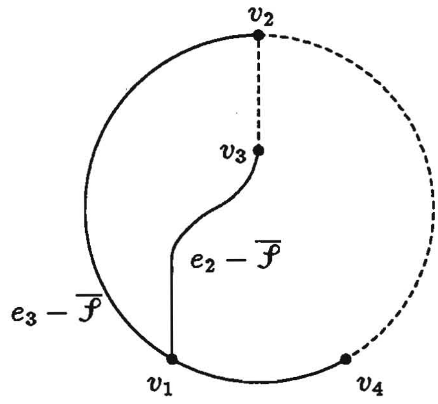


Figure 14: $V(R) \cap \overline{\mathcal{F}}$ is shown dashed.

Lemma 6 *Every vertex v of $V(R \cup \{s\})$ is contained in $V_{unch} \cup V_{chang} \cup V_{new}$.*

Proof: For every vertex v of $V(R \cup \{s\})$ there are two sites p and q different from s such that an edge e' of $V(R \cup \{s\})$ separating p- and q–region is incident to v. Also, there is an edge e of $V(R)$ with $e' \subseteq e$. Thus v is either a vertex of $V(R)$ or v lies on edge e of $V(R)$. In the latter case, v is an endpoint of $e - \overline{\mathcal{F}}$ and hence $v \in V_{new}$. In the former case, e is not clipped at v by s and hence $v \in V_{unch} \cup V_{chang}$. □

Lemma 7 *Let $v \in V_{unch}$. Then $U \cap V(R) = U \cap V(R \cup \{s\})$ for all sufficiently small neighborhoods U of v; in particular, v is a vertex of $V(R \cup \{s\})$.*

Proof: If $v \in V_{unch}$ then no edge of $V(R)$ incident to v is clipped at v by s. Thus $U \cap V(R) \cap \overline{\mathcal{F}} \subseteq v$ for all sufficiently small neighborhoods of v. Lemma 1 thus implies $U \cap V(R) \cap \overline{\mathcal{F}} = \emptyset$ and hence $U \cap V(R) = U \cap V(R \cup \{s\})$. □

Lemma 8 *Let $v \in V_{chang}$.*

1. *In the clockwise ordering of edges of $V(R)$ around v, there are edges f'' and f' ($f' = f''$ is possible) such that all edges between f'' and f' (inclusive) are not clipped at v by s and all edges between f' and f'' (exclusive) are clipped at v by s.*

2. *Let e' be the edge following f' and let e'' be the edge preceding f'' in the clockwise ordering of edges of $V(R)$ around v, cf. Figure 15. Let f' and e' border p–region and e'' and f'' border q–region. Then v is a vertex of $V(R \cup \{s\})$ incident to the following edges: all edges between f'' and f' (inclusive), an edge separating p- and s–region, and an edge separating s- and q–region.*

Proof: a) Since some but not all edges incident to v are clipped at v by s, v must lie on *bd* $\mathcal{F}$. Since *bd* $\mathcal{F}$ is a simple closed curve passing through v, the edges clipped at v by s and the edges not clipped at v by s must form contiguous subsequences in the clockwise ordering of edges around v. This proves a).
b) This is an immediate consequence of part a) and the piece of pie fact. □

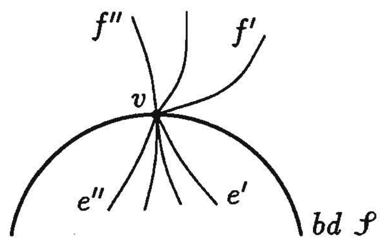


Figure 15: A node $v \in V_{chang}$.

Lemma 9 *Let $v \in V_{new}$ and let v lie on an edge e of $V(R)$ separating p- and q–region. Then v is a vertex of degree three in $V(R \cup \{s\})$. The three edges incident to v separate p- and q-, q- and s-, and s- and p–region respectively.*

Proof: Obvious. □

Example (continued): By Lemmas 6 to 9, the vertices v_2, v_3 and v_4 become incident to two new edges each, cf. Figure 16.

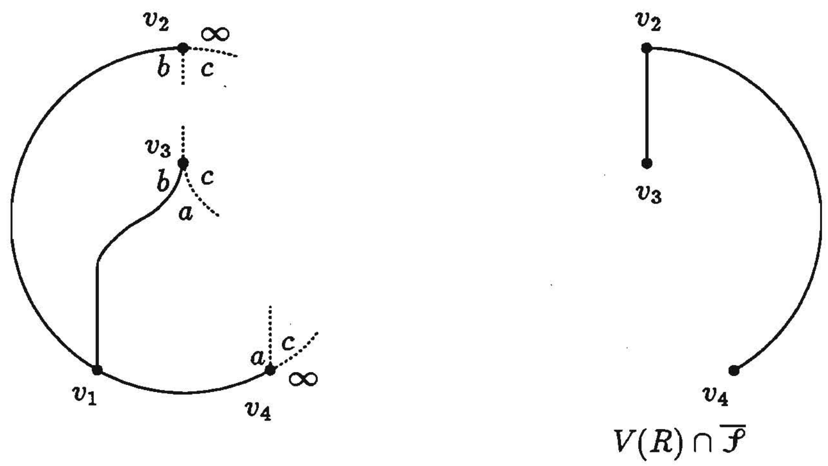


Figure 16: On the left, for each vertex $v \in V_{chang} \cup V_{new}$ the two new edges incident to v are indicated by dots. The deleted part of $V(R)$ is shown on the right.

