New Results on Abstract Voronoi Diagrams

Dissertation

Zur
Erlangung des Doktorgrades (Dr. rer. nat.)
der
Mathematisch-Naturwissenschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Cecilia Bohler

aus Leonberg

Bonn, 2015

Angefertigt mit der Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn

1. Gutachter: Prof. Dr. Rolf Klein, Universität Bonn
2. Gutachter: Prof. Dr. Heiko Röglin, Universität Bonn

Tag der Promotion: 10.07.2015
Erscheinungsjahr: 2015

Acknowledgment

This thesis was created during my employment at the Institute of Computer Science I at the University of Bonn. Most of my time there, I was funded by the European Science Foundation (ESF) and the German Research Foundation (DFG) in the EUROCORES collaborative research project EuroGIGA/VORONOI.

Most of all I want to thank my advisor Prof. Dr. Rolf Klein for offering me this position in his workgroup, introducing me to the field of computational geometry, and providing me the possibility to write my thesis about abstract Voronoi diagrams. I want to thank all my colleagues, co-authors and anonymous referees, for helpful discussions, many constructive comments, and fruitful research collaborations. Also I am thankful to the other members of the committee, especially Prof. Dr. Heiko Röglin who agreed to be the co-referee of this thesis. I want to thank Antje Bertram for her friendly nature and her continual helpfulness regarding all kinds of administrative tasks. Last but not least many thanks go to my boyfriend Dennis Hansen who always encouraged and helped me in so many ways.

Abstract

Voronoi diagrams are a fundamental structure used in many areas of science. For a given set of objects, called sites, the Voronoi diagram separates the plane into regions, such that points belonging to the same region have got the same nearest site. This definition clearly depends on the type of given objects, they may be points, line segments, polygons, etc. and the distance measure used. To free oneself from these geometric notions, Klein introduced *abstract Voronoi diagrams* as a general construct covering many concrete Voronoi diagrams. Abstract Voronoi diagrams are based on a system of bisecting curves, one for each pair of abstract sites, separating the plane into two dominance regions, belonging to one site each. The intersection of all dominance regions belonging to one site p defines its Voronoi region. The system of bisecting curves is required to fulfill only some simple combinatorial properties, like Voronoi regions to be connected, the union of their closures cover the whole plane, and the bisecting curves are unbounded. These assumptions are enough to show that an abstract Voronoi diagram of n sites is a planar graph of complexity $O(n)$ and can be computed in expected time $O(n \log n)$ by a randomized incremental construction.

In this thesis we widen the notion of abstract Voronoi diagrams in several senses. One step is to allow disconnected Voronoi regions. We assume that in a diagram of a subset of three sites each Voronoi region may consist of at most s connected components, for a constant s, and show that the diagram can be constructed in expected time $O\left(s^2 n \sum_{j=3}^{n} \frac{m_j}{j}\right)$, where m_j is the expected number of connected components of a Voronoi region over all diagrams of a subset of j sites. The case that all Voronoi regions are connected is a subcase, where this algorithm performs in optimal $O(n \log n)$ time, because here $s = m_j = 1$.

The next step is to additionally allow bisecting curves to be closed. We present an algorithm constructing such diagrams which runs in expected time $O\Big(s^2 n \log(\max\{s, n\}) \sum_{j=2}^{n} \frac{m_j}{j}\Big)$. This algorithm is slower by a $\log n$-factor compared to the one for disconnected regions and unbounded bisectors. The extra time is necessary to be able to handle special phenomenons like islands, where a Voronoi region is completely surrounded by another region, something that can occur only when bisectors are closed. However, this algorithm solves many open problems and improves the running time of some existing algorithms, for example for the farthest Voronoi diagram of n simple polygons of constant complexity.

Another challenge was to study *higher order abstract Voronoi diagrams.* In the concrete sense of an order-k Voronoi diagram points are collected in the same Voronoi region, if they have the same k nearest sites. By suitably intersecting the dominance regions this can be defined also for abstract Voronoi diagrams. The question arising is about the complexity of an order-k Voronoi diagram. There are $\binom{n}{k}$ subsets of size k but fortunately many of them have an empty order-k region. For point sites it has already been shown that there can be at most $O(k(n-k))$ many regions and even though order-k regions may be disconnected when considering line segments, still the complexity of the order-k diagram remains $O(k(n-k))$. The proofs used to show this strongly depended on the geometry of the sites and the distance measure, and were thus not applicable for our abstract higher order Voronoi diagrams. Nevertheless, we were able to

come up with proofs of purely topological and combinatorial nature of Jordan curves and certain permutation sequences, and hence we could show that also the order-k abstract Voronoi diagram has complexity $O(k(n-k))$, assuming that bisectors are unbounded, and the order-1 regions are connected.

Finally, we discuss Voronoi diagrams having the shape of a tree or forest. Aggarwal et. al. showed that if points are in convex position, then given their ordering along the convex hull, their Voronoi diagram, which is a tree, can be computed in linear time. Klein and Lingas have generalized this idea to Hamiltonian abstract Voronoi diagrams, where a curve is given, intersecting each Voronoi region with respect to any subset of sites exactly once. If the ordering of the regions along the curve is known in advance, all Voronoi regions are connected, and all bisectors are unbounded, then the abstract Voronoi diagram can be computed in linear time. This algorithm also applies to diagrams which are trees for all subsets of sites and the ordering of the unbounded regions around the diagram is known. In this thesis we go one step further and allow the diagram to be a forest for subsets of sites as long as the complete diagram is a tree. We show that also these diagrams can be computed in linear time.

Contents

Chapter 1

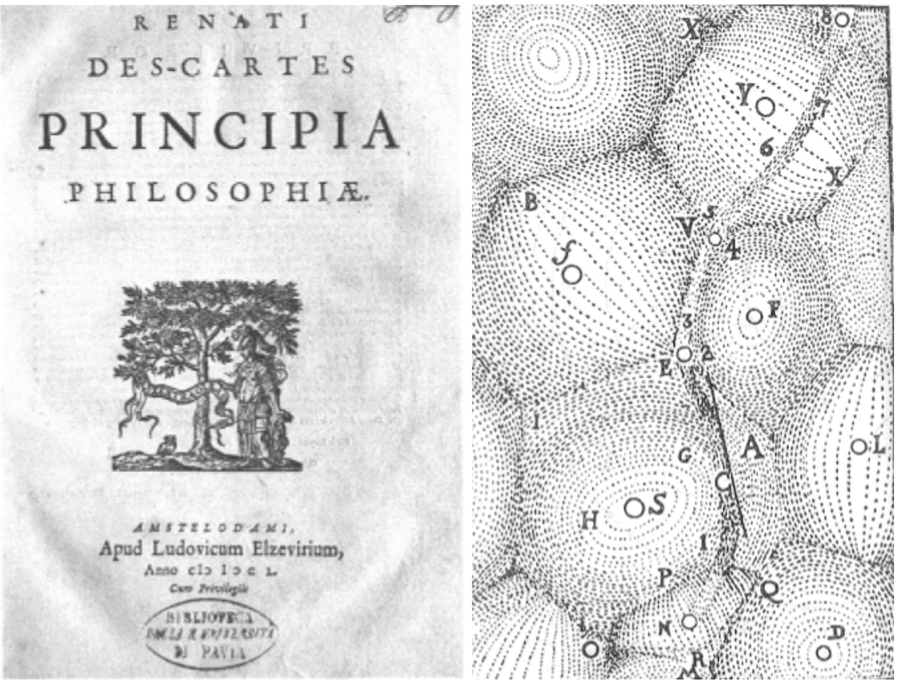
Introduction

During the 16th century the french philosopher and mathematician René Descartes was watching the clear sky. He saw stars and planets each surrounded by a cloud of matter in circular formations. He explained this phenomenon by a vortex-theory in his book "Principia Philosophia" [32] which was published in Amsterdam in 1644. He could not imagine space to be empty, thus there must be some substance, which he called Aether, filling it up. Because the Aether is packed so tight it can not move freely, but only in circular or vortex formations. This results in centrifugal forces, where light matter moves to the outside of a vortex and heavy matter, like a star or planet, stays in the middle. Where two vortices meet there appears a straight line separating them and where three vortices meet there appears a vertex, see Figure 1.1. What Descartes did not know at this time is that even though his vortex theory turned out to be wrong, the figure he drew is the first known *Voronoi diagram.*

These diagrams received further attention and a more precise mathematical definition and analysis 200 years later, first by the German mathematician Peter Gustav Lejeune Dirichlet [34], and the American meteorologist Alfred Henry Thiessen [70], and later by the Ukrainian mathematician Georgi Feodosjewitsch Voronoi [71, 72]. Dirichlet and Voronoi used the structure of the diagram to study quadratic forms, and Thiessen used it to compute the precipitation average over large areas. That is why in the literature the diagrams are also called *Dirichlet tesselation* or *Thiessen polygons.* Here some point shaped sites are given in the space, which is partitioned into regions, where points belonging to the same region have got the same nearest site, and distance is measured by the Euclidean metric.

In 1975 Michael Shamos and Dan Hoey published a paper at the FOCS conference entitled "Closest-point problems". They presented six challenging geometric problems for which no algorithm with optimal run time existed and showed that they can be solved much faster using a single structure, the Voronoi diagram. For example the computation of all nearest neighbors of a set of n points in the plane can be solved in linear time once the Voronoi diagram is given. The preprocessing time to compute the Voronoi diagram is $O(n \log n)$, which results in an optimal running time for the *All-Nearest-Neighbor-Problem.* More of these problems will be discussed in Section 2.1.2.

What nobody could foresee at this time was that a new area of computer science was born, called Computational Geometry. Voronoi diagrams turned



RENATI
DES-CARTES
PRINCIPIA
PHILOSOPHIÆ.

AMSTELODAMI,
Apud Ludovicum Elzevirium,
Anno cIↄ Iↄ c L.
Cum Privilegio

BIBLIOTECA
DELLA UNIVERSITÀ
DI PAVIA

Figure 1.1: Explaining his vortex theorie, René Descartes drew the first known Voronoi diagram [32].

out to be a fundamental structure in this area and can be used to solve many geometric problems.

Geometric problems and hence also Voronoi diagrams have applications in a wide range of scientific areas, reaching from physiology, biology, geology, geography, meteorology, physics, crystallography, and mathematics to computer science. Voronoi diagrams can represent cell structures, the distribution of light around stars, the cracks of drying mud, land conquered by different kings, and a good approximation for 3-dimensional surfaces.

The requests of the applications can vary in many ways. If you look at your skin, then you will see that not all cells have the same size, also the stars whose size influences on their luminosity can vary, the mud may not have been equal wet at all places, the kings may have different strengths, and start with different advances, and of course surfaces can be as diverse as you wish.

To meet all these conditions many different Voronoi diagrams have been introduced. They vary in their given set of sites, they need not be point shaped but may resemble line segments, polygons, or even more complex two-dimensional sets, and the distance function to measure the distance from an arbitrary point to a site can be anything from L_p-metrics to convex distance functions.

Because there are so many different ways to define a Voronoi diagram the effort put into analyzing them and finding efficient algorithms was enormous. That is why the temptation was high to come up with a unifying concept covering as many Voronoi diagrams as possible.

One elegant approach was taken by Edelsbrunner and Seidel [39], who did not explicitly name any concrete sites and distance measures, but defined n

surfaces in $\mathbb{R}^3$, one for each indexed site $p, q, \ldots$. The corresponding Voronoi diagram was now defined as the lower envelope of the surfaces projected to the plane, also called the *minimization diagram* of the surfaces. In the euclidean case each surface would represent a cone with an interior angle of 90° rising vertically from the point-site p.

Independently, Klein [48] took a different approach and came up with *abstract Voronoi diagrams*, AVD's for short. Again the idea was to develop a theory and an algorithm computing Voronoi diagrams, independently of which types of sites and distance functions it represents. He succeeded by detecting that most Voronoi diagrams have one feature in common, that is for each pair of sites there exists a bisecting curve consisting of all points having the same distance to both sites, and these curves bear enough information to construct the complete Voronoi diagram. Thus, instead of having sites and distance measures as input, Klein used a set of bisecting curves, one curve for each pair of sites. Of course, these curves have to fulfill some properties, e. g., they should be homeomorphic to a line, and each pair of them may intersect in only a finite number of points. Also for each triple of sites the Voronoi regions should be connected and their closures should cover the whole plane, that is, each point of the plane either lies on the Voronoi diagram or is contained in the interior of a Voronoi region.

With these assumptions Klein [48] was able to show that the abstract Voronoi diagram can be represented as a planar graph in a natural way, with $O(n)$ faces (Voronoi regions), edges (Voronoi edges), and vertices (Voronoi vertices).

Later Klein, Mehlhorn, and Meiser [53] were able to show that the abstract Voronoi digram can be computed in expected time $O(n \log n)$ and expected space $O(n)$. This was great progress, because many concrete Voronoi diagrams are covered by these assumptions, and need no longer a separate analysis. For example, the first optimal algorithm for constructing nearest Voronoi diagrams of disjoint convex objects, and of line segments under the Hausdorff metric was provided by the abstract concept.

The aim of this thesis is to widen the notion of abstract Voronoi diagrams. In Chapter 2 we start with discussing the state of the art. How it all began with euclidean Voronoi diagrams, how they can be computed and which applications they have. Then we continue with some variations of the type of sites and distance measures and finally we define abstract Voronoi diagrams like in [53] and [49], and discuss their important properties.

However, still there are interesting concrete Voronoi diagrams not covered by the current AVD concept. For example when point shaped sites exert influence on their surrounding space, but this influence is not monotonically decreasing when increasing the distance to the site, then Voronoi regions may become disconnected, a phenomenon not covered by Klein's initial AVD concept. In Chapter 3 we show that when in each diagram of a subset of three sites of the given site-set, each Voronoi region has at most s connected components, for a constant s, then the corresponding AVD can be computed in expected time $O\left(s^2 n \sum_{j=3}^{n} \frac{m_j}{j}\right)$ and space $O(\sum_{j=4}^{n} m_j)$. The variable m_j is the expected number of connected components of a Voronoi region over all Voronoi diagrams of a subset of j sites. For the former case where only connected Voronoi regions were allowed and thus $s = 1$, this resulted in the optimal run time $O(n \log n)$

because here $m_j = 1$.

But what if the bisecting curves are not necessarily unbounded any longer. This may happen, e. g., for non convex polygons, or when the influence of some site is stronger than the influence of another one. Then bisecting curves may be closed, and the Voronoi region of a site with low influence may be completely surrounded by the region of a site with high influence, and again Voronoi regions may be disconnected. This has been an open problem for many years, because these Voronoi diagrams turned out to carry devastatingly different features and are much more complicated to handle than the previously introduced AVD's. Nevertheless, we address this wider type of AVD's in Chapter 4 and are able to present an algorithm which runs in expected time $O\Big(s^2 n \log(\max\{s,n\}) \sum_{j=2}^{n} \frac{m_j}{j}\Big)$ and space $O(\sum_{j=3}^{n} m_i)$. Again s is the maximum number of connected components of a Voronoi region in any diagram of 3 sites and m_j is the expected number of connected components of a Voronoi region in any diagram of j sites of the given site-set. For the known cases where regions are connected ($s = m_j = 1$) and all bisecting curves are unbounded this algorithm takes a $\log n$-factor longer. However, it improves the running time for, e. g., the farthest Voronoi diagram of n simple polygons of constant complexity. Here s is constant and the Voronoi diagram is of linear size, thus our algorithm requires time $O(n \log^2 n)$, whereas the previously best known algorithm needed time $O(n \log^3 n)$.