At this point, we have characterized the vertex set of $V(R \cup \{s\})$ and also the set of edges incident to each vertex of $V(R \cup \{s\})$ in their clockwise ordering around v. It remains to link the two occurrences of each edge. As above, let $\mathcal{E} = V(R) \cap \overline{\mathcal{F}}$.

We know an embedding of $\mathcal{E}$ into the plane. The boundary of the outer face of $\mathcal{E}$ is a closed curve since $\mathcal{E}$ is connected. Also, the vertices on *bd* $\mathcal{F}$ lie on $\mathcal{E}$ and *bd* $\mathcal{F}$ is a simple closed curve. Hence a clockwise traversal of the boundary of the outer face of $\mathcal{E}$ yields the cyclic ordering of the "half-edges" of *bd* $\mathcal{F}$, cf. Figure 17. This allows us to link the two occurrences of each edge. We conclude:

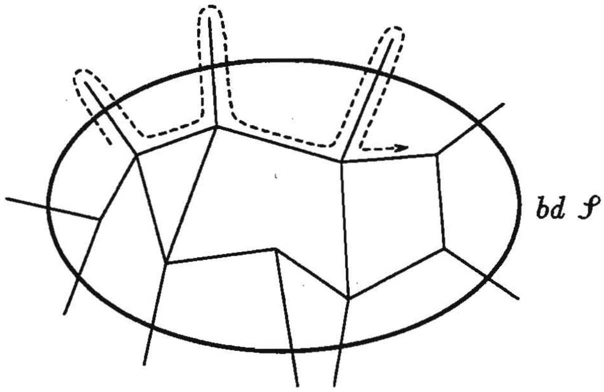


Figure 17: Walking along the boundary of the outer face of $\mathcal{E}$.

Lemma 10 *Given E_s, $V(R \cup \{s\})$ can be constructed from $V(R)$ in time $O(|E_s|)$.*

Proof: Given E_s, one can determine the sets V_{del}, V_{chang}, and V_{new} in time $O(|E_s|)$. In the same time bound, one can update the cyclic adjacency lists of these vertices. Finally the traversal of $\mathcal{E}$ takes time $O(|E_s|)$. □

Example (continued): The clockwise march around $\mathcal{E}$ and joining the two occurrences of each edge yields $V(R \cup \{c\})$ as shown in Figure 7.

Step 3: Computation of $\mathcal{H}(R \cup \{s\})$

We first characterize the set of vertices $\mathcal{H}(R \cup \{s\})$ which are not already vertices of $\mathcal{H}(R)$. Call an edge e of $V(R \cup \{s\})$ *new* if it is not a subset of any edge of $V(R)$, *shortened* if it is a proper subset of some edge of $V(R)$, *affected* if e is an edge of $V(R)$ and there is a vertex $v \in V_{chang}$ such that e is one of the edges f' or f'' defined in Lemma 8, cf. Figure 18.

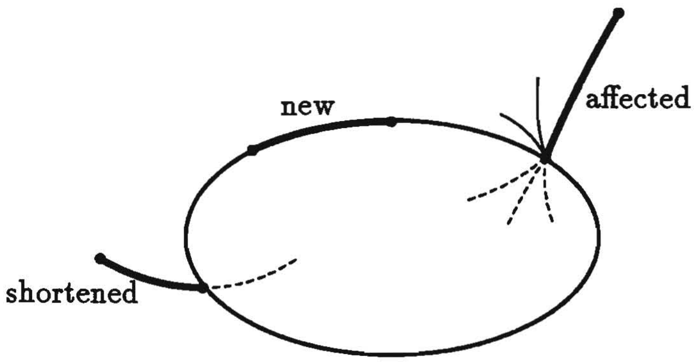


Figure 18: A characterization of edges.

Lemma 11 *Let $V_{\mathcal{H}}(R)$ and $V_{\mathcal{H}}(R \cup \{s\})$ be the vertex sets of $\mathcal{H}(R)$ and $\mathcal{H}(R \cup \{s\})$ respectively. Then $V_{\mathcal{H}}(R \cup \{s\}) - V_{\mathcal{H}}(R) = \{D_{R \cup \{s\}}(e) \mid$ e is a new, shortened, or affected edge of $V(R \cup \{s\})\}$.*

Proof: Let e be an edge of $V(R \cup \{s\})$ which is neither new, shortened, nor affected. Then e has already been an edge of $V(R)$ and hence its endpoints must lie in $V_{unch} \cup V_{chang}$. Also, if an endpoint v of e belongs to V_{chang} then e lies strictly between the edges f'' and f' defined in Lemma 8. Thus e's descriptions with respect to $V(R)$ and $V(R \cup \{s\})$ are identical.

Conversely, if e is new, shortened or affected, then s contributes to e's description D and therefore D cannot be contained in $V_{\mathcal{H}}(R)$. □

We next discuss which edges have to be added to $\mathcal{H}(R \cup \{s\})$ in order to maintain the history-graph invariants.