Another Voronoi diagram which was not even yet defined for the abstract setting is the *higher order Voronoi diagram*. Here, each Voronoi region collects points having the same k nearest neighbors among a given set of sites. So far higher order Voronoi diagrams have been considered for points and line segments in L_p metrics only. In the former case Voronoi regions are still connected whereas already for line segments they may be disconnected. The first important question is how many Voronoi regions are nonempty in an order-k Voronoi diagram. There are $\binom{n}{k}$ many subsets of size k but fortunately many of them are empty. Lee [57] showed that for point sites at most $O(k(n-k))$ regions are nonempty. The same holds for line segments, as Papadopoulou and Zavershynskyi [63] were able to show, and even though here a single region may have $\Omega(n)$ connected components, the total number of connected components still remains $O(k(n-k))$. In Chapter 5 we introduce *higher order abstract Voronoi diagrams* and show that also their complexity remains $O(k(n-k))$. To prove this we had to come up with purely topological and combinatorial arguments. Most proofs in the previous papers depended on geometric arguments which are not applicable for our abstract setting. Instead we use the topological nature of Jordan curves and combinatorial properties of certain permutation sequences.

Finally, there are Voronoi diagrams having a certain simpler structure. For example the nearest Voronoi diagram of points in convex position is always a tree. The same holds for line segments bounding a convex polygon and for the farthest Voronoi diagram of point sites (the diagram where points belonging to the same region have got the same *farthest* site). For points in convex position Aggarwal, Guibas, Saxe, and Shor [5] proposed a divide&conquer algorithm computing the Voronoi diagram in linear time, assuming that the ordering of the points along their convex hull is known in advance. The same algorithm applies to the diagram of line segments bounding a convex polygon, which equals the medial axis in the interior of the polygon. Klein and Lingas [50] took a first step towards abstracting the algorithm. They introduced the *Hamiltonian abstract Voronoi diagram*, where an unbounded curve is given intersecting each Voronoi region with respect to any subset of sites exactly once. In addition Voronoi regions were assumed to be connected and bisecting curves unbounded. Given such a Hamiltonian curve together with the ordering of the Voronoi regions along it, they presented an algorithm, free of geometric arguments, computing the Voronoi diagram in linear time. This algorithm also applies to the previous cases. But what if the Voronoi diagram is a forest rather than a tree? In Chapter 6 we consider Voronoi diagrams, where the final diagram is a tree but for subsets of sites it may be a forest. Given the ordering of the unbounded regions of the final diagram we show that it can be computed in linear time.

Chapter 2

From Concrete to Abstract Voronoi Diagrams

Lets pretend it is election day. The votes can be given at different election offices all over the country, and each resident gets a ballot paper saying in which office he is supposed to give his vote. The set of all residents can thus be divided into groups all voting in the same office. To do this in a convenient way it is assumed that each person votes at the office nearest to his place of residence. If the term "nearest" is interpreted as nearest by beeline, then the resulting division looks like the graph in Figure 2.1, the *euclidean Voronoi diagram*.

But what if you live near a river and there is no bridge nearby. Then all offices on the other side of the river are very "far" away from you, in the sense that the travel distance and travel time to get there are very long. That is why it is not always reasonable to consider the beeline, but perhaps the travel distance or travel time instead. The "nearest" office can be considered by, e. g., the *city metric*, where a graph of streets is given, and you are allowed to move on the streets only. Then the resulting division of the residents may look completely different, the *city Voronoi diagram*.

Of course one can think of many more variations to define a distance measure and solve the so called *post office problem* in the corresponding Voronoi diagram.

In this chapter, we first give a formal definition of the euclidean Voronoi diagram, we explain some nice properties, show how to compute the diagram in optimal time, and name some applications. Then we present some variations with other types of sites and other distance measures, and finally, to get a general construct, we introduce *abstract Voronoi diagrams*.

2.1 Euclidean Voronoi Diagrams

Given a set S of n point-sites in the plane, the *(euclidean) Voronoi region* of a site $p \in S$ is defined to be the set of all points in the plane, which are closer to p than to any other site from S, i. e.,

$$\mathrm{VR}(p, S) := \{x \in \mathbb{R} \ : \ |p - x| < |q - x| \ \forall q \in S \setminus \{p\}\},$$

where $|x - y|$ denotes the euclidean distance from x to y. The *Voronoi Diagram* of S is then defined by

$$V(S) := \mathbb{R}^2 \setminus \bigcup_{p \in S} \mathrm{VR}(p, S),$$

see Figure 2.1 for an example of a Voronoi Diagram.

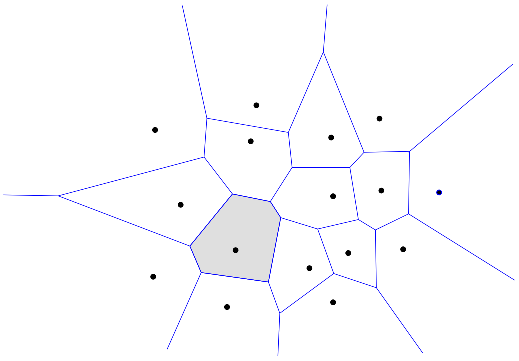


Figure 2.1: The euclidean Voronoi Diagram of a set of point-sites in the plane, the shaded area is a Voronoi region.

The diagram consists of Voronoi edges, where exactly two Voronoi regions meet, and Voronoi vertices, where three or more regions meet. By this observation, the diagram can be seen as a geometric graph with vertices and edges.

In order to get a better understanding of Voronoi diagrams it is helpful to introduce the notion of bisectors. For each pair of sites p and q in S, their bisector is defined by

$$B(p,q) := \{x \in \mathbb{R} \ : \ |p - x| = |q - x|\},$$

which is the set of points having the same distance to both p and q. This corresponds to an unbounded straight line separating the plane into two open unbounded domains $D(p,q)$ and $D(q,p)$. The first one is the set of all points which are nearer to p than to q, and the latter one is the set of all points which are nearer to q than to p, see Figure 2.2.

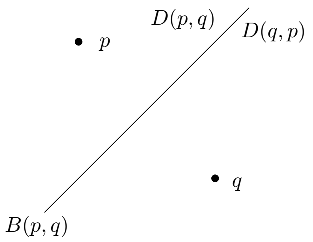


Figure 2.2: The bisector $B(p,q)$ of two point-sites p and q.

Now it is clear that the Voronoi region of a point-site p is the intersection of $n - 1$ open halfplanes. Thus, it is connected and convex. Furthermore, it contains its site p and is hence not empty.

Using the *Euler formula* and the fact that each Voronoi vertex is of degree ≥ 3 (by definition) the following lemma can be proven.

Lemma 1. *The euclidean Voronoi diagram of a set of n sites consists of $O(n)$ edges, vertices, and faces.*

The next question is how to compute the diagram. Let us first observe the following lemma, see Lemma 2.2 in [14] for a proof.

Lemma 2. *The Voronoi region of a site $p \in S$ is unbounded iff p lies on the convex hull of S.*

Now the Voronoi diagram is connected, iff the points in S do not all lie on a single straight line. Otherwise there would be a site $p \in S$ whose region disconnects $V(S)$ and thus the boundary of its region consists of at least two disjunct unbounded polygonal chains. Because all regions are convex, the boundary must consist of exactly two straight lines, and because the region is the intersection of $n-1$ open halfplanes, defined by the bisectors, all bisectors between p and any other site $q \in S$ must be parallel. This implies that all sites lie on a single straight line. On the other hand, if all points are in such position, the diagram consists of $n-1$ parallel straight lines, see Figure 2.3.

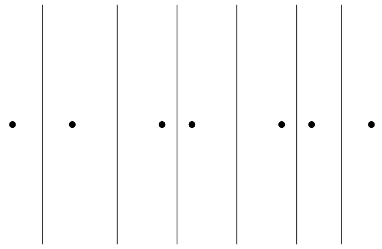


Figure 2.3: If all n sites are collinear, the corresponding Voronoi diagram consists of $n-1$ parallel straight lines.

To prevent disconnectedness one can consider a large curve Γ enclosing the "interesting" part of the diagram, i. e., Γ intersects each bisector exactly twice and encloses all bisector intersections.

To store the Voronoi diagram it makes sense to use a *doubly connected edge list (DCEL)* or even a *quad edge data structure (QEDS)*, see Guibas and Stolfi [43]. Here, one can efficiently output the adjacent Voronoi edges in sorted order around each given vertex. Similar, one can output all Voronoi edges in sorted order along the boundary of each given region.

So, if we have the large curve Γ around the diagram one can easily output all unbounded regions in sorted order around the diagram. Each unbounded region belongs to a site on the convex hull of S, see Lemma 2, thus given the diagram, we can in linear time compute the convex hull of S. This directly implies that the construction of the Voronoi diagram is in $\Omega(n \log n)$.

2.1.1 Basic Algorithms

The first worst-case optimal algorithm for computing the Voronoi diagram was presented by Shamos and Hoey [67]. Their algorithm is based on a *Divide&Conquer* approach, where the set of sites is first ordered according to their x- (and y)-coordinates, and then split into two subsets L (points to the left) and R (points to the right) of about the same size. Then the Voronoi diagrams of L and R are computed recursively. The crucial part is how to merge $V(R)$ and $V(L)$ in linear time. For this purpose a so called *merge chain* $B(L, R)$ is computed, which contains all edges of $V(S)$ separating a region belonging to a site in L and a region belonging to a site in R. This merge chain is y-monotone and all regions of sites in L are to the left of it and all regions of sites in R to the right of it. Thus, $V(S)$ can be obtained by glueing together $B(L, R)$ with the parts of $V(L)$ to the left of it and the parts of $V(R)$ to the right of it.

It remains to construct $B(L, R)$, compare Figure 2.4. Based on a method by Chew and Drysdale [26], an unbounded starting edge is found by simultaneously walking around the unbounded regions of $V(R)$ and $V(L)$ and finding two regions $\mathrm{VR}(l, L)$ belonging to $V(L)$ and $\mathrm{VR}(r, R)$ belonging to $V(R)$, whose intersection contains an unbounded piece of $B(l, r)$. Such an endpiece of $B(l, r)$ must belong to the merge chain $B(L, R)$. Once we know where to start, we can trace $B(L, R)$ through the diagrams by testing if $B(l, r)$ hits the boundary of $\mathrm{VR}(l, L)$ or $\mathrm{VR}(r, R)$ first. Here, the edges of the region of l are scanned counterclockwise and the edges of the region of r clockwise. Because $B(L, R)$ is y-monotone, once an edge on the boundary of a region $\mathrm{VR}(l, L)$ or $\mathrm{VR}(r, R)$ has been tested unsuccessfully, it never has to be considered again. This guarantees, that $B(L, R)$ can be constructed in time $O(n)$. For more details see, e. g., [14].

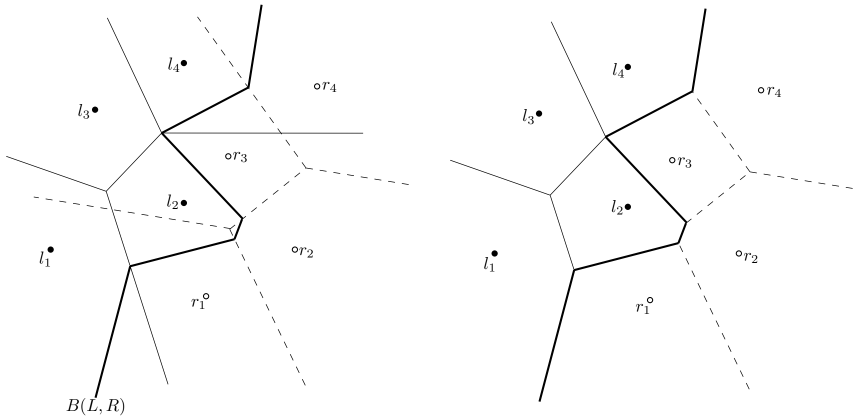


Figure 2.4: The merge chain $B(L, R)$, where $L = \{l_1, \ldots, l_4\}$ and $R = \{r_1, \ldots, r_4\}$, is traced through the overlapping of $V(L)$ and $V(R)$, and $V(L \cup R)$ is constructed.

Theorem 1. *The Voronoi diagram of n sites can be computed in optimal time $O(n \log n)$ by* Divide&Conquer*.*

Another approach to construct the Voronoi diagram is to use the *sweep-line-method*. The first idea was to sort the sites according to their x-coordinates, sweep the plane from left to right, and compute the Voronoi diagram to the left of the sweep line L, by incrementally inserting the regions of the sites, when hit by the sweep line. The problem with this idea is, that it is not clear at all how to perform the insertion process. The new region may reach far to the left and cause many new Voronoi vertices to the left of the sweep line, which seems to make it impossible to spend only $O(\log n)$ time on the insertion.

Here, Fortune [40] found the clue. He did not compute the Voronoi diagram of only the sites to the left of the sweep line, but the diagram of these sites together with the sweep line L itself. By doing this, one makes sure, that the new Voronoi region of a site hit by the sweep line always is contained in the current region of L and no Voronoi edge or vertex ever has to be changed once it has been constructed.

The sweep line is now followed by a "wavefront" while moving to the right, see Figure 2.5.

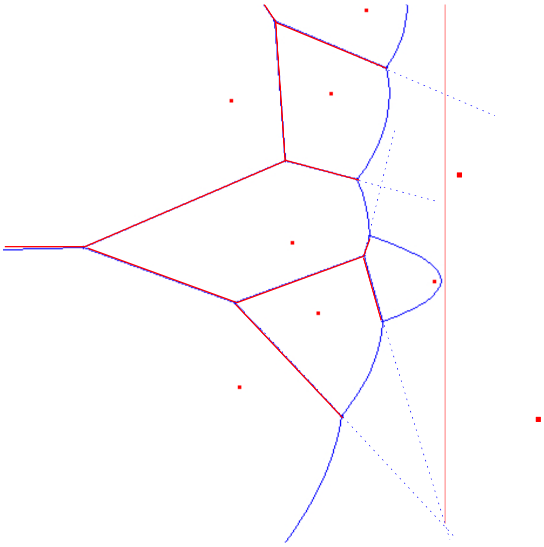


Figure 2.5: The sweep line is followed by a wavefront. The waves are separated by spikes which may intersect to the right of the sweep line.

The wavefront consists of waves separating the regions of a site and the region of L, and the Voronoi edges incident to the waves can be prolongated to the right by spikes. We need to store the ordering of the waves and the spikes along the wavefront, and we have an event order for the following events.

1. The sweep line hits a site s.
   Here the wave of s together with its two adjacent spikes has to be inserted into the wavefront. The intersection point of the upper and lower spikes, which are no longer adjacent, has to be deleted from the event queue (if it exists) and the two new spikes have to be tested for intersection with their neighbors, and if the intersection points are to the right of the wavefront, they have to be inserted into the event queue.

2. The wavefront hits the intersection of two adjacent spikes.
   The wave disappearing at this point together with its two adjacent spikes has to be deleted from the wavefront and the upper and lower spikes now

adjacent have to be tested for intersection. If the intersection point lies to the right of the wavefront, it has to be inserted into the event queue.

If each wave has the name of the site whose region it bounds, then the ordering of the waves along the wavefront is a Davenport-Schinzel sequence of order 2.

Davenport-Schinzel-Sequences are defined as follows, see, e. g., [68].

Definition 1. *An (n, s)-*Davenport-Schinzel sequence*, (n, s)-DSS for short, is a sequence of integers $\langle u_1, \ldots, u_k\rangle$, where $1 \leq u_i \leq n$ for all $1 \leq i \leq k$, no two adjacent integers are equal and there exists no (possibly uncontinuous) subsequence $\langle u_i \ldots u_j \ldots u_i \ldots\rangle$, where u_i and u_j alternate more than s times.*

We refer to s as the *order* of the DSS and n the *number of symbols.*

Definition 2. *The maximum length of an (n, s)-DSS is denoted by $\lambda_s(n)$.*

We have the following lemma, see again [68].

Lemma 3. *For any (n, s)-DSS we have*

(i) $\lambda_1(n) = n$.

(ii) $\lambda_2(n) = 2n - 1$.

With this lemma it is clear that the length of the wavefront is $O(n)$, thus using a data structure, like, e. g., a balanced binary tree with linked leaves, the operations of inserting and deleting waves of the wavefront, and inserting and deleting spike events of the event queue, can be performed in $O(\log n)$ time each. Because the Voronoi diagram has $O(n)$ vertices there are as many spike events and of course there are $O(n)$ site events.

Theorem 2. *The Voronoi diagram of n sites can be computed in optimal time $O(n \log n)$ using the sweep line method.*

2.1.2 Applications

Let us get back to the topic, why Voronoi diagrams are so useful. The following applications are all mentioned by Shamos and Hoey [67].

One of the most known applications is that Voronoi diagrams can help us to solve the *All-Nearest-Neighbor-Problem.* Let S be the set of points given in the plane and $s \in S$. If we remove s from S, then its region is divided into several pieces, which are given to some other regions. Because all Voronoi regions are connected, only those regions having a common border with the region of s can gain part of its region. If we now reinsert s, then s lies in the region of its nearest neighbor and this must be one of the sites having a common border with it. Thus the nearest neighbor of a site always lies in an adjacent region. Because the diagram has $O(n)$ Voronoi edges, we have to compare only $O(n)$ pairs of sites to solve the *All-Nearest-Neighbor-Problem.*

Also very well known is the *Post-Office-Problem*, named in the introduction of this chapter. Here n sites (the post-offices) together with a query point q are given and we want to know which of the *post-offices* is the nearest one,

i. e., we need to know in which Voronoi region q is located. Given the Voronoi diagram of the n sites, we can build a data structure mentioned in Dobkin and Lipton [36], called the *slab method*. It consists of the Voronoi diagram together with $O(n)$ horizontal straight lines, one through each Voronoi vertex, and the Voronoi regions in each strip are ordered, compare Figure 2.6. Given a query point q, we can first search for its vertical location between the horizontal lines, and then for its horizontal location between the regions. This implies that the *Post-Office-Problem* can be solved in time $O(\log n)$. Unfortunately we may need $\Omega(n^2)$ time and space to build the data structure, because the horizontal lines altogether may intersect the Voronoi diagram in $\Omega(n^2)$ many points. This can though be improved to linear time and space by using the ideas from Edelsbrunner [37] and Seidel [66] instead.

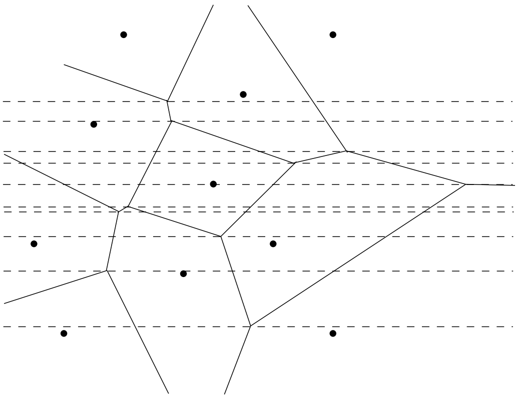


Figure 2.6: A data structure for point location in a Voronoi diagram.

Now, suppose someone wants to build a permanent repository for nuclear waste within a given area, as far away as possible from all places of residence. If the places of residence can be represented by a set of n point shaped sites S, then the repository should be build at the center point of a largest circle within the area, not containing any sites from S. Such a *largest empty circle* must have its center point on a Voronoi vertex of the Voronoi diagram of S; for each Voronoi vertex v there exists a circle having its center point at v, three sites of S on its boundary, and no site in its interior. This implies that the problem of finding a largest empty circle can be solved in time $O(n)$ once the Voronoi diagram has been constructed.

The dual of the Voronoi diagram is a straight line graph and is defined as the *Delaunay tesselation.* If the sites are in general position, i. e., no four sites lie on a circle and no three points are collinear, then all Voronoi vertices are of degree 3, and the dual graph is a triangulation, the *Delaunay triangulation*, $D(S)$ for short. It is famous for its nice features. For each triangle in $D(S)$ its circumcircle does not contain any other site, this is because otherwise, the three vertices of the triangle would not define a vertex in the Voronoi diagram and thus the triangle would not be part of the dual graph. This observation also implies that the Delaunay triangulation is a triangulation which maximizes the smallest angles in the triangles. Let $(\alpha_1, \ldots, \alpha_k)$ be a non decreasing sequence of the interior angles of all triangles of $D(S)$. Then for all triangulations of the same set of sites with non decreasing angle sequence $(\beta_1, \ldots, \beta_k)$, let $1 \leq i \leq k$ be the smallest index for which $\alpha_i \neq \beta_i$. Then $\alpha_i > \beta_i$. Because of this property, Delaunay triangulations are often used for mesh generation applications in terrain modeling, see, e. g., Lawson [55].

The next observation is that the edges of a *minimum spanning tree, MST,* of a given set of sites is a subgraph of the Delaunay triangulation. Because the triangulation is of linear size, to compute the MST it is enough to consider $O(n)$ edges instead of all $\binom{n}{2}$ edges. Using Kruskal's algorithm the MST can be computed in time $O(n \log n)$ given $D(S)$, using Prim's algorithm improves it to $O(n \log \log n)$. Cheriton and Tarjan [25] showed that this bound even can be decreased to $O(n)$, see also Preparata and Shamos [64].

2.2 Variations

As already mentioned before, the euclidean Voronoi diagram is not always the most reasonable one when meeting practical needs. There may be instances requiring other types of sites and other distance measures. Motivated by this many variations of Voronoi diagrams have been introduced and we will discuss some of them in this section.

2.2.1 General L_p-Metrics

Suppose you live in Manhattan. The city consists of only horizontal and vertical streets. If you want to know which distance you have to travel to get from one point x in the city to another one y, you have to compute the distance along the grid pattern. This corresponds to the L_1-metric, also called Manhattan-metric, and is defined by

$$|x - y|_1 := |x_1 - y_1| + |x_2 - y_2|,$$

where $x = (x_1, x_2)$ and $y = (y_1, y_2)$ in cartesian coordinates.

Now the bisector of two point shaped sites p and q looks like depicted in Figure 2.7.

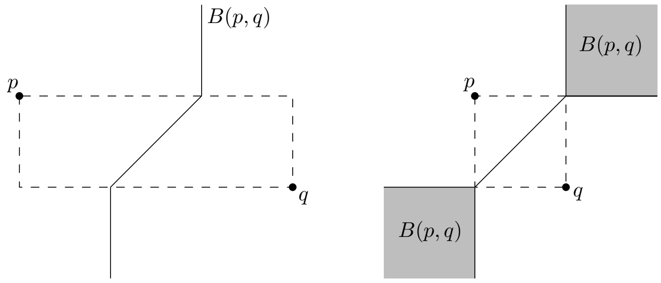


Figure 2.7: The shape of the L_1-bisector.

If the two sites are in general position, then their corresponding bisector consists of two unbounded either vertical or horizontal line segments connected by the diagonal of a square. If p and q are diagonal vertices of an axis aligned square, then the bisector consists of the diagonal separating the sites, and two unbounded squares. This causes a problem, because normally only one-dimensional bisectors are allowed. To avoid nasty algorithms usually points in general position are assumed, thus no two sites may be diagonal vertices of an axis aligned square.

In Figure 2.8 an example of the Voronoi diagram of point sites in the L_1-metric, and is corresponding diagram under the L_2-metric is depicted. Interestingly, the graph structures are different; under the L_1-metric all regions are unbounded, but under L_2 one region is bounded.

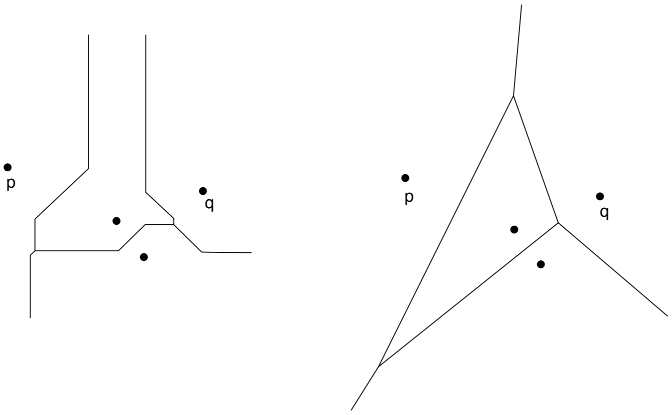


Figure 2.8: The Voronoi diagram of point sites in the L_1-metric (left), and its corresponding L_2-diagram (right), (from [14]).

Another special case is the L_∞-metric, defined by

$$|x - y|_\infty = \max\{|x_1 - y_1|, |x_2 - y_2|\}.$$

Here the bisectors look like depicted in Figure 2.9. In general, the bisector consists of two unbounded diagonals connected by a horizontal or vertical line segment. The problem is when two sites lie on a vertical or horizontal line. Then again we have a 2-dimensional bisector and a general position assumption is necessary to avoid complications.

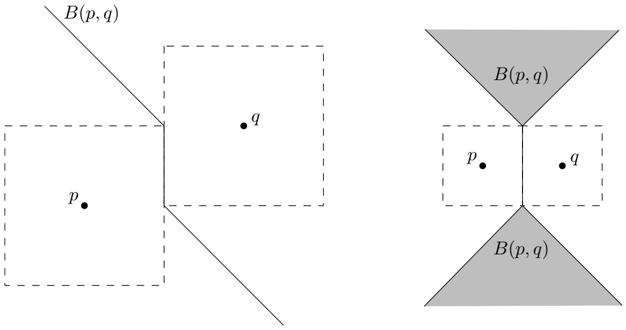


Figure 2.9: The shape of the L_∞-bisector.

Of course one can also define the Voronoi diagram according to general L_p-

metrics

$$|x-y|_p := \sqrt[p]{(x_1-y_1)^p+(x_2-y_2)^p}.$$

Here all bisectors are one-dimensional and thus easier to handle.

2.2.2 Weighted Voronoi diagrams

All the Voronoi diagrams discussed so far can be represented as follows. Suppose some kings are positioned at the point-shaped sites and they send out their troops at the same time with the same speed in all directions, i. e., in circular formations. Where different troops meet, they have to stop (a Voronoi edge emerges), where three or more different troops meet we get a Voronoi vertex. Now each king has conquered the area, where his troops reached first, his Voronoi region. In the L_1-metric, the "unit circle" is a unit square with sides oriented at a 45° angle to the coordinate axes. Here, the troops of the kings would spread out in this square formation. For the L_∞-metric the "unit circle" is an axes aligned unit square.

But what happens if the troops start moving at different times, e. g., if one king declares war and it takes some time before the others realize it. Or if they move with different speeds, because they are equipped with different vehicles. Such resulting plane divisions are called *weighted Voronoi diagrams*. In the first case they are *additively weighted*; formally the distance from a point $x \in \mathbb{R}^2$ to a site p is

$$d_p(x) := |x-p| - w_p,$$

where $w_p \geq 0$ is the weight of p. It can be interpreted as the advance of the expansion of the "circle" around p. Thus a site with high weight gains a greater Voronoi region than a site with low weight.

The bisectors $B(p,q)$ now have the shape of hyperbolas and are unbounded, unless $|p-q| \leq |w_p - w_q|$. Then, assuming $w_p < w_q$, the region of p would be empty. Also it is easy to show that all nonempty Voronoi regions are star shaped seen from their defining site and thus connected. An example of an additively weighted diagram, also called *power diagram*, is depicted in Figure 2.10; here the weights correspond to the radii of the circles.

For *multiplicatively weighted* Voronoi diagrams the circles all start expanding at the same time, but with different speeds. The distance from x to site p is then defined by

$$d_p(x) := \frac{1}{w_p}|x-p|.$$

Again a site with high weight gains a greater region than a site with low weight. Interestingly, if $w_p \neq w_q$, then the bisector $B(p,q)$ is a circle enclosing the site with lower weight (but with center point different from the site). So, different from the Voronoi diagrams discussed so far, in this case we have to deal with closed bisecting curves. Also the Voronoi diagram may now contain "islands", Voronoi regions completely enclosed by just one other region and thus being bounded by an edge without vertex, see Figure 2.11.

Another new phenomenon is that Voronoi regions need no longer be connected. The whole Voronoi diagram may even consist of $\Theta(n^2)$ many faces, but each Voronoi region consists of $O(n)$, see Figure 2.12. Half of the sites $p_1, \ldots, p_{\frac{n}{2}}$ are positioned on a horizontal line, and have low weights. The other

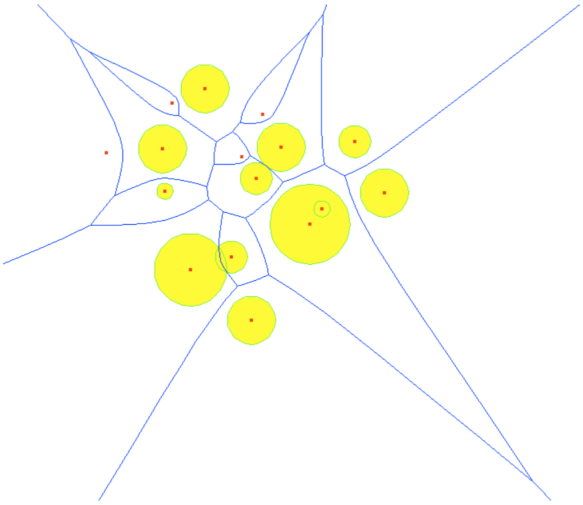


Figure 2.10: An additively weighted Voronoi diagram.

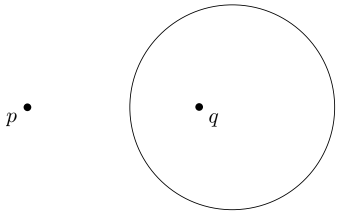


Figure 2.11: Two sites p and q with multiplicative weights.

half $q_1, \ldots, q_{\frac{n}{2}}$ are positioned on a vertical line, and have high weights. Then the region of each $q_i, i \neq 1, \frac{n}{2}$, is split into $\Theta(n)$ faces, and the size of the whole diagram is $\Theta(n^2)$.

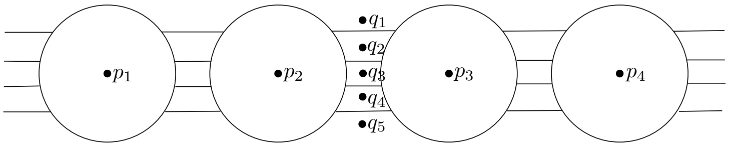


Figure 2.12: The size of a multiplicatively weighted Voronoi diagram may be $\Theta(n^2)$.

Aurenhammer and Edelsbrunner proved the properties from above in [12] and made use of them to present an algorithm computing the diagram in optimal worst case time $\Theta(n^2)$. For each bisector they defined a sphere, which they mapped to a halfplane by inversion at a point I. Now each Voronoi region corresponds to the intersection of halfspaces, a polyhedron, which is connected and hence much simpler to compute. By transforming the polyhedrons back to the plane one obtains the Voronoi diagram.

2.2.3 Voronoi Diagram of Line Segments

In areas like biology, geography, pattern recognition, computer graphics, and motion planning, applications making use of Voronoi diagrams of line segments can be found, see, e. g., the book by Siddiqi and Pizer [69].

Let a set S of n line segments be given, and assume that they are pairwise disjoint except for their endpoints, i. e., they may touch only at their endpoints. Such a set of line segments defines a planar straight line graph.

Now the Voronoi region of a line segment $l \in S$ is the set of all points $x \in \mathbb{R}^2$ which are nearer to l than to any other line segment from S. The distance from a point x to a line segment l is defined as the distance from x to a nearest point on l.

Let us take a look at the bisector $B(l, s)$ for two line segments $l, s \in S$. If l and s are both single points, then the bisector equals the euclidean one, defined in Section 2.1. If l is a line segment with endpoints a and b, where $a \neq b$, but s is still a single point, then the bisector $B(l, s)$ consists of three components, a segment of the bisector between a and s, a segment of the bisector between b and s and an arc of the parabola bisecting l and s in between, see Figure 2.13a for an example. If both l and s are line segments with endpoints a and b, and c and d respectively, where $a \neq b$ and $c \neq d$, then the bisector consists of four components, segments of the bisectors $B(c, a)$ and $B(d, b)$ in the beginning and end, and in between first an arc of the parabola bisecting c and l (or a and s), then a segment of the angular bisector between l and s, followed by an arc of the parabola bisecting d and l (or b and s), see Figure 2.13b.