Lemma 12 *Let e be a shortened or affected edge of $V(R \cup \{s\})$, let e' be the edge of $V(R)$ with $e \subseteq e'$, and let $t \in S - R - \{s\}$ intersect e with respect to $R \cup \{s\}$. Then t intersects e' with respect to R.*

Proof: The lemma follows immediately from $e \cap \overline{VR(t, R \cup \{s,t\})} \subseteq e' \cap \overline{VR(t, R \cup \{t\})}$. □

Thus for each shortened or affected edge e we add the edge $(D_R(e'), D_{R \cup \{s\}}(e))$ to the history graph, where e' is the edge of $V(R)$ with $e \subseteq e'$.

For a new edge e of $V(R \cup \{s\})$ the situation is more complicated. We show that it is sufficient to make e a child of all edges traversed during e's construction. To this end, let x_1 and x_2 be the endpoints of e, and let $p \in R$ be such that e separates p– and s–region in $V(R \cup \{s\})$. By Lemma 1 there must be a path $\mathcal{P}$ in $V(R) \cap \overline{\mathcal{F}}$ connecting x_1 to x_2. Without loss of generality we may assume that $\mathcal{P}$ is part of $bd\ VR(p, R)$. $\mathcal{P}$ is the part of $V(R) \cap \overline{\mathcal{F}}$ traversed during the construction of e. Furthermore define the edges e_1 and e_2 of $V(R)$ as follows. If $x_1 \in V_{new}$, then let e_1 be the edge of $bd\ VR(p, R)$ containing x_1. If $x_1 \in V_{chang}$, then let e_1 be the edge of $bd\ VR(p, R)$ incident to x_1 and not contained in $\mathcal{P}$. The edge e_2 is defined analogously with respect to x_2. The reader may think of e_1 and e_2 as prolongations of $\mathcal{P}$ outside $\overline{\mathcal{F}}$. See Figure 19 for an illustration of these definitions.

Lemma 13 *Let e, e_1, e_2 and $\mathcal{P}$ be defined as above. Let $t \in S - R - \{s\}$ intersect e w.r.t. $R \cup \{s\}$. Then there is an edge $g \in e_1 \cup \mathcal{P} \cup e_2$ such that t intersects g with respect to R.*

Proof: We assume for the sake of a contradiction that t does not intersect any edge $g \in e_1 \cup \mathcal{P} \cup e_2$. By the definition of e_1 and e_2 there are unique edges e_1' and e_2' of $V(R \cup \{s\})$ such that $e_1' \subseteq e_1$ and $e_2' \subseteq e_2$.

Claim 1 *t does not clip e, e_1' or e_2' at x_1 or x_2 with respect to $R \cup \{s\}$.*

Proof: We first deal with edge e. Assume the contrary, say t clips e at x_1 w.r.t. $R \cup \{s\}$. Assume first that $x_1 \in V_{new}$. Then e_1 is the edge of $V(R)$ containing x_1 and $x_1 \in e_1 \cap \overline{VR(t, R \cup \{s,t\})} \subseteq e_1 \cap \overline{VR(t, R \cup \{t\})}$ and hence t intersects e_1, a contradiction. Assume next that $x_1 \in V_{chang}$. Then $x_1 \in \overline{VR(t, R \cup \{s,t\})} \subseteq \overline{VR(t, R \cup \{t\})}$. In $V(R)$, t thus clips one of the two edges of $bd\ VR(p, R)$ incident to x_1, because $\overline{VR(t, R \cup \{t\})}$ cannot

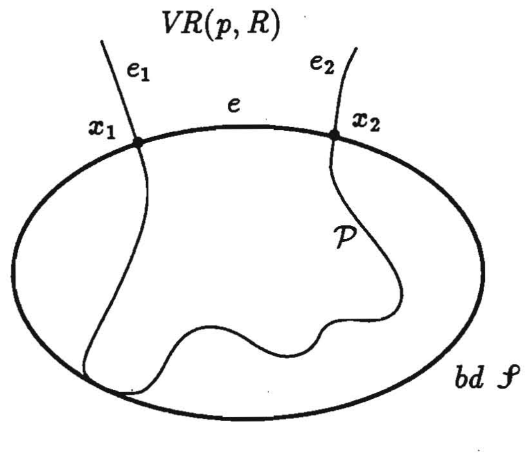


Figure 19: The path $\mathcal{P}$ and edges e_1 and e_2.

contain an isolated point x_1, due to Lemma 1. Since both edges belong to $e_1 \cup \mathcal{P}$, we have a contradiction.
Since $e_1' \subseteq e_1$ and $e_2' \subseteq e_2$ and because t does not intersect e_1 or e_2 by assumption, t cannot clip e_1' or e_2'. □

Claim 2 *$\mathcal{P} \subseteq \overline{\mathcal{f}}$ and there is a point $x \in e \cap \overline{VR(t, R \cup \{s,t\})}$ which does not lie on $\mathcal{P}$.*

Proof: $\mathcal{P} \subseteq \overline{\mathcal{f}}$ holds by definition. Since t intersects e but does not clip e at x_1 or x_2, the intersection $e \cap \overline{VR(t, R \cup \{s,t\})}$ is a non-empty subsegment of e not extending to either endpoint of e. This subsegment must contain a point x not in $\mathcal{P}$ since t does not intersect any edge on path $\mathcal{P}$. □