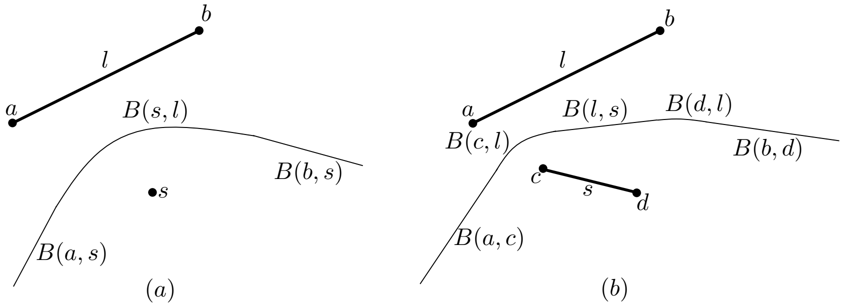


Figure 2.13: The bisector of line segments.

The Voronoi diagram of line segments is a geometric graph, whose edges are arcs of straight lines and parabolas. It is easy to see that the whole region of a line segment l is visible from it, thus all regions are simply connected, i. e., they are connected and they have no holes. Using the *Euler formula* like in Section 2.1, we obtain the following lemma.

Lemma 4. *The Voronoi diagram of a set of n line segments is of size $O(n)$, i. e., it consists of $O(n)$ edges, vertices, and faces.*

There are several algorithms computing the Voronoi diagram in optimal time. Kirkpatrick [46], Lee [56], and Yap [73] developed *divide&conquer algorithms* running in time $O(n \log n)$ and space $O(n)$. Fortune [40] showed how to compute the Voronoi diagram by *plane sweep method*, and Boissonat et. al.

[21] and Klein et. al. [53] by *randomized incremental construction*, all in time $O(n \log n)$.

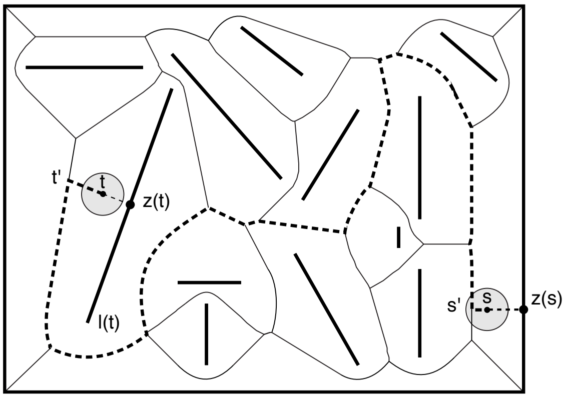


Figure 2.14: The Voronoi diagram of line segments and a possible path for a disk to be moved from s to t, (from [47]).

A nice application of these Voronoi diagrams can be found in the area of motion planning, see [14]. Suppose you have a disk-shaped robot and you want it to move from a starting point s to a termination point t. The problem is that there are obstacles having the shape of line segments in its way. The question is, if there is a possible path from s to t, such that the robot never touches an obstacle. Because each point on the Voronoi diagram of the obstacles maximizes its minimum distance to any line segment, it is enough to find out if there is a collision free path on the diagram, see Figure 2.14. Each edge of the diagram is an arc of a bisector defined by two line segments and we can in constant time compute the minimum distance from a point on the edge to its nearest line segment. If this distance is less than the radius r of the robot, the edge can be removed from the diagram, because the robot would not be able to move along it. After testing all edges, a graph remains, where all edges can be traversed by the robot. We can now by breadth-first-search test whether t is still reachable from s and if yes, we will find a path connecting s and t.

Lemma 5. *Once the Voronoi diagram of n noncrossing line segments is constructed, we can for a disk-shaped robot of radius r, in time $O(n)$ compute a collision free path from a starting point s to a termination point t, or decide that no such path exists.*

2.2.4 Voronoi Diagram of Polygons

An interesting special case of the Voronoi diagram of line segments is the diagram of a convex polygon, i. e., the diagram of line segments forming a convex polygon. Because all line segments are in convex position, each Voronoi region is unbounded, analogously to point-sites in convex position, see Section 2.1. This implies that the Voronoi diagram has a tree structure and it is easy to see that it is even a tree with straight edges. It equals the *medial axis* in the interior of the polygon, see Figure 2.15.

Such a diagram can be computed in time $O(n)$, see Aggarwal et. al. [5]. The same technique also works for the euclidean Voronoi diagram of points in convex

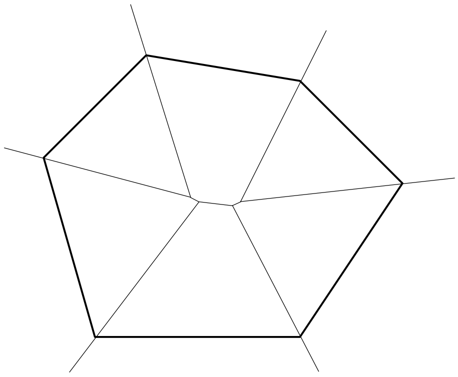


Figure 2.15: The Voronoi diagram of line segments forming a convex polygon.

position, which also has a tree structure, because all Voronoi regions of points on the convex hull are unbounded, compare Section 2.1.

The details of this algorithm though are quite involved. They have been both simplified and generalized by Klein and Lingas [50] to an algorithm working for *Hamiltonian abstract Voronoi diagrams*[1], where a curve is given intersecting each Voronoi region exactly once. If the ordering of the regions along the curve is given, the abstract Voronoi diagram can be computed in linear time. This applies to diagrams having the shape of a tree and where the ordering of the unbounded regions around the diagram is known. More about this will be discussed in Section 6.

Next, let us shortly consider the Voronoi diagram of k polygonal sites and let the sum of all edges over the k polygons be $O(n)$. Then the Voronoi diagram can be constructed in time $O(n \log n)$ by first computing the line segment diagram of the polygon edges, and then joining those regions belonging to a line segment of the same polygon.

2.2.5 More Voronoi Diagrams

Of course there are many more variations of Voronoi diagrams, see also [10, 22, 41, 62], and it is impossible to name them all. We will just mention some more, which can be used as nice examples later on.

First, let us introduce convex distance functions, compare [14]. Let C be a compact convex set in the plane containing the origin in its interior. To measure the distance from a point p to a point q, translate the set C by the vector p. Then the halfline from p to q intersects C at a unique point q'. Now we define the distance from p to q with respect to the set C as

$$d_C(p, q) = \frac{d(p, q)}{d(p, q')},$$

compare Figure 2.16.

It can be shown that the triangle inequality $d_C(p, q) \leq d_C(p, r) + d_C(r, q)$ is fulfilled, with equality when the three points p, q, r are collinear. The symmetry

[1] Abstract Voronoi diagrams will be introduced in Section 2.3.

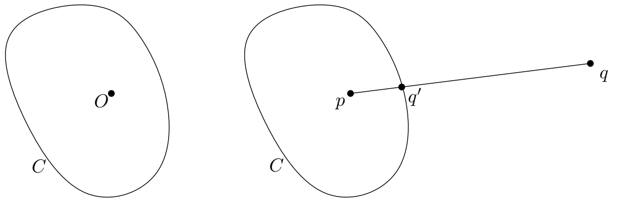


Figure 2.16: The definition of the convex distance from p to q with respect to a convex set C.

is holding only when C is point symmetric to the origin. In this case d_C is a metric. Further, the convex set C corresponds to the unit-circle, because the distance from p to any point on C is 1. For L_p-metrics the unit circles are all symmetric about their centerpoint, and they all define a convex distance function, which is even a metric.

As already seen in Section 2.2.1, the bisector of points in the L_1- and L_∞-metric can be two-dimensional. This is because the unit-circle is not strictly convex, i. e., it contains line segments on its boundary. If C ist strictly convex, then the bisectors are all one-dimensional, unbounded, and even homeomorphic to a line. The bisector of two points p and q can be constructed by drawing lines parallel to the upper and lower tangent of the two copies C_p and C_q of C, translated by the vectors p and q. Where these lines intersect C_p and C_q in 4 points, two of them, p' on C_p and q' on C_q, are facing each other. The two lines through p and p', and q and q' intersect in a point x which, because of the intercept theorem, lies on the bisector $B(p,q)$, compare Figure 2.17.

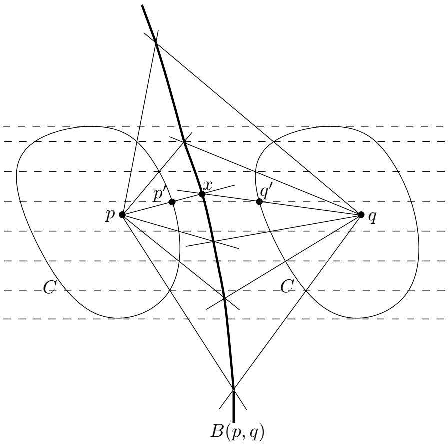


Figure 2.17: The bisector $B(p,q)$ with respect to a convex distance function d_C.

Next, let us consider the *Karlsruhe-metric*, sometimes also called *Moscow-metric*. Here, an origin O is given, and the shortest distance from a point p to a point q is measured as the shortest path from p to q running along radii from O and circles around O, see Figure 2.18.

If points are not in general position, e. g., they are positioned on the same circle around the origin, then the following problem depicted in Figure 2.19

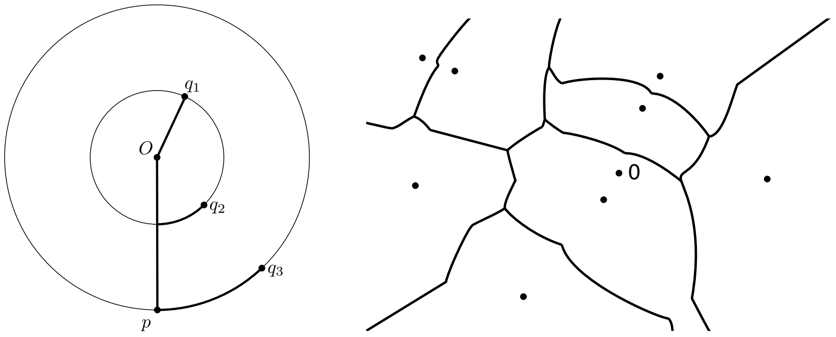


Figure 2.18: The Karlsruhe metric and its Voronoi diagram, (from [14]).

can occur. In the left drawing the bisector of the two points p and q is two-dimensional. This is because all shortest paths from p or q to a point in the shaded area runs through the origin. Simple geometry show that a shortest path from a point p to a point x runs through the origin, if an only if the angle between $|Op|$ and $|Ox|$ exceeds $114,59°$.

In the right drawing of Figure 2.19 the bisectors are one-dimensional, but the region of the site p has a cut-point at the origin and is thus not connected seen as an open set. Again it spares you a lot of case analysis to assume general position.

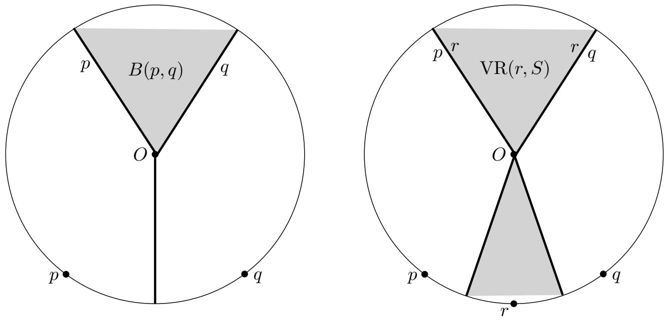


Figure 2.19: The bisector and Voronoi region according to the Karlsruhe-metric of points not in general position.

Another example is the *anisotropic Voronoi diagram*, a generalization of multiplicatively weighted Voronoi diagrams, see Labelle and Shewchuk [54]. In the "expanding circle view" of Voronoi diagrams, here we do not have expanding circles, instead we have an expanding ellipse from each site. The bisectors are quadratic curves and the size of the diagram is $\Omega(n^2)$ and $O(n^{2+\epsilon})$. It can be constructed in time $O(n^{2+\epsilon})$ by an algorithm of Agarwal, Schwarzkopf, and Sharir [4, 68] for minimization diagrams. Anisotropic Voronoi diagrams are suitable, e. g., in mesh generation, especially for domains where long and skinny triangles are required.

2.2.6 Farthest Voronoi Diagrams

Up to now we have discussed many Voronoi diagrams, dividing the plane into regions, such that all points from one region have got the same *nearest* site. But what if we instead want to consolidate in regions all points having the same *farthest* site. This defines the *farthest Voronoi diagram* $V^{-1}(S)$. Formally, let $d_p(x)$ be the distance from a point $x \in \mathbb{R}^2$ to a site $p \in S$, then the *farthest Voronoi region* of p is

$$\mathrm{VR}^{-1}(p, S) := \{x \in \mathbb{R}^2 |\ d_p(x) > d_q(x) \forall q \in S \setminus \{p\}\}.$$

Such a diagram would, e. g., solve the problem of finding a smallest disk intersecting all sites. Such a disk has its centerpoint on either a vertex or an edge of the *farthest Voronoi diagram.*

In the euclidean case, the farthest Voronoi diagram $V^{-1}(S)$ is always a tree. This is because only the sites lying on the convex hull of S have got a nonempty region, and these regions are all unbounded and connected, see Figure 2.20 for an example.

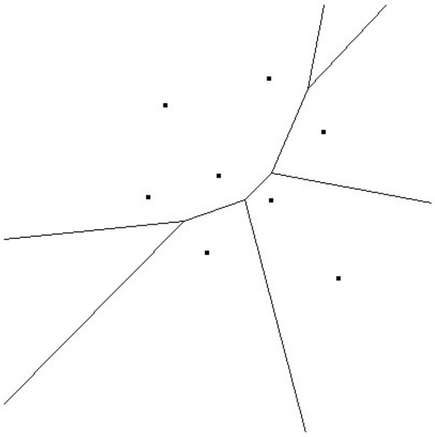


Figure 2.20: The euclidean farthest Voronoi diagram of a set of points.

The bisectors of the farthest diagram are the same as the bisectors from the nearest diagram by just switching their labels, i. e., if in the nearest diagram the dominance region of p lies to the left of the curve and the dominance region of q to the right, then it is the other way around in the farthest Voronoi diagram.

There are several methods to construct the diagram in $O(n \log n)$ time, see [14]. One is by lifting the diagram into three-dimensional space, where the bisectors are represented by hyperplanes. The lower envelope of these hyperplanes then corresponds to the farthest Voronoi diagram. Another possibility is to construct the farthest Delaunay triangulation, the dual of the Voronoi diagram, which in the farthest case is a convex polygon. This can be done with a simple *ear-clipping algorithm.*

Interestingly, the farthest Voronoi diagram of line segments has also got a tree structure, but the Voronoi regions need no longer be connected, as depicted in Figure 2.21. Nevertheless, the size of the diagram is linear in n and it can be constructed in $O(n \log n)$ time, as shown by Aurenhammer, Drysdale, and Krasser [11].

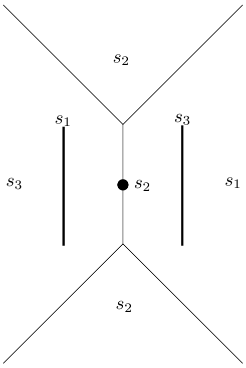


Figure 2.21: The farthest Voronoi diagram of line segments. The region of s_2 consists of two disjoint faces.

2.2.7 Higher Order Voronoi Diagrams

We have already discussed the notion of nearest and farthest Voronoi diagrams. The extension of these are called *higher order Voronoi diagrams*. The first order diagram $V^1(S)$ is the nearest one. In the order-k diagram, for $1 \leq k \leq n-1$, points belonging to the same Voronoi region are having the same k-nearest sites. Obviously, the order $n-1$ diagram is the farthest Voronoi diagram.

Let H be a subset of S of size k. The Voronoi region of H, i. e., the set of points having the sites in H as their k nearest neighbors, is

$$\begin{aligned}\mathrm{VR}^k(H,S) &= \{x \in \mathbb{R}^2 | d(x,h) < d(x,p), \forall h \in H \land p \in S \setminus H\} \\ &= \bigcap_{h \in H, p \in S \setminus H} D(h,p),\end{aligned}$$

where $D(h,p)$ is the *dominance region* of h with respect to p, i. e., the set of points which are nearer to h than to p.

A Voronoi region of order-k may be empty and it need not contain its defining sites, but because it is the intersection of halfplanes it is always connected, see Figure 2.22 for an example.

There are $\binom{n}{k}$ subsets of S of size k, and the question is, how many of these do have a nonempty region in the diagram of order-k. D. T. Lee [57] showed that $V^k(S)$ has $O(k(n-k))$ nonempty regions and is thus of the same size. He also developed an algorithm to construct the order-k diagram in time $O(k^2 n \log n)$ and space $O(k^2(n-k))$, which is efficient for small k but not for large ones. Also, this algorithm actually computes all orders from 1 to k. It is still an open problem to compute only the order-k diagram, in output sensitive time.

For line segments, as already discussed in the previous section for the case of the farthest diagram, the higher order Voronoi regions need not be connected, see also Figure 2.23. Still, Papadopoulou and Zavershynskyi [63] could show,

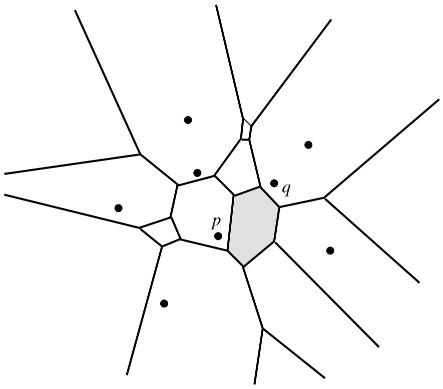


Figure 2.22: The euclidean order-2 diagram of point-sites, (from [14]).

by extensively using the geometry of the sites, that the order-k diagram of line segments has $O(k(n-k))$ faces.

Very recently it has been shown, that the order-k Voronoi diagram of line segments can be constructed in time $O(kn^{1+\epsilon})$, by a general algorithm computing the order-k abstract higher order Voronoi diagram, [20], which will be introduced in Chapter 5.

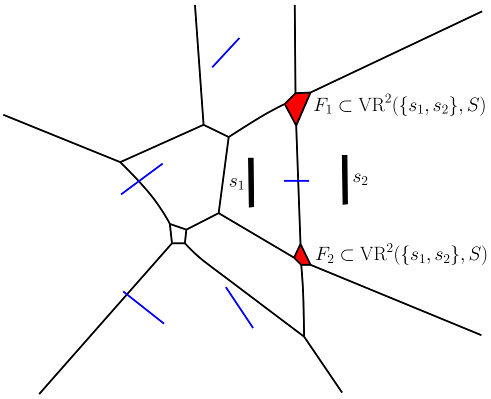

$F_1 \subset \mathrm{VR}^2(\{s_1, s_2\}, S)$
s_1
s_2
$F_2 \subset \mathrm{VR}^2(\{s_1, s_2\}, S)$

Figure 2.23: The order-2 Voronoi diagram of line segments, (from [63]).

2.3 Abstract Voronoi Diagrams

As we have seen, there are extremely many different kinds of Voronoi diagrams, based on different types of sites and distance measures. Every time a new Voronoi diagram was defined, one had to go through its properties and develop a new algorithm to compute it. This is why the temptation was high to come up with a unifying concept covering as many Voronoi diagrams as possible. That was when Klein [48, 14] introduced *abstract Voronoi diagrams*, AVD's for short.

He realized that most concrete Voronoi diagrams have one feature in common. For each pair of sites p and q, there is a set of points having the same distance to both sites. This set is often a one-dimensional curve separating the plane into two domains, one consists of all points that are nearer to p than to q, and the other one is the set of points that are nearer to q than to p, see the previous sections for examples.

The discovery was that the dominance regions are actually enough to know how to construct the Voronoi diagram. Thus the idea was to assume that a set S of n non-physical abstract sites, indexed by $\{p, q, r, \ldots\}$ are given, and for each pair of sites p and q there is a bisecting curve $J(p, q)$ separating the plane into the two dominance regions, $D(p, q)$ on one side of $J(p, q)$, and $D(q, p)$ on the other side.

Now the abstract Voronoi region of a site p is defined by

$$\mathrm{VR}(p, S) := \bigcap_{q \in S \setminus \{p\}} D(p, q),$$

and the abstract Voronoi diagram by

$$V(S) := \mathbb{R}^2 \setminus \bigcup_{p \in S} \mathrm{VR}(p, S).$$

It is clear that this definition coincides with the definitions used to define concrete Voronoi diagrams, where $D(p, q)$ was the set of points that are nearer to p than to q. Now, the first question arising is which assumptions are needed to make the set of bisectors $\mathcal{J} = \{J(p, q) : p, q \in S\}$ admissible.

Definition 3. *A set of bisecting curves $\mathcal{J} = \{J(p, q) : p, q \in S\}$ is called admissible if for each subset $S' \subseteq S$ of size at least 3 the following axioms are fulfilled.*

(A1) By stereographic projection to the sphere, each curve $J(p, q)$, $p, q \in S$, can be completed to a closed Jordan curve through the northpole.

(A2) Each Voronoi region $V(p, S')$ is path-connected.

(A3) Each point of the plane belongs to the closure of a Voronoi region $\overline{VR(p, S')}$.

Each curve $J(p, q)$ fulfilling axiom (A1) separates the plane into two unbounded open domains $D(p, q)$ and $D(q, p)$. By the Jordan-Schönflies theorem, each point x on $J(p, q)$ can be accessed from $D(p, q)$ (and $D(q, p)$) by an arc that runs entirely in $D(p, q)$ (or $D(q, p)$) except for its endpoint x.

Because of the Jordan curve property and axiom (A3) the following lemma can be shown, see, e. g., [49, Lemma 3].

Lemma 6. *Let $\mathcal{J}$ be an admissible curve system for a set of sites S. Then,*

$$V(S) = \bigcup_{p \neq q \in S} \overline{VR(p, S)} \cap J(p, q) = \bigcup_{p \neq q \in S} \overline{VR(p, S)} \cap \overline{VR(q, S)}.$$

A *Voronoi edge* e of $V(S)$ can now be defined as a maximal connected subset of $V(S)$ such that all points x on e lie on the boundary of exactly two different Voronoi regions of sites p and q. Edge e separates the Voronoi regions of p and q, and is an arc of the bisector $J(p, q)$. A *Voronoi vertex* v is a point x of $V(S)$ which lies on the boundary of the regions of at least three sites p, q, and r of S.

For physical sites p, q, r and a distance measure, we have a transitivity property, saying that if a point x is nearer to p than to q, and nearer to q than to r, then x is nearer to p than to r. Surprisingly, a similar fact can be shown for abstract Voronoi diagrams, see [49, Lemma 5].

Lemma 7. *Let $\mathcal{J}$ be an admissible curve system and $p, q, r \in S$. Then*

$$D(p, q) \cap D(q, r) \subseteq D(p, r).$$

Observe that to prove this lemma axioms (A1) and (A3) are enough. The transitivity turned out to be a very strong and useful property.

For example it helps us to show that the abstract Voronoi diagram has a graph structure. At first this was proven using an additional assumption stating that two arbitrary bisectors have only a finite number of intersections, [48]. This property seemed to be fulfilled in most applications, but some years later Corbalan et. al. [31] came up with an example of a smoth convex distance function whose bisectors could intersect in an infinite number of components. Motivated by this, Klein, Langetepe, and Nilforoushan [49] showed that abstract Voronoi diagrams can be handled the same way even without the finite-intersection-property. They showed that the crucial *piece-of-pie Lemma* is still valid, compare [49, Lemma 11] and Figure 2.24.

Lemma 8 (piece-of-pie)**.** *Let S' be a subset of S. We have the following.*

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

Let v be a point of $V(S')$, and $U(v)$ an arbitrarily small neighborhood of v, whose boundary is a simple closed curve, such that the following holds.

(ii) Either v is interior point of some Voronoi edge $e \subseteq J(p, q)$ of $V(S')$. Then $U(v)$ is divided by e into exactly two domains, one contained in $VR(p, S')$, the other one in $VR(q, S')$.

(iii) Or v is a Voronoi vertex of $V(S')$ of degree $k \geq 3$. After suitably renumbering S', the Voronoi edges e_i adjacent to v belong to $J(p_i, p_{i+1})$ in clockwise order, where $1 \leq i \leq k$ and $e_{k+1} = e_1$. The edges e_i and e_{i+1} together with ∂U bound a piece of pie contained in $VR(p_i, S')$. These pieces are domains with Jordan curve boundaries and the sites $p_1, \ldots, p_k$ are pairwise different.

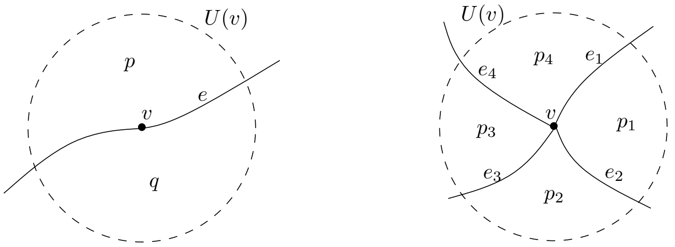


Figure 2.24: In the left drawing v is a point of a Voronoi edge separating the Voronoi regions of p and q. In the right drawing v is a Voronoi vertex of degree 4 lying on the boundary of the regions of $p_1, \ldots, p_4$.

Because all bisectors are unbounded, all Voronoi regions (and also the closures of the Voronoi regions) must be simply connected. Otherwise there would be a site q whose region is completely enclosed by the region of a site p, but then the bisector $J(p,q)$ would also have to be enclosed by the region of p and thus it would be bounded (or not simple), a contradiction to axiom (A1).

Furthermore, each Voronoi region is connected and there are n sites in S, thus by the *Euler formula* we obtain the following lemma.

Lemma 9. *The graph $V(S)$ is of size $O(n)$.*

For many purposes it is helpful to consider only the bounded part of $V(S)$, which can be achieved by adding a large curve Γ around the diagram and cutting off all bisectors outside of it. The curve Γ should intersect each bisector exactly twice and outside of Γ each pair of bisectors is either equal or disjunct. The existence of such a curve can be proven similarly to Lemma 8, compare also [48, Lemma 2.3.1], which is proven by [65, 40.20].

If one now wants to apply the concept of abstract Voronoi diagrams to a concrete case, one needs to verify whether the axioms from Definition 3 are fulfilled. Fortunately, the following lemma shows that it is not necessary to verify the axioms for all subsets of S; it is enough to consider all subsets of size three.

Lemma 10. *A curve system $\mathcal{J}$ is admissible if it fulfills the axioms stated in Definition 3 for all subsets $S' \subseteq S$ of size 3.*

The Voronoi diagrams of 3 sites can be categorized into four cases, see Figure 2.25. In the first one, all three bisectors intersect in exactly one point. This would, e. g., be the case in the euclidean diagram of point-sites in general position. In the second case the bisectors intersect in exactly two points. This can occur in the diagram of line segments, where a short line segment p is in the middle, and two longer ones q and r to the right and left of it. In the third case the bisectors are disjoint and all regions are nonempty, something that happens for three collinear points. In the fourth case the bisectors are again disjoint, but now the region of r is empty, this happens in the farthest diagram of three collinear points.

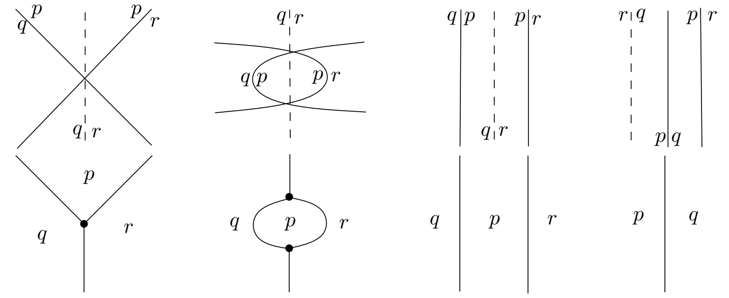


Figure 2.25: The four cases of an abstract Voronoi diagram of 3 sites p, q, and r. The upper drawings show the bisector system and the lower ones the corresponding Voronoi diagrams.

Another very useful fact is that we can define a total ordering on the set of sites. For $p, q \in S$ and $x \in \mathbb{R}^2$ we define

$$p <_x q \iff x \in D(p,q).$$

In a concrete setting this would mean that x is closer to the site p than to q. For each $x \in \mathbb{R}^2$ not lying on any bisecting curve, we can sort the set $S = \{p_1, \ldots, p_n\}$ according to the ordering. Let $p_1 <_x p_2 <_x \ldots <_x p_n$, then this gives us a sorted order of the distances from x to the sites. Most of all we will need this in Chapter 5 to consider *higher order abstract Voronoi diagrams.* But let us first get back to order-1 diagrams.

Klein, Mehlhorn, Meiser, and Ó'Dúnlaing [60, 53] have developed a randomized incremental algorithm to compute abstract Voronoi diagrams in expected time $O(n \log n)$. They start with the diagram of three sites and then incrementally add the sites region by region, assuming that the insertion order is randomized. Each time a new region shall be inserted, the edges of the old diagram intersected by the new region have to be detected. We say that these edges are in *conflict* with the new site. This is facilitated by a *history graph*, storing all edges ever constructed in a DAG. New edges arising, when inserting a new region, are made a successor of those nodes of the history graph representing edges of the current diagram intersected by the new edge. This makes sure that the edges of the current diagram always are leaves of the graph, and those which are in conflict with the new region can be found in expected time $O(\log n)$, by walking through the history graph, starting from its root, along its vertices in conflict with the new site. Altogether, the diagram can be constructed in expected time $O(n \log n)$ and expected space $O(n)$.

The algorithm and its construction time is valid for all concrete Voronoi diagrams, whose bisector systems are admissible in the sense of Definition 3. This condition is met by the diagram of point sites in all L_p-metrics under a general position assumption (Section 2.2.1), additively weighted Voronoi diagrams (Section 2.2.2), the diagram of non-crossing line segments (Section 2.2.3), and convex polygons of constant size (Section 2.2.4).

Furthermore, for point sites, a metric d can be defined, satisfying that points in general position have unbounded bisecting curves, d-circles are of constant algebraic complexity, each d-circle contains an L_2-circle and vice versa, and for any two points $a \neq c$ there is a third point $b \neq a, c$ such that $d(a, c) = d(a, b) + d(b, c)$ holds. Such a metric d is called *nice metric* and its bisector system with respect to a given set of point sites is admissible. This includes all convex distance functions of constant complexity, and also the Karlsruhe metric (Section 2.2.5).

Another example are disjoint compact, convex objects under the L_2- or Hausdorff-metric, where the distance from a point x in the plane to a compact, convex set A is measured by the minimum distance $d(x, a)$ from x to any point $a \in A$ with respect to a metric d. More applications of abstract Voronoi diagrams can be found in [1, 6, 7, 13, 15, 45, 60].

Not covered by the current approach are bisector systems resulting in disconnected Voronoi regions, or closed bisecting curves, as can occur for multiplicatively weighted Voronoi diagrams (Sections 2.2.2), anisotropic Voronoi diagrams (Section 2.2.5), and the farthest Voronoi diagram of line segments (Section 2.2.6). Such Voronoi diagrams will be addressed in the following two chapters.

Chapter 3

Disconnected Regions

In Section 2.3 the first setting of *abstract Voronoi diagrams* introduced by Klein in [48, 14] has been discussed. A crucial point was axiom (A2), stating that all Voronoi regions are path-connected. Yet, there are concrete Voronoi diagrams where this axiom fails to hold. In this chapter we allow Voronoi regions to be disconnected and explain how to compute them. By combining the randomized incremental construction technique [60] with trapezoidal decompositions [66] we obtain an algorithm that runs in expected time $O(s^2 n \sum_{j=3}^{n} m_j/j)$ and space $O(\sum_{j=4}^{n} m_j)$, where s is the maximum number of faces a Voronoi region in a subdiagram of three sites can have, and where m_j denotes the average number of faces per region in any subdiagram of j sites. In the connected case, where $s = 1 = m_j$, this results in the known optimal bound $O(n \ \sum_{j=3}^{n} 1/j) \ = \ O(n \log n)$.