Now consider the wedge at x_1 formed by e and e_1'. According to the above claim, t does not clip e at x_1 in $V(R \cup \{s\})$. Thus all points in the wedge belong to $VR(p, R \cup \{s,t\})$. The same holds true for the wedge at x_2. Since $VR(p, R \cup \{s,t\})$ is connected, there is a path $\mathcal{Q}$ from x_1 to x_2 running completely inside $VR(p, R \cup \{s,t\}) \subseteq VR(p, R \cup \{t\})$ except at the endpoints, cf. Figure 20. We may assume that $\mathcal{Q}$ does not touch $bd\ \mathcal{f}$ (and therefore x does not lie on $\mathcal{Q}$). Thus x lies in the interior of the cycle $x_1 \circ \mathcal{P} \circ x_2 \circ \mathcal{Q}$; otherwise $VR(p, R)$ would not be simply connected. The point x belongs to $\overline{VR(t, R \cup \{t\})}$. Since $VR(p, R \cup \{t\})$ is simply connected, the region $VR(t, R \cup \{t\})$ cannot be contained in the cycle $x_1 \circ \mathcal{P} \circ x_2 \circ \mathcal{Q}$. Since $\mathcal{Q} \cap \overline{VR(t, R \cup \{t\})} = \emptyset$, we conclude $\mathcal{P} \cap \overline{VR(t, R \cup \{t\})} \neq \emptyset$. The intersection cannot consist of a single point and hence t must intersect an edge $g \in e_1 \cup \mathcal{P} \cup e_2$, a contradiction. This completes the proof of Lemma 13. □

In view of Lemma 13 we add edges $(D_R(e'), D_{R \cup \{s\}}(e))$ for any new edge e of $V(R \cup \{s\})$ and all $e' \in e_1 \cup \mathcal{P} \cup e_2$.

Lemma 14 *The history-graph invariants hold for $\mathcal{H}(R \cup \{s\})$.*

Proof: Part 3) of the history-graph invariant is maintained by Lemmas 12 and 13. Part 2) is trivial. For part 1) first observe that only leaves of $\mathcal{H}(R)$ can get new children. Thus

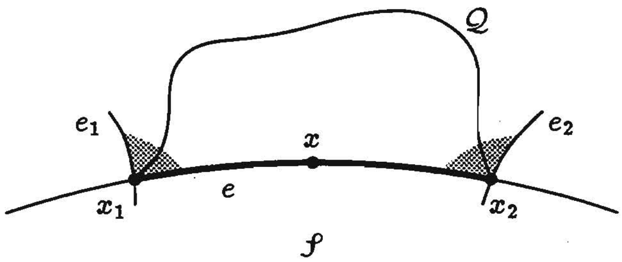


Figure 20: The path $\mathcal{Q}$ and the wedges at x_1 and x_2.

the outdegree of inner nodes of $\mathcal{H}(R)$ does not change. We now show that a leaf of $\mathcal{H}(R)$ gets at most five children. We distinguish several cases. Let e' be an edge of $V(R)$. If e' is also an edge of $V(R \cup \{s\})$ and $D_R(e') = D_{R\cup\{s\}}(e')$, then no edges out of $D_R(e')$ have been added to the history graph. Otherwise, either $e' \subseteq \overline{\mathcal{f}}$ or there is a shortened or affected edge e of $V(R \cup \{s\})$ with $e \subseteq e'$. In the former case, e' belongs to at most two paths $\mathcal{P}$. In the latter case, e' assumes the role of e_1 or e_2 at most four times and is also parent of e, i.e., the outdegree of e' is at most 5. It remains to prove that the descriptions of edges in $V(R \cup \{s\})$ are leaves of $\mathcal{H}(R \cup \{s\})$. This follows from the fact that only those leaves of $\mathcal{H}(R)$ get children that are no longer descriptions of edges of $V(R)$. □

Figure 21 shows a situation where the outdegree is actually 5.

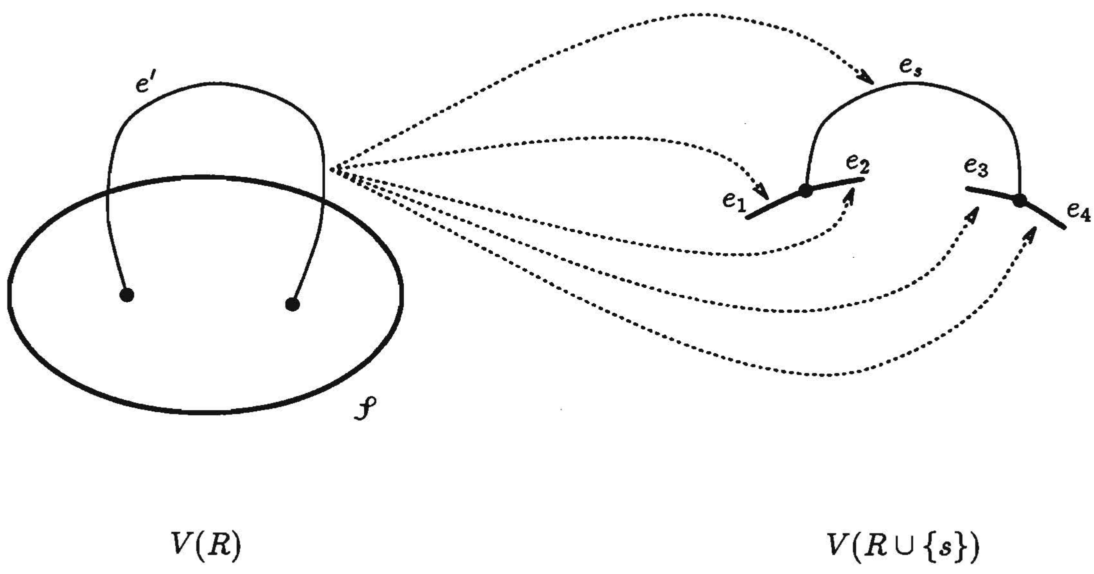


Figure 21: $D_R(e')$ has five children in the history graph. $e_s \subseteq e'$ is a shortened edge. $e_1, \ldots, e_4$ are new edges on $bd\ \mathcal{f}$. Observe that $e_2 \neq e_3$ is possible.

Lemma 15 *Given E_s, $\mathcal{H}(R \cup \{s\})$ can be constructed from $V(R)$ and $\mathcal{H}(R)$ in time $O(|E_s|)$.*

Proof: Following from the discussion above. □

We summarize in:

Theorem 1 *a) Let* $\infty \in R$ *and* $s \in S - R$. *Then* $V(R \cup \{s\})$ *and* $\mathcal{H}(R \cup \{s\})$ *can be constructed from* $V(R)$ *and* $\mathcal{H}(R)$ *in time* $O(c)$ *where* c *is the number of vertices of* $\mathcal{H}(R)$ *intersected by* s.
b) For $R \subseteq S$, $|R| = 3$ *and* $\infty \in R$, *the data structures* $V(R)$ *and* $\mathcal{H}(R)$ *can be set up in time* $O(1)$.

Proof: a) Comprises Lemmas 5, 10 and 15.
b) The Voronoi diagram $V(R)$ for three sites ∞, p and q has the structure shown in Figure 22. The history graph $\mathcal{H}(\{\infty, p, q\})$ for these three sites simply consists of a node for the source and each of the three edges of $V(\{\infty, p, q\})$. The descriptions of the 3 edges of $V(\{\infty, p, q\})$ are made children of the source. Both structures can certainly be set up in time $O(1)$. □

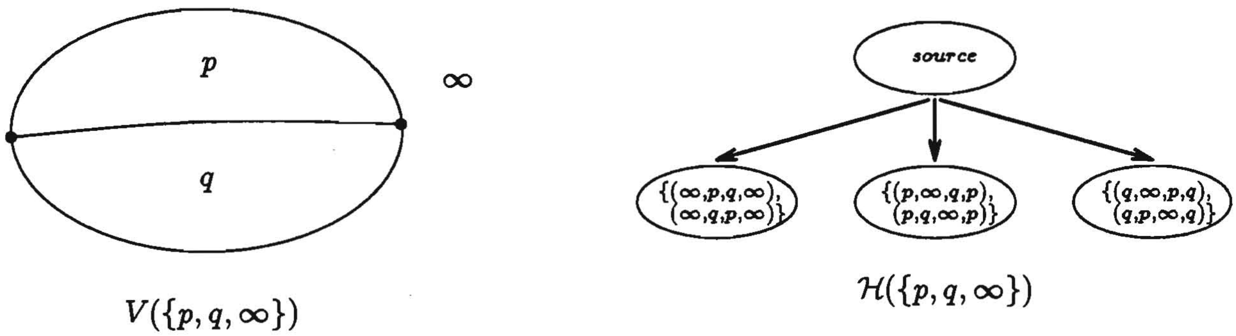


Figure 22: Initialization of Voronoi diagram and history graph.

5 Analysis

The analysis of randomized incremental algorithms is always done in terms of objects, regions and conflicts between them. In our case the objects are the sites and the regions are descriptions.

Definition 6 *Let* $R \subseteq S$.

1. *A* description D *over* R *is a set* $\{(r_q, q, p, r_p), (t_p, p, q, t_q)\}$, *where* $\{p, q, r_p, r_q, t_p, t_q\} \subseteq R$, *and* $V(\{p, q, r_p, r_q, t_p, t_q\})$ *contains a bounded edge with description* D. $\mathcal{F}(R)$ *denotes the set of all descriptions over* R *and* $set(D)$ *denotes the set* $\{p, q, r_p, r_q, t_p, t_q\}$.

2. *Let* D *be a description over* R *and let* $s \in S - set(D)$ *be a site. Site* s conflicts with D *iff there is no bounded edge in* $V(set(D) \cup \{s\})$ *with description* D. *Define* $\mathcal{F}_0(R) = \{D \in \mathcal{F}(R) \mid D$ *does not conflict with any site* $s \in R - set(D)\}$.

If a site s intersects a description D, then it also conflicts with description D. The converse is not true. Namely, if in Figure 13 a site s clips any of the edges g_p, g_q, f_p, f_q at their common endpoint with e, then s conflicts with $D_R(e)$. The sole motivation for defining the notion of conflict is the following bijection lemma. It makes the general results about randomized incremental constructions available for the analysis of our algorithm.