A preliminary version of this chapter has been presented at ISAAC'13 [17].

3.1 Motivation

One example of Voronoi diagrams, where Voronoi regions can be disconnected, arises from the expanding-circle-view, presented in Section 2.2.2. Here, we have point-shaped sites from which circles expand. Where two circles meet we get a Voronoi edge and where three or more circles meet, we get a Voronoi vertex, see the left drawing of Figure 3.2. If the circles start to expand at different times, we obtain additively weighted Voronoi diagrams, and if the circles expand with different speeds, we obtain multiplicatively weighted Voronoi diagrams. In the latter case, Voronoi regions can be disconnected, but also axiom (A1) is violated, because in general all bisectors are closed curves. Such diagrams will be discussed later in Chapter 4. In this chapter we deal with unbounded bisecting curves only.

Lets keep the expanding-circle-view for a while and suppose that the circles may vary their expansion-speeds. For simplicity, assume that each circle either expands with very high, nearly infinite, speed, or its speed is nearly zero. Each circle may change its speed up to some constant s times. This means that the radius of each circle increases by a step function, with $O(s)$ steps, as depicted in Figure 3.1. At segments drawn vertically, the radius functions $p(t), q(t)$, and $r(t)$ increase very quickly, while being constant in between. The rightmost horizontal edges have an identical, gently increasing, slope. If, at time t, $|p(t) - q(t)| <$

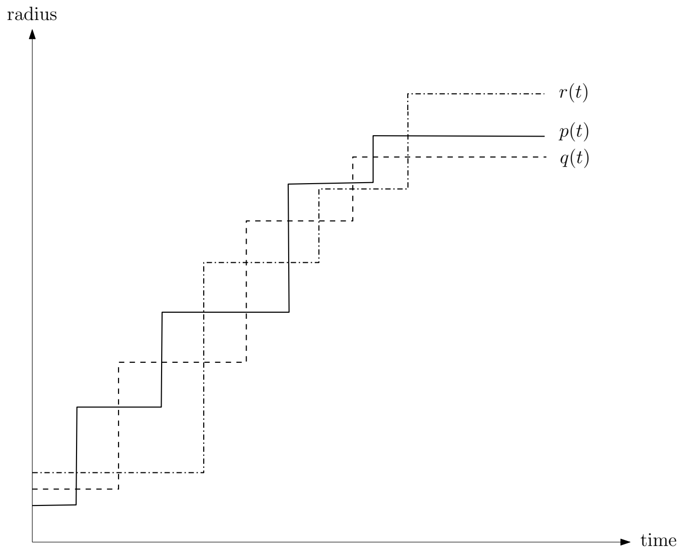


Figure 3.1: Expanding circles leading to the Voronoi diagram shown in Figure 3.2.

$|p-q] < p(t)+q(t)$ holds then the two circles of radii $p(t)$ (resp. $q(t)$) centered at p (resp. at q) intersect twice and thus define two points on the bisector of p and q. When $q(t)$ stays constant while $p(t)$ increases, these two bisector points move along the circle of radius $q(t)$ around q, writing out circular bisector arcs (an extreme case of a multiplicatively weighted Voronoi diagram). Clearly, such arcs can be shared by the bisectors $J(p,q)$ and $J(q,r)$. The slightly increasing tails of the radius functions lead to circular bisector segments, as in additively weighted Voronoi diagrams.

The complexity of the Voronoi diagram of three sites can be bounded as follows.

Lemma 11. *Let $\varepsilon > 0$ be an arbitrarily small number and let the radius functions $p(t), q(t)$, and $r(t)$ be piecewise linear, with graphs consisting of $\leq s$ line segments each, with slopes either $< \varepsilon$ or $> \frac{1}{\varepsilon}$. Furthermore, any two radius functions intersect in at most $s-1$ points. Then ε can be chosen such that the Voronoi diagram $V(\{p,q,r\})$ has at most $3s$ Voronoi vertices.*

Proof. If x is a Voronoi vertex, then all three circles of p, q, and r intersect at this point at the same time. We can choose $\varepsilon > 0$ such that an intersection of all three circles occurs only while one of the circles is expanding with high slope $> \frac{1}{\varepsilon}$. Each time a circle is expanding with high slope, it can overrun the two intersection points of the other two circles. Thus during each expansion phase there can arise at most 2 vertices. Because each circle expands with high slope at most $\frac{s}{2}$ times, and there are 3 circles, $V(S)$ can get at most $3s$ Voronoi vertices. □

This means that each Voronoi region in $V(\{p,q,r\})$ has at most $O(s)$ connected components. The diagram of 3 sites p, q, r may then look like depicted in the right drawing of Figure 3.2.

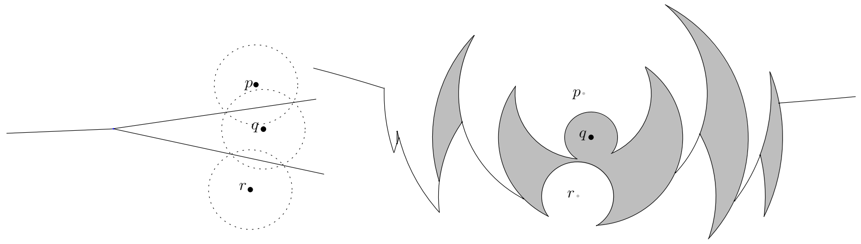


Figure 3.2: The Voronoi diagram of three sites, to the left with expanding circles not variating their speeds, and to the left with variating speeds.

If an expanding circle never completely overruns another one, then all bisecting curves are unbounded and consist of $O(s)$ circular arcs each. Formally, this means that $p(t) < q(t) + |p - q|$ for all sites p and q. In this case our algorithm, discussed in this chapter, can be adapted. As will be seen later, the crucial point is to determine the expected number of faces of a Voronoi region in any diagram of a subset of j sites of the given site-set. This number may strongly depend on the chosen radius functions.

Now, let us explicitly define which Voronoi diagrams we are considering in this chapter. We still require the following two axioms, claimed to hold for each subset S' of S, compare Definition 3 from Section 2.3.

(A1) By stereographic projection to the sphere, each curve $J(p, q)$ can be completed to a closed Jordan curve through the northpole.

(A3) Each point of the plane belongs to the closure of a Voronoi region $\overline{VR(p, S')}$.

But instead of axiom (A2), we require the following relaxed version,

(A2') For any $p, q, r \in S$, each Voronoi region in $V(\{p, q, r\})$ has at most s path-connected components.

Also, we assume the finite intersection property, which directly implies that our AVD has an intuitiv graph structure.

(A4) For any two curves $J(p, q)$ and $J(r, t)$, their intersection has only finitely many connected components.

Axiom (A2') implies that two bisecting curves $J(p, q)$ and $J(q, r)$ sharing a site q can cross at most $O(s)$ times, whereas arbitrary bisectors $J(p, q)$ and $J(r, t)$ not sharing a site can have any finite number of intersections. While this axiom bounds the possible number of faces per region in all AVD's of three sites of S, in an AVD of $n > 3$ sites a single region may have as many as $\Theta(sn^2)$ faces; see Lemma 18 below. To capture the complexity of the input system of bisecting curves we denote by m_j the average number of faces per region, over all AVD's of j sites from S.

Furthermore, to make use of trapezoidal decompositions [66] we will in Section 3.5 require one additional assumption to bound the number of trapezoids in the decomposition.

(A5) Each bisector $J(p, q)$ has constantly many points of vertical tangency.

This holds, e. g., for algebraic functions of constant algebraic degree. Moreover, given a direction of $J(p,q)$ and two points x and y on it, we assume that one can in constant time decide which point is visited first. Furthermore, for simplicity we assume general position in the sense that no two distinct points of vertical tangency lie on the same vertical line. These are assumptions commonly made in the design of algorithms for computing arrangements of curves, [68].

In this chapter we are proving the following result.

Theorem 3. *Given a system of bisecting curves that satisfies Axioms A1, A2', A3, A4, and A5. Then its abstract Voronoi diagram $V(S)$ can be constructed in expected time*

$$O\left(s^2 n \sum_{j=3}^{n} \frac{m_j}{j}\right).$$

Theorem 3 can be seen as a generalization of the optimal bound obtained in [60] for AVD's with connected regions, because in this case $s = 1 = m_j$ holds and $\sum_{j=3}^{n} 1/j \in O(\log n)$. However, it seems that the standard randomized incremental construction method does not generalize to the disconnected case in a straightforward way. One difficulty is that more conflict information must be stored, in order to locate all faces of the region of a new site. But then one cannot easily bound the outdegree of the history graph. We overcome this difficulty by maintaining a trapezoidal decomposition [66] of the current AVD during the incremental construction.

The rest of this chapter is organized as follows. In Section 3.2 we present some basic facts and in Section 3.3 complexity results about AVD's with disconnected regions. In Section 3.4 we discuss preliminary observations with respect to an algorithm. We proceed to studying it in Section 3.5 and carry out the analysis in Section 3.6.

3.2 Basic Facts

In this section we present some basic facts about abstract Voronoi diagrams with disconnected regions, which are akin to facts about the old AVD's with connected regions, and can be proven similarly, compare Section 2.3. Let our given system of bisecting curves satisfy axioms A1, A2', A3, and A4, and let s denote the maximum number of connected components of a Voronoi region $\mathrm{VR}(p, S')$, where $p \in S' \subseteq S$, $|S'| = 3$, $|S| = n$.

Lemma 12. *Let $p, q, r \in S$. Then $D(p,q) \cap D(q,r) \subseteq D(p,r)$.*

Proof. The same as the proof of Lemma 5 in [49], since it only uses axiom (A4). □

Lemma 13. *The faces of a Voronoi region and their closures are simply connected.*

Proof. Lemma 2.2.4 in [48] applied on a connected component of a Voronoi region and its closure. □

Because of axiom (A3), the following "piece of pie" lemma can be shown, which compares with Lemma 11 in [49], see also Figure 3.3. One of the main differences is, that several faces of the same region may be adjacent to the same vertex, like depicted in Figure 3.6c.

Lemma 14. *Let S' be a subset of S. For each point $v \in V(S')$ there exists an arbitrarily small neighborhood U of v, whose boundary is a simple closed curve, such that the following holds.*

(i) Either v is an interior point of some Voronoi edge e of $V(S')$ separating the Voronoi regions of p and q. Then $p \neq q$ and U is divided by e in exactly two domains, one contained in $VR(p, S')$, the other in $VR(q, S')$.

(ii) Or v is a Voronoi vertex of $V(S')$, of degree $k \geq 3$. After suitably renumbering S', the Voronoi edges e_i incident to v separate the Voronoi regions of p_i and p_{i+1} in clockwise order, where $0 \leq i \leq k-1$ is counted mod k. The edges e_{i-1} and e_i together with ∂U, bound a piece of pie contained in $VR(p_i, S')$; these pieces are domains with Jordan curve boundaries. The sites $p_0, \ldots, p_{k-1}$ do not have to be pairwise different, but $p_0 p_1 \ldots p_{k-1}$ is a Davenport-Schinzel-Sequence of order 2, and if $p_i = p_j$ for $i \neq j$, then the pieces of pie of p_i and p_j belong to different faces of the Voronoi region of $p_i = p_j$.

Proof. We will prove only the last part of (ii) saying that $p_0 p_1 \ldots p_{k-1}$ is a Davenport-Schinzel-Sequence of order 2, and if $p_i = p_j$ for $i \neq j$, then the pieces of pie of p_i and p_j belong to different faces of the Voronoi region of $p_i = p_j$. For the rest see Lemma 11 in [49].

For the sake of a contradiction, let $p \neq q$, and assume that the faces around v belong to the sites $\ldots p \ldots q \ldots p \ldots q \ldots$ in this order. These faces have to be separated by the bisector $J(p, q)$. But then for any $\epsilon > 0$, the bisector $J(p, q)$ has to appear 4 times in U, implying that it can not be a simple curve, a contradiction.

Next, suppose that $p_i = p_j$ and the pieces of pie of p_i and p_j belong to the same face. Then the closure of this face would not be simply connected, a contradiction to Lemma 13. □

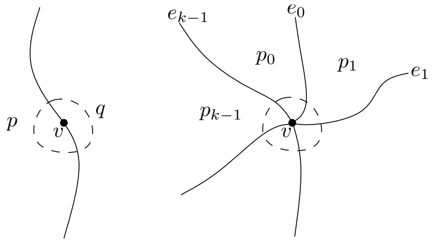


Figure 3.3: Piece of Pie Lemma

The Piece-Of-Pie-Lemma also shows that each point of the plane either belongs to a Voronoi region, or a Voronoi edge, or is a Voronoi vertex. An edge e is defined by two sites p and q, whose regions are separated by e. Voronoi vertices v are defined by at least three sites p, q, r, whose regions meet in v.

Theorem 4. *$V(S)$ is a finite planar graph with $O(\#faces\ of\ V(S))$ vertices and edges.*

Proof. By Lemma 14, $V(S)$ is a finite planar graph. From the Euler formula it follows, that $V(S)$ has $O(\#\text{faces of } V(S))$ vertices and edges. □

3.3 Complexity of V(S)

To estimate the running time of our algorithm computing abstract Voronoi diagrams, it is important to know their complexity. For connected regions it is known, that the size of $V(S)$ is $O(n)$, because it is a planar graph with $O(n)$ faces. But what happens if regions are allowed to consist of several faces? Surprisingly, already when a region in a diagram of 3 sites may have up to 2 faces, the total diagram $V(S)$ can have a complexity of $\Theta(n^2)$, as will be seen below.

In the sequel it will sometimes be helpful to restrict attention to the "finite part" of $V(S)$. For this purpose define Γ to be a large simple closed curve, intersecting each bisecting curve $J(p,q)$, $p,q \in S$ exactly twice, such that in the outer domain of Γ any two bisectors $J(p,q)$ and $J(r,t)$ are either equal or disjunct. If we add Γ to $V(S)$ and cut off all parts contained in the outer domain of Γ, we obtain a connected graph without unbounded edges.

In the next two lemmata we want to connect the number of faces of a Voronoi region with the number of intersection points of the corresponding bisectors. It may not surprise that these numbers are of the same order.

Lemma 15. *$V(\{p,q,r\})$ has at most $6s-4$ Voronoi vertices, i. e., $J(p,q)$ and $J(p,r)$ can intersect in at most $6s-4$ points that result in a Voronoi vertex in $V(\{p,q,r\})$. Further, this bound is tight.*

Proof. Let $S = \{p,q,r\}$ and consider $V(S) \cup \Gamma$. Cut off all edges outside of Γ and define this as the graph G. Then G is a connected finite planar graph with at least two more vertices and exactly one more face than $V(S)$. Now the Euler formula tells us $v-e+f = c+1$. Because of our assumption, $f \le 3s+1$, $c=1$, and because each Voronoi vertex is of degree ≥ 3, we have $3v \le 2e$. Substituting this into the Euler formula we obtain $v \le 6s-2$ for G and $v \le 6s-4$ for $V(S)$. Figure 3.4 shows an example where the bound is tight. It can be extended to all $s \ge 1$. □

Let x be a connected component of the intersection of two bisectors $J(p,q) \cap J(r,t)$. We say that $J(p,q)$ and $J(r,t)$ intersect *transversally* in x, if $J(r,t) \subset D(p,q)$ right before x, and $\subset D(q,p)$ right after x, or vice versa. For computing Voronoi diagrams only points resulting in Voronoi vertices are of interest. Thus, because of the previous lemma, we can in the following assume, that each pair of p-bisectors, bisectors $J(p,q)$ and $J(p,r)$ sharing site p, intersects transversally in at most $6s-4$ points, and each intersection point results in a Voronoi vertex of $V(\{p,q,r\})$.