Lemma 16 (bijection lemma) *Let $\infty \in R \subseteq S$. Then $e \mapsto D_R(e)$ is a bijection between the edges of $V(R)$ and the descriptions in $\mathcal{F}_0(R)$.*

Proof: Note first that all edges of $V(R)$ are bounded since $\infty \in R$. Let e be an edge of $V(R)$, let $D = D_R(e)$, and let $s \in R - set(D)$. By the basic operation lemma, e is also an edge of $V(set(D))$ and $V(set(D) \cup \{s\})$. The same argument shows that the description of e in $V(set(D))$ and $V(set(D) \cup \{s\})$ is still D. In particular s does not conflict with D, i.e., $D \in \mathcal{F}_0(R)$. We have now shown that the mapping $e \mapsto D_R(e)$ maps the set of edges of $V(R)$ into $\mathcal{F}_0(R)$. The mapping is injective since the existence of two different edges with the same description clearly contradicts Lemma 3. It remains to show surjectivity.
Let $D \in \mathcal{F}_0(R)$ be arbitrary. We show that there is an edge in $V(R)$ with description D. Assume the contrary. Then there are a set R', $set(D) \subseteq R' \subseteq R$, and a site $s \in R - R'$ such that $V(R')$ contains an edge e with description D, but $V(R' \cup \{s\})$ does not. Since $D \in \mathcal{F}_0(R)$ there is an edge with description D in $V(set(D))$ as well as in $V(set(D) \cup \{s\})$. By the basic operation lemma both edges are equal to edge e. We now consider e with respect to $V(R')$ and distinguish several cases according to whether or not s intersects e w.r.t. R'. Thus let $e \cap \overline{VR(s, R' \cup \{s\})} \neq \emptyset$. By the basic operation lemma we have $e \cap \overline{VR(s, set(D) \cup \{s\})} \neq \emptyset$ in contradiction to the claim that e is edge of $V(set(D) \cup \{s\})$. Thus let now $e \cap \overline{VR(s, R' \cup \{s\})} = \emptyset$. Then e is also edge of $V(R' \cup \{s\})$. But then the description of e w.r.t. $R' \cup \{s\}$ must be different from D, say D'. By the basic operation lemma e then also has description D' in $V(set(D) \cup \{s\})$, a contradiction. □

Let $s_1, s_2, \ldots, s_n$ be the sequence in which the algorithm processes the sites and let $R_r = \{s_1, s_2, \ldots, s_r\}$, for $1 \leq r \leq n$. The bijection lemma provides an alternative characterization of the vertex set of the history graph.

Lemma 17 *The set of nodes of $\mathcal{H}(R_r)$ equals $\{source\} \cup \bigcup_{3 \leq i \leq r} \mathcal{F}_0(R_i)$.*

Proof: Obvious. □

Lemma 17 characterizes the vertex set of the history graph as a set of combinatorial objects defined by a small number of input sites. We can therefore apply the results of [CS89, BDS+92, CMS92] to the analysis of our algorithm. To do so assume that the algorithm processes the sites in random order. [CS89, BDS+92, CMS92] give bounds on the expected size of the history graph and the number of its vertices in conflict with a input site in terms of f_r, the expected size of $\mathcal{F}_0(R)$ for a random subset $R \subseteq S$, $|R| = r$.

Lemma 18 ([CMS92], Theorems 3 and 4) *1. The expected size of $\mathcal{H}(R_r)$ is $O(\sum_{i \leq r} \frac{f_i}{i})$.*

2. The expected number of vertices of $\mathcal{H}(R_{r-1})$ in conflict with site s_r is $O(\sum_{i \leq r} \frac{f_i}{i(i-1)})$.

Since a Voronoi diagram of i sites has at most $3i - 6$ edges the bijection lemma implies $f_i = O(i)$.

Theorem 2 *An abstract Voronoi diagram of n sites can be computed by a randomized algorithm in expected time $O(n \log n)$ and expected space $O(n)$. Moreover, the expected time for inserting the r-th object is $O(\log r)$. Randomization here only concerns the order in which the sites are inserted.*

Proof: At any time the size of the history graph clearly dominates the size of the Voronoi diagram. Thus the expected space used by the algorithm is $O(n)$ by Lemma 18.
By Theorem 1 the time needed to insert the r-th site s_r is proportional to the number c_r of vertices in $\mathcal{H}(R_{r-1})$ which are intersected by s_r. Since each intersection implies a conflict we have $c_r = O(\log r)$ by Lemma 18. This yields the claimed time bounds. □

6 Simple Voronoi diagrams

In this section we introduce a subclass of abstract Voronoi diagrams for which the basic operation can be replaced by two operations each requiring only four sites as input.

Definition 7 *A system of bisectors is called* simple *if for any three (finite) sites the induced Voronoi diagram contains at most one vertex.*

Observe that in general three sites p, q and r can produce two vertices, a pqr- and a prq-vertex, see Figure 3 for an example. Simple systems of bisectors are generated, for instance, by point sites under the Euclidean metric or under the L_1-metric (as defined by Lee [Lee80]), and by Powerdiagrams (see [Aur87]).

Now consider a Voronoi diagram $V(R)$, $\infty \in R \subseteq S$, generated by a simple system of bisectors. As in the previous sections let $s \in S - R$. We investigate again the type of intersection between an edge of $V(R)$ and site s. For this investigation edges on Γ must be treated separately.