Lemma 16. *Let $S = \{p,q,r\}$. If $J(p,q)$ and $J(p,r)$ intersect in at most $s-1$ points, then $VR(p,S)$ has at most s connected components.*

Proof. There are at most $s-1$ p-vertices (vertices bordering the Voronoi region of p) in $V(S)$. Each p-vertex borders at most two different faces of the p-region. Further, the p-region can have at most two unbounded faces. Thus, all other faces must have at least two p-vertices on their boundary. Altogether, p can have at most s connected components. □

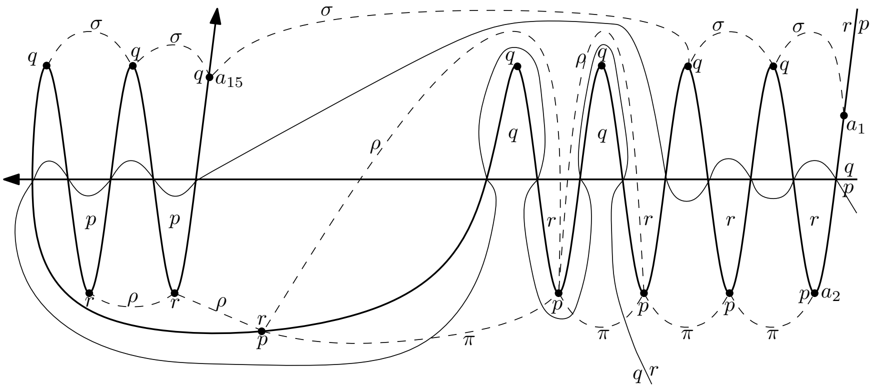


Figure 3.4: An example where $J(p,q)$ and $J(p,r)$ intersect in $6s-4=14$, $s=3$, points resulting in as many Voronoi vertices in $V(\{p,q,r\})$. Each π is a path $\subset \mathrm{VR}(p,S)$, $\sigma \subset \mathrm{VR}(q,S)$, and $\rho \subset \mathrm{VR}(r,S)$.

In the next lemma we use the notation $\lambda_s(n)$, which stands for the maximum length of a (n,s) *Davenport-Schinzel sequence*, see [68] and Definition 1, 2, and Lemma 3 in Section 2.1.1. The idea of how to prove the lower bound in the following lemma is from Agarwal [2].

Lemma 17. *Let F be a face of $V(S)$. Then F has a complexity of $O(\lambda_{6s-4}(n-1))$. For $s \geq 4$, there exists an AVD having a face of complexity $\Omega(\lambda_{s-1}(n-1))$.*

Proof. There are $(n-1)$ p-bisectors of which each pair is allowed to intersect transversally at most $6s-4$ times, see Lemma 15. The upper bound follows from the results for arbitrary arrangements of Jordan curves, see [68].

To show the lower bound, choose an arrangement of unbounded Jordan curves $J(p_1,p_2)$, $\ldots$, $J(p_1,p_n)$, where each pair of them has at most $s-1$ intersections, such that there is a face F having complexity $\Omega(\lambda_{s-1}(n-1))$. Now the Jordan curves are our bisecting curves, and by suitably labeling them we can claim $F \subseteq D(p_1,p_i)$ for all $2 \leq i \leq n$. Thus, F is a face of the Voronoi region of p_1. To show how this curve arrangement can be turned into an AVD, we define $J(p_i,p_j)$, $i<j$ equal to $J(p_1,p_j)$, and $D(p_i,p_j) = D(p_1,p_j)$. It is easy to see that if $x \in \mathbb{R}^2 \setminus V(S)$, then either $x \in \mathrm{VR}(p_1,S)$ or $x \in \mathrm{VR}(p_i,S)$, where i is maximal such that $x \in D(p_i,p_1)$. By our definition each pair of bisectors intersects in at most $s-1$ points, hence, because of Lemma 16, each Voronoi region has at most s connected components. □

We observe that this lemma does not imply the same complexity bound for a whole region of a site p, quite in contrary a region with several faces can have complexity $\Theta(sn^2)$, see also Figure 3.5 (for $s=2$).

Lemma 18. *The number of faces and the complexity of a region of $V(S)$ is $O(sn^2)$. This bound can be attained.*

Proof. There are $O(n)$ many p-bisectors of which each pair intersects in at most $O(s)$ points, implying that there are $O(sn^2)$ p-vertices in $V(S)$. Two edges of $V(S)$ having the same endpoints must belong to different bisectors, otherwise the bisector would be closed. That means, that two vertices having several parallel

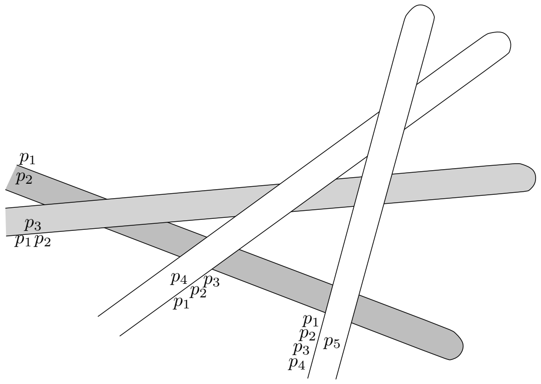


Figure 3.5: Here $s = 2$ and the Voronoi region of p_1 has $\Theta(sn^2)$ many faces, all faces outside the sticks.

edges count as an intersection for each pair of bisectors contributing to these edges. This implies the upper bound $O(sn^2)$ for the complexity of the p-region, which again is an upper bound for the number of connected components.

To show the lower bound, choose an arrangement of unbounded Jordan curves $J'(p_1, p_2), \ldots, J'(p_1, p_n)$, where each pair of them has at most $\frac{s-1}{4}$ intersections, such that the arrangement has $\Omega(sn^2)$ faces. Such an arrangement exists, see [68] and Figure 3.5 (for $s = 2$). Let Γ be a circle around the arrangement, large enough such that there are no more intersections outside of it. Now choose a circle C with sufficiently small radius centered at the origin. For each Jordan curve $J'(p_1, p_i)$ of the arrangement, we cut off one endpiece outside of Γ and generate the Minkowski sum of the remaining curve with the circle C. Let this curve be $J(p_1, p_i)$ and let $D(p_1, p_i)$ be outside the Minkowski sum and $D(p_i, p_1)$ inside it. Now, each intersection point of the previous curves $J'(p_1, p_i)$ and $J'(p_1, p_j)$ results in 4 intersection points of the new bisectors $J(p_1, p_i)$ and $J(p_1, p_j)$, which means, that each pair of bisectors in the new arrangement has at most $s - 1$ intersections. To get an admissible set of bisector $\{J(p_i, p_j) | i \neq j\}$, choose, like in the previous proof, $J(p_i, p_j)$, $i < j$, equal to $J(p_1, p_j)$ and $D(p_i, p_j) = D(p_1, p_j)$. Now, the Voronoi region of p_1 has $\Omega(sn^2)$ many faces and so is its complexity. □

Lemma 19. *The complexity and number of faces of $V(S)$ is $O(sn^3)$.*

Proof. It follows directly from the fact, that each region has a complexity and number of connected components in $O(sn^2)$, Lemma 18. □

Theorem 5. *Let $J := \{J(p, q) : p \neq q \in S\}$, $|S| = n$, be a curve system fulfilling axioms A1, A2', A3, and A4 for all subsets $S' \subseteq S$ of size 3. Then J fulfills axioms A1, A3, and A4 for all subsets $S' \subseteq S$ of size ≥ 3, and each Voronoi region has $O(sn^2)$ faces.*

Proof. It remains to show that (A4) is still valid for larger sets of sites. This follows from the transitivity, Lemma 12, compare the proof of Lemma 13 in [49]. □

3.4 Towards an Algorithm

We present a randomized incremental algorithm, which computes abstract Voronoi diagrams with disconnected regions. Let $R \subset S$, t a site of S not in R, and $\mathcal{T} := \mathrm{VR}(t, R \cup \{t\})$ its new Voronoi region. Allowing Voronoi regions to be disconnected causes some new technical phenomena, which did not occur in the connected case [53]. For example, the intersection of a face of the new region $\mathcal{T}$ with the old Voronoi diagram $V(R)$ need not be connected. Or there may be so-called "touch points" with respect to t where different faces of the region of t meet (Figure 3.6c).

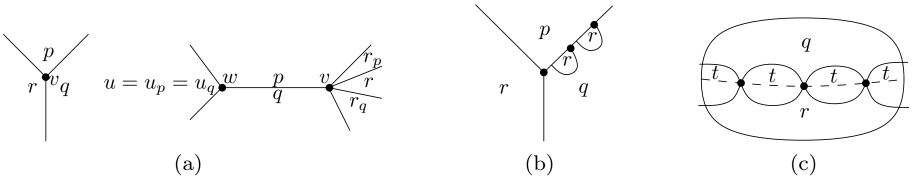


Figure 3.6: (a) A *pqr*-vertex v and *prqu*-edge e with description $D_R(e) = \{(r_q, q, p, r_p, (x, y)),\ (u_p, p, q, u_q, (x', y'))\}$, where $v = (x, y)$ and $w = (x', y')$. (b) There can be s different *pqr*-vertices in $V(\{p, q, r\})$. (c) Touch points with respect to t on an edge of $V(R \setminus \{t\})$.

All these difficulties can be dealt with, as discussed in Section 3.4.1. Fortunately, it is still true, that the fate of an edge e of $V(R)$ upon insertion of site t, can be decided locally. Suppose that in $V(R)$ edge e separates faces of p and q, and that its endpoints are defined by two more sites r and u. Then e is intersected by the new region $\mathrm{VR}(t, R \cup \{t\})$ of t in $V(R)$, if and only if e is intersected by the region $\mathrm{VR}(t, \{p, q, r, u, t\})$ of t in the Voronoi diagram of the four sites p, q, r, u, see Lemma 4 in [53]. The same holds, if there is a touch point v with respect to t on e. Then v is still a touch point with respect to t on e in the diagram of the five sites p, q, r, u, and t, see Lemma 21 below. Therefore, our algorithm will be built on a basic operation, which is only slightly more complicated than the one used in [53]. Recall that Voronoi edges e and Voronoi regions $\mathrm{VR}(p, S)$ are defined as open sets, i. e., edges e do not contain their endpoints, and regions $\mathrm{VR}(p, S)$ do not contain their boundaries. With $\overline{\mathrm{VR}(p, S)}$ we denote the closure of the Voronoi region.

Further, we use the same definitions to describe vertices v and edges e of $V(R)$ by sites as in [53]. A vertex v of $V(R)$ is called a *pqr-vertex*, if there is a p-, q-, and r-region in clockwise order around v and an edge e of $V(R)$ is called a *prqt-edge* if e separates a p- and q-region, and its endpoints are *prq*- and *qtp*-vertices, see Figure 3.6a.

Let e be an edge with endpoints v and w separating p- and q-region. Further, let r_q and r_p be the preceding and following regions of q and p in clockwise order around v, and u_p and u_q the preceding and following regions of p and q in clockwise order around w. Then the *description* $D_R(e)$ with respect to $R \subseteq S$ is the set $\{(r_q, q, p, r_p, \mathrm{coor}(v)), (u_p, p, q, u_q, \mathrm{coor(w)})\}$, where $\mathrm{coor}(v)$ and $\mathrm{coor}(w)$ are the coordinates of v and w respectively, see again Figure 3.6a. Because of Lemma 13, no two edges bounding p- and q-regions can have the same endpoints. Hence, each edge has a unique description. This would not be the case, if the description was defined without the coordinates, see Figure 3.6b. Moreover, with $\operatorname{set}(D_R(e))$ we denote the set $\{r_p, r_q, u_p, u_q, p, q\}$.

Basic Operation
Input: A 5-tuple (p, r, q, u, t), such that

(1) $V(\{p, r, q, u\})$ contains a $prqu$-edge e, and

(2) $t \notin \{p, r, q, u\}$.

Output: The combinatorial structure of $e \cap \overline{\mathrm{VR}(t, \{p, r, q, u, t\})}$ in the form of:

- Number of connected components of $e \cap \overline{\mathrm{VR}(t, \{p, r, q, u, t\})}$,
- The placement of the connected components of $e \cap \overline{\mathrm{VR}(t, \{p, r, q, u, t\})}$ on e, i. e., the coordinates of the endpoints of $e \cap \overline{\mathrm{VR}(t, \{p, r, q, u, t\})}$. A special case is $e \subset \overline{\mathrm{VR}(t, \{p, r, q, u, t\})}$.
- Touch points with respect to t on e by their coordinates.

Lemma 20 below shows that the intersection $e \cap \overline{\mathrm{VR}(t, \{p, r, q, u, t\})}$ can have at most $O(s)$ connected components. Moreover, by Lemma 22 there are $O(s)$ touch points with respect to t on an edge e of $V(R)$. That is why we charge $O(s)$ time to each call of the Basic Operation.

However, of much greater concern is that the history graph used in [53] for finding the edges e of $V(R)$ intersected by the region of the new site, t, does no longer work. In Figure 3.7a, an example is depicted. Here, the sites p, r, t are inserted in this order. The edge e emerges after inserting r and is made a successor of the edges $e_1, \ldots, e_5$ in the history graph. The region of t intersects edge e but none of the edges $e_1, \ldots, e_5$. This can happen, because $\mathcal{T}$ disconnects the region of p, implying that we will not be able to find this intersection by walking through the history graph along the edges intersected by $\mathcal{T}$, see Lemma 14 in [53].

Thus, we need a history graph with more information, but still with bounded outdegrees in order to avoid super-linear cost for walking through it. For this purpose, we will maintain a trapezoidal decomposition, see [66]. To make sure that the number of trapezoids is bounded, we will in the following require the additional axiom (A5) from Section 3.1. Furthermore, for simplicity, we assume that no two points of vertical tangency lie on the same vertical line.

3.4.1 Some Technical Issues

For an AVD with connected regions $V(R) \cap \overline{\mathcal{T}}$, where $\mathcal{T} := \mathrm{VR}(t, R \cup \{t\})$, is a connected set [53], whereas for disconnected regions it can split up. The intersection of $\overline{\mathcal{T}}$ with an edge e of $V(R)$ can consist of up to $O(s)$ connected components. Note that Voronoi edges e (and Voronoi regions P) are defined as open sets, i. e., edges e do not contain their endpoints (likewise regions P do

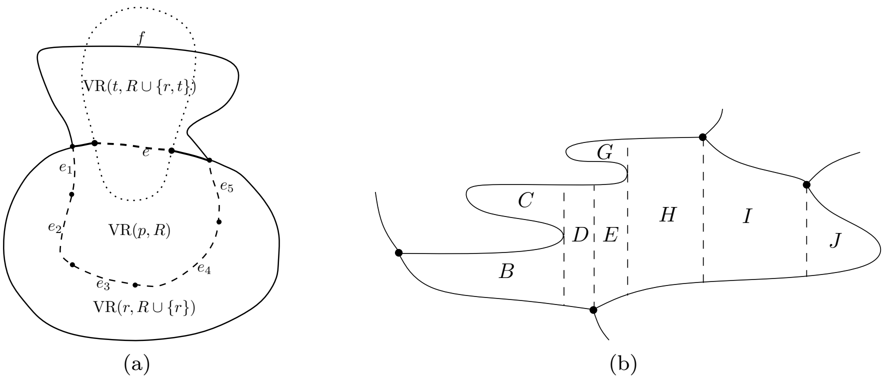


Figure 3.7: (a) The sites p, r, t are inserted in this order, the intersection between the region of t and the edge e is not found in the standard history graph from [53]. (b) Trapezoidal Decomposition of a face.

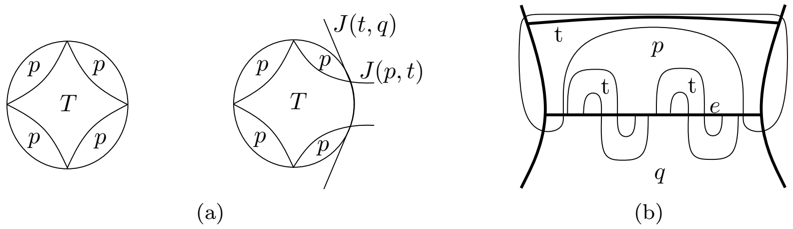


Figure 3.8: (a) Intersection between $V(R)$ and a connected component T of $\mathcal{T}$, in the left picture $J(p, t)$ would be closed. (b) Intersection between an edge e and $\mathcal{T}$. Here $s = 3$ and the intersection consists of $3s - 1 = 8$ connected components.

not contain their boundaries). Before we can start to describe the algorithm, we need to discuss some rather technical observations.