To this purpose, let e be a $prqt$–edge of $V(R)$ not on Γ, i. e., $p, q \neq \infty$. Since $V(\{p,q,s\})$ contains at most one vertex, cases 2d and 3 of our basic operation are excluded. Furthermore cases 1, 2a, 2b, and 2c can be distinguished simply by deciding whether s clips e at its prq- or qtp–endpoint.

$$basic\_op(p,r,q,t,s) = \begin{cases} 1 & \text{iff} \quad s \text{ does not clip } e \text{ at either endpoint} \\ 2a & \text{iff} \quad s \text{ clips } e \text{ at both endpoints} \\ 2b & \text{iff} \quad s \text{ clips } e \text{ only at the } prq\text{–endpoint} \\ 2c & \text{iff} \quad s \text{ clips } e \text{ only at the } qtp\text{–endpoint} \end{cases}$$

However, clipping can be decided by looking at only four sites:

Lemma 19 *Let e be a $prqt$–edge of $V(R)$ and let $s \in S - R$. Let e' be the unique edge of $V(\{p,q,r\})$ incident to the prq–vertex of $V(\{p,q,r\})$ and separating p–region from q–region. Then $e \subseteq e'$ and s clips e at the prq–endpoint w.r.t. R iff s clips e' at the prq–endpoint w.r.t. $\{p,q,r\}$.*

Proof: First observe that by the basic operation lemma e is also a $prqt$–edge of $V(\{p,q,r,t\})$ and that $e \cap \overline{VR(s,\{p,q,r,s,t\})} = e \cap \overline{VR(s, R \cup \{s\})}$. By removing t from $\{p,q,r,t\}$ edge e cannot shrink, but possibly grow at its qtp–endpoint. Thus $e \subseteq e'$. For the remaining claim note that t has no influence on whether s clips e at its prq–endpoint or not. □

Thus, for edges not on Γ the basic operation on five sites is reduced to a four sites clipping operation.

Let us now turn to edges on Γ. For these edges cases 2d and 3 of our basic operation are also possible. Thus the basic operation cannot be reduced to clipping as before. However, for all these edges, one of the four "defining" sites is always ∞. Moreover, if e is a $prqt$–edge on Γ the either $p = \infty$ or $q = \infty$. Since each $prqt$–edge is also a $qtpr$–edge, we can assume $p = \infty$. Computing the outcome of the basic operation for a $prqt$–edge on Γ can thus be handled by a special *basic_op_Γ* operation that inputs only the four sites q, r, t and s.

To give an impression of the amount of programming hidden inside the basic operation, we sketch the implementation for a Voronoi diagram of points under the Euclidean metric. Let e be a $prqt$–edge not on Γ, and furthermore let $r \neq \infty$. Then s clips e at its prq–endpoint iff s lies inside the circumcircle of p, r and q, or s lies on the circumcircle of p, r and q and p, q and s form a rightturn. If $r = \infty$ then e is an "unbounded" edge of $V(R - \{\infty\})$. The

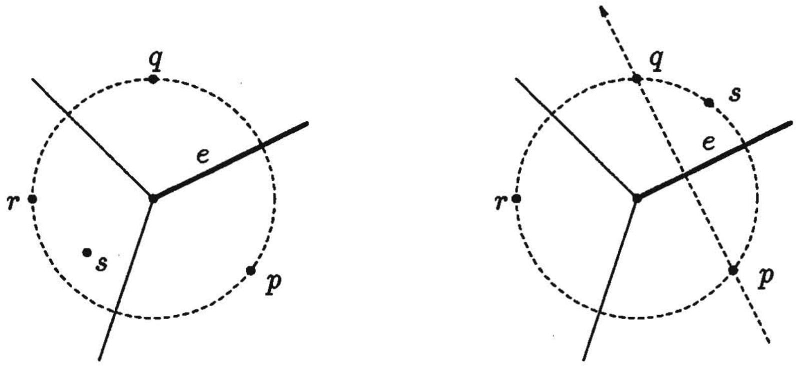


Figure 23: s clips e at the prq–endpoint.

circumcircle of p, r and q then becomes the "infinite circle" through p, ∞ and q, i.e., the line through p and q; point s lies inside the "infinite circle" iff s lies to the right of the line through p and q directed from p to q. Furthermore, if s lies on the line through p and q then s clips e iff s lies between p and q.

Let e now be an ∞rqt–edge on Γ. Cases 2d and 3 of our basic operation can occur only if $r = t$ and q, r and s are collinear, see Figure 24. If $r \neq t$ or q, r and s are not collinear then the outcome of the basic operation is once again completely determined by the way s clips e at its endpoints. Here s clips e at its ∞rq–endpoint iff r, q and s form a rightturn, or s lies on the line through r and q between r and q.

The test whether a point lies inside, on, or outside the circumcircle of three other points, and the test whether three points are collinear, or form a left- or rightturn are fundamental

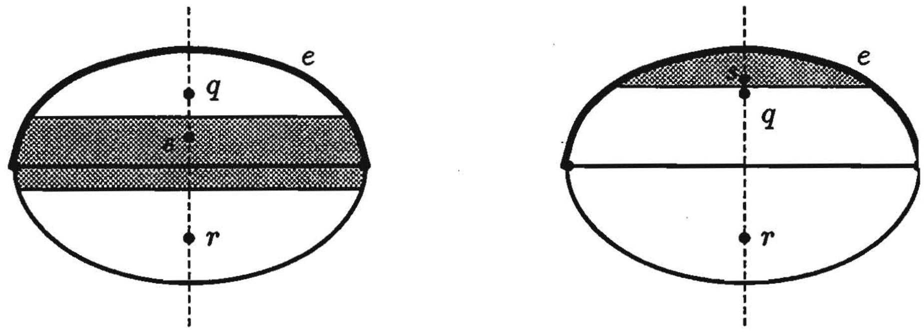


Figure 24: Site s intersects an edge e on Γ.

tests in computational geometry. Observe that all algorithms which use only the incircle-test do not handle four cocircular or three collinear points.

For Powerdiagrams the implementation is very similar, for diagrams of points under the L_1-metric it is more involved.

7 Conclusion

We have shown that the construction of abstract Voronoi diagrams can be reduced efficiently and purely combinatorially to the construction of abstract Voronoi diagrams for five sites, respectively four sites in some cases. This is also true for furthest site abstract Voronoi diagrams, see [MMR92]. Many previously considered types of Voronoi diagrams can thus be handled by the same simple algorithm.

References

[Aur87] F. Aurenhammer. Power diagrams: Properties, algorithms and applications. *SIAM Journal on Computing*, 16(1):78–96, 1987.

[Aur91] F. Aurenhammer. Voronoi diagrams — A survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23(3):345–405, 1991.

[BD89] J.D. Boissonnat and M. Devillers-Teillaud. On the randomized construction of the Delaunay tree. Technical Report 1140, INRIA Sophia-Antipolis, 1989.

[BDS+92] J.D. Boissonnat, O. Devillers, R. Schott, M. Teillaud, and M. Yvinec. Applications of random sampling to on-line algorithms in computational geometry. *Discrete and Computational Geometry*, 8:51–71, 1992.

[BDT90] J.D. Boissonnat, O. Devillers, and M. Teillaud. An on-line construction of higher order Voronoi diagrams and its randomized analysis. In J. Urrutia, editor, *Proc. of the 2nd Canadian Conference on Computational Geometry*, pages 278–281, Ottawa, 1990.

[Bro79] K.Q. Brown. Voronoi diagrams from convex hulls. *Information Processing Letters*, 9:223–228, 1979.

[CMS92] K.L. Clarkson, K. Mehlhorn, and R. Seidel. Four results on randomized incremental constructions. Technical Report MPI-I-92-112, Max-Planck-Institut für Informatik, 1992. A preliminary version was presented at the 9th Symposium on Theoretical Aspects of Computer Science (STACS '92), LNCS 577, pages 463–474.

[CS89] K.L. Clarkson and P.W. Shor. Applications of random sampling in computational geometry, II. *Discrete and Computational Geometry*, 4:387–421, 1989.

[ES86] H. Edelsbrunner and R. Seidel. Voronoi diagrams and arrangements. *Discrete and Computational Geometry*, 1:25–44, 1986.

[For87] S. Fortune. A sweepline algorithm for Voronoi diagrams. *Algorithmica*, 2(2):153–174, 1987.

[GKS92] L.J. Guibas, D.E. Knuth, and M. Sharir. Randomized incremental construction of Delaunay and Voronoi diagrams. *Algorithmica*, 7(4):381–413, 1992.

[GS85] L.J. Guibas and J. Stolfi. Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *ACM Transactions on Graphics*, 4(2):74–123, 1985.

[Kle88a] R. Klein. Abstract Voronoi diagrams and their applications (extended abstract). In H. Noltemeier, editor, *Proc. of the Workshop on Computational Geometry and its Applications*, pages 148–157, Würzburg, 1988. LNCS 333.

[Kle88b] R. Klein. Voronoi diagrams in the Moscow metric (extended abstract). In J. van Leeuwen, editor, *Proc. of the Workshop on Graphtheoretic Concepts in Computer Science*, pages 434–441, Amsterdam, 1988. LNCS 344.

[Kle89a] R. Klein. Combinatorial properties of abstract Voronoi diagrams. In M. Nagl, editor, *Proc. of the Workshop on Graphtheoretic Concepts in Computer Science*, pages 356–369, Rolduc, 1989. LNCS 411.

[Kle89b] R. Klein. *Concrete and Abstract Voronoi Diagrams.* Springer Verlag, 1989. LNCS 400.

[Lee80] D.T. Lee. Two-dimensional Voronoi diagrams in the L_p-metric. *Journal of the ACM*, 27(4):604–618, 1980.

[LS86] D. Leven and M. Sharir. Intersection and proximity problems and Voronoi diagrams. In J. Schwartz and C.K. Yap, editors, *Advances in Robotics, Volume 1*, pages 187–228. Lawrence Erlbaum, 1986.

[MMO91] K. Mehlhorn, S. Meiser, and C. Ó'Dúnlaing. On the construction of abstract Voronoi diagrams. *Discrete and Computational Geometry*, 6:211–224, 1991.

[MMR92] K. Mehlhorn, S. Meiser, and R. Rasch. Furthest site abstract Voronoi diagrams. Technical Report MPI-I-92-135, Max-Planck-Institut für Informatik, 1992.

[SH75] M.I. Shamos and D. Hoey. Closest point problems. In *Proc. of the 16th Annual IEEE Symposium on Foundations of Computer Science*, pages 151–162, 1975.

[Zim92] N. Zimmer. Die Konstruktion abstrakter Voronoidiagramme. Master's thesis, Universität des Saarlandes, 1992. (In German).