To make life easier, in the following, we will make use of the curve Γ defined in the beginning of Section 3.3. Without explicit notion, from now on $V(S)$ means $V(S)$ restricted to the bounded domain of Γ together with Γ itself, meaning that all Voronoi regions are bounded.

Lemma 20. *If $\mathcal{T} \neq \emptyset$, then for any face $T \subseteq \mathcal{T}$ and edge e of $V(R)$:*

(i) the intersection $V(R) \cap \overline{T}$ is an indiscrete set of points,

(ii) the intersection $e \cap \overline{\mathcal{T}}$ consists of at most $3s - 1$ connected components.

Proof. (i): Suppose $V(R) \cap \overline{T}$ consists of only discrete points. Since T is an open set, this means that T lies within a face of $\mathrm{VR}(p, R)$, which is bounded by definition. Then $\partial T \setminus \partial\mathrm{VR}(p, R) \subseteq J(p, t)$, but since $\partial T \cap \partial\mathrm{VR}(p, R)$ consists of only discrete points and $J(p, t)$ is continuous, it follows, that $\partial T \subseteq J(p, t)$, which means that $J(p, t)$ is a closed curve, a contradiction. See also Figure 3.8a.

(ii): Now let e be an edge separating a p- and q-region. Lemma 4 in [53] shows that $e \cap \overline{\mathcal{T}} = e \cap \overline{\mathrm{VR}(t, \{p, q, t\})}$, and since $e \subseteq J(p, q)$, the number of connected components of $e \cap \overline{\mathrm{VR}(t, \{p, q, t\})}$ is less or equal the number of connected components of $J(p,q) \cap \overline{\mathrm{VR}(t,\{p,q,t\})}$. By Lemma 15, $V(\{p,q,t\})$ can have at most $6s-4$ Voronoi vertices. Each bounded connected component of $J(p,q) \cap \overline{\mathrm{VR}(t,\{p,q,t\})}$ borders two vertices, and unbounded components border one vertex; there can be at most two unbounded components. Further, no two components of $J(p,q) \cap \overline{\mathrm{VR}(t,\{p,q,t\})}$ can border the same vertex, hence, there can be at most $3s-1$ components, see also Figure 3.8b. □

We use the Basic Operation to efficiently compute the intersection between $\mathcal{T}$ and the current Voronoi diagram $V(R)$. This works, because of the following lemma. Observe that the boundaries of two or more faces of a region $\mathrm{VR}(t,S)$ can intersect in only vertices. Because of Lemma 13, the boundaries of two faces can intersect in at most one point. We call such a point a *touch point* with respect to t. Because of Lemma 14, the faces of $\mathrm{VR}(t,R)$ bordering v have to be separated by different regions, see Figure 3.6c. This implies the touch point v to be positioned on the Voronoi diagram of $R \setminus \{t\}$, either on an edge separating q- and r-region, or on a vertex.

Lemma 21. *Let e be a prqu-edge of $V(R)$ and let v be a point lying on e. Then e is also a prqu-edge of $V(R')$ where $\{p,r,q,u\} \subseteq R' \subseteq R$. Further, for all $t \notin R$ it holds*

(i) $e \cap \overline{VR(t,R \cup \{t\})} = e \cap \overline{VR(t,R' \cup \{t\})}$.

(ii) v is a touch point with respect to t in $V(R \cup \{t\})$, iff v is a touch point with respect to t in $V(\{p,q,t\})$.

Proof. For a proof of the first part and (i) see the proof of Lemma 4 in [53]. To prove (ii) choose a sufficiently small neighborhood U of v such that $U \subseteq e \cup \mathrm{VR}(p,R) \cup \mathrm{VR}(q,R)$ and show that $U \cap \mathrm{VR}(t,R \cup \{t\}) = U \cap \mathrm{VR}(t,\{p,q,t\})$. Since $\mathrm{VR}(t,R \cup \{t\}) \subseteq \mathrm{VR}(t,\{p,q,t\})$, "$\subseteq$" is clear. Now let $x \in U \cap \mathrm{VR}(t,\{p,q,t\})$. Then $x \in D(t,p) \cap D(t,q)$, and, because of transitivity, $x \in \mathrm{VR}(t,R \cup \{t\})$. □

Lemma 22. *Let e be an edge of $V(R)$. In $V(R \cup \{t\})$ there are $O(s)$ touch points with respect to t on e.*

Proof. Let e separate the regions of p and q. By Lemma 21, v is a touch point with respect to t on e, iff v is a touch point with respect to t on e in $V(\{p,q,t\})$, see also Figure 3.6c. Thus, consider the diagram $V(\{p,q,t\})$. Put a vertex in each connected component of $\mathrm{VR}(t,\{p,q,t\})$ and for each touch point v on e draw an edge between two vertices corresponding to adjacent regions. Because the region of t in $V(\{p,q,t\})$ has at most s connected components, and the closure of the region is simply connected, the resulting graph is a tree with s vertices. Thus, it has $O(s)$ edges. □

3.5 Trapezoidal Decomposition

Let $V^*(R)$ be the vertical decomposition of $V(R)$, i.e., $V^*(R)$ decomposes each face of $V(R)$ into pseudo-trapezoidal cells, for brevity we call them trapezoids, see Figure 3.7b. Such decompositions have been introduced by Seidel [66], and they are also used in [68] to compute arrangements. Again edges e and trapezoids A of $V^*(R)$ are defined as open sets, i.e., e does not contain its endpoints, and A does not contain its boundary. Moreover, when we speak of

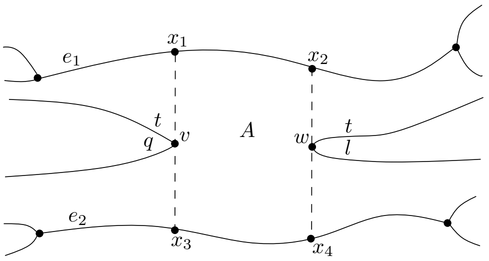


Figure 3.9: The description of the trapezoid A is the set $D_R(e_1) \cup D_R(e_2) \cup \{(q, r, \mathrm{coor}(v)), (l, l, \mathrm{coor}(w)), (\mathrm{coor}(x_1), \mathrm{coor}(x_2), \mathrm{coor}(x_3), \mathrm{coor}(x_4))\}$.

an "edge of $V^*(R)$" we always mean a Voronoi edge and not a vertical line segment. Also recall the definition $\mathcal{T} := \mathrm{VR}(t, R \cup \{t\})$, the region of the new site t, and remember that we always have the large curve Γ around our Voronoi diagram making it finite.

Definition 4. *We say that a site $t \in S \setminus R$* interferes *with an edge e of $V(R)$ or a trapezoid A of $V^*(R)$, if $\overline{\mathcal{T}}$ intersects $\overline{e}$ or $\overline{A}$ in more than finitely many points.*

There can be edges or trapezoids intersected in only finitely many points by $\overline{\mathcal{T}}$ and thus not being interfered by t in the sense of our definition. These edges and trapezoids are taken into account later during the process of inserting $\mathcal{T}$ into $V^*(R)$.

A general trapezoid A of $V^*(R)$ is bounded by two Voronoi edges and two vertical line segments. The vertical line segments have a Voronoi vertex as start- or endpoint, or they are bounded from the side by a point of vertical tangency, or by a vertex, whose incident edges all emanate to one side, see Figure 4.5.

By these observations we can define the *description* $D_R(A)$ of a trapezoid A. It is a set containing four 5-tuples, two 3-tuples, and one 4-tuple. Two of the 5-tuples each equal the descriptions of the two edges e_1 and e_2 bounding A from above and below. The two 3-tuples belong to the points bordering the vertical line segments of A from left and right. Let A be contained in the region of p and let v be the point bordering the left line segment of A. Further, the uppermost edge incident to v separates the regions of p and q and the lowermost edge incident to v separates the regions of p and r. Then the corresponding 3-tuple is defined as $(q, r, \mathrm{coor}(v))$. Analogously, we have a 3-tuple for the point bordering the left line segment of A. Finally, the 4-tuple contains the coordinates of the four corners of A, making the description unique. Observe that tuples may contain empty entries, e. g., if a vertical line segment has a vertex as start or endpoint and no other point bordering it from right or left. For an example see Figure 4.5.

Again, $\mathrm{set}(D_R(A))$ is the set of all sites contained in a tuple of $D_R(A)$. In the following we will not distinguish between a trapezoid A of $V^*(R)$ and its description.

Definition 5. *We say that a site $t \in S \setminus R$ is in conflict with an edge e of $V(R)$ or a trapezoid A of $V^*(R)$, if there is no edge or trapezoid with the same description in $V(R \cup \{t\})$ resp. $V^*(R \cup \{t\})$.*

Observe that if an edge or trapezoid interferes with t then it is also in conflict with t but not necessarily vice versa. This can happen, e. g., when an edge or trapezoid is intersected in only finitely many points by the region of t, or when an edge bounding several trapezoids is intersected in a component bordering only some of them. In the example depicted in Figure 4.7, trapezoid C is in conflict but does not interfere with t, because only the bottom right corner is intersected by $\overline{\mathcal{T}}$. Also, trapezoid A is in conflict with t without being intersected by $\overline{\mathcal{T}}$ at all, but the right endpoint of its bottom edge is intersected and thus the edge gets a new description.

Next we define a data structure, called *history graph*, to determine the trapezoids of $V^*(R)$ interfering with t. Let $R_i = \{p_1, \ldots, p_i\}$ be a set of sites inserted in this order and $R = R_j$. The history graph $\mathcal{H}(R)$ is a DAG with a single source and nodes

$$
\begin{aligned}
\{\text{source}\} \cup & \bigcup_{3 \le i \le j} \{D_{R_i}(A) | A \text{ is a trapezoid of } V^*(R_i)\} \\
\cup & \bigcup_{4 \le i \le j} \{D_{R_i}(e) | e \text{ is an edge on } \partial \mathrm{VR}(p_i, R_i)\},
\end{aligned}
$$

such that each trapezoid of $V^*(R)$ is a leaf of $\mathcal{H}(R)$.

Let a node of $\mathcal{H}(R)$ be called *trapezoid-node*, if it refers to a trapezoid, and *edge-node*, if it refers to an edge. Further, each node is linked to its corresponding trapezoid or edge in $V^*(R_i)$ and vice versa.

To construct $\mathcal{H}(R)$, we start with the diagram of three sites p_1, p_2, p_3 and make the trapezoids of $V^*(\{p_1, p_2, p_3\})$ the successors of the source.

Now, assume we already have constructed $\mathcal{H}(R)$ and let $t \in S \setminus R$, randomly chosen, be the next site to be inserted. Let e be an edge on the boundary of $\mathcal{T}$, meaning that e with its description $D(e)$ has not yet been part of $V(R)$. There are two cases:

1. Edge e intersects some trapezoids $A_1, \ldots, A_l$ of $V^*(R)$ in their interior. By assumption these trapezoids are leaves of $\mathcal{H}(R)$. Edge e is now made a successor of all $A_1, \ldots, A_l$, compare edges e_1, e_4 in Figure 4.7.

2. Edge e does not intersect any trapezoids of $V^*(R)$, meaning that e runs along edges $e_1, \ldots, e_l$ of $V(R)$, i. e., along boundaries of trapezoids of $V^*(R)$. Each of these edges e_i has some adjacent trapezoids on both sides. Because e lies on the boundary of $\mathcal{T}$, only one side of e_i can be intersected by $\mathcal{T}$ in an ε-neighborhood around e_i. Let $A_1, \ldots, A_l$ be the trapezoids adjacent to e_i on this side. Again, they are leaves of $\mathcal{H}(R)$ and e is now made successor of them. Compare edges e_2, e_3 in Figure 4.7.

Next let A be a trapezoid of $V^*(R \cup \{t\})$, which has not yet been part of $V^*(R)$. Again, there are two cases:

1. Trapezoid A is contained in a p-region for an old site $p \in R$. Then A is made a successor of all trapezoids A' of $V^*(R)$ intersected by A. Note, that the trapezoids are defined as open sets, hence, two trapezoids do not intersect if just their boundaries intersect. Compare trapezoids $A_1, C_1, D_1, F_1, F_2, F_3, G_1, H_1$ in Figure 4.7.

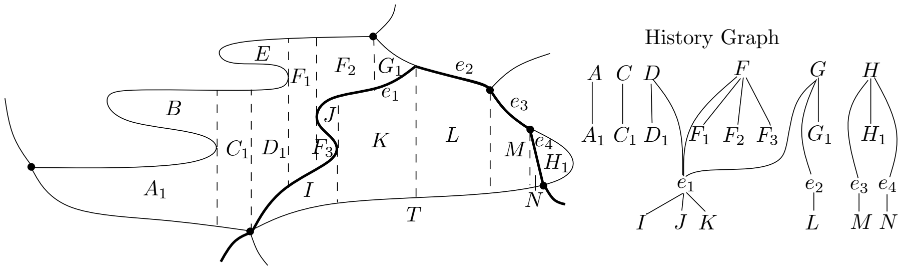


Figure 3.10: Face $T \subseteq \mathcal{T}$ is inserted in the trapezoidal decomposition of Figure 3.7b and the history graph is updated.

2. Trapezoid A is contained in $\mathcal{T}$. Then A is made a successor of the edges on the boundary of $\mathcal{T}$ bordering A. Compare Trapezoids I, J, K, L, M, N in Figure 4.7.

It follows directly from the constructions, that the trapezoids of $V^*(R)$ are indeed leaves of $\mathcal{H}(R)$.

For our further discussion we define the following two sets of trapezoids, compare Definitions 4 and 5.

$$\begin{aligned} & E_t := \{A \mid A \text{ is a trapezoid of } V^*(R) \text{ interfering with } t\} \\ \text{and} \quad & \overline{E_t} := \{A \mid A \text{ is a trapezoid of } V^*(R) \text{ in conflict with } t\}. \end{aligned}$$

We have $E_t \subseteq \overline{E_t}$, but not necessarily vice versa. This is, e. g., because there can be an edge bounding A, which is intersected by the region of t in a component not bordering A. Then there is a trapezoid A' in $V^*(R \cup \{t\})$ that has the same coordinates as A, but a different description. Thus, A is in conflict with t, but not interfering with it. For an example see the trapezoids A and A_1 in Figures 3.7b and 4.7, and observe that the description of the bottom edge has changed after inserting t. In this example $E_t = \{D, F, G, H\}$ and $\overline{E_t} = \{A, C, D, F, G, H\}$.

However, all parts of $V^*(R)$ intersected by $\overline{\mathcal{T}}$ are contained in $\bigcup_{A \in E_t} \overline{A}$. Thus, to compute the intersections between the current diagram $V^*(R)$ and $\mathcal{T}$, it is enough to compute the set E_t. Given this set we can then insert $\mathcal{T}$ into $V(R)$ and obtain the superset $\overline{E_t}$, which is needed to update all trapezoid-descriptions, during the process.

Further, to be able to analyse the algorithm we define

$$c := \sum_{i=3}^{j} \# \text{ trapezoids of } V^*(R_i) \text{ in conflict with } t.$$

For each edge of $V(R_i)$ in conflict with t, there is at least one trapezoid of $V^*(R_i)$ in conflict with t. On the other hand, each trapezoid is bounded by at most two edges, thus for each trapezoid in conflict with t, there are at most two edges in conflict with t. This implies:

$$\begin{aligned}
&\#\text{ edge-nodes of } \mathcal{H}(R) \text{ in conflict with } t\\
\le\ &2\ \#\text{ trapezoid-nodes of } \mathcal{H}(R) \text{ in conflict with } t\\
\le\ &2\ c.
\end{aligned}$$

Thus we have

$$\#\text{ nodes of } \mathcal{H}(R) \text{ in conflict with } t \ \le\ 3c.$$

3.5.1 Computation of E_t

As mentioned before, we will first compute the set E_t. Given E_t we will then be able to efficiently insert $\mathcal{T}$ into $V^*(R)$ and determine the full set $\overline{E_t}$ during the process. To test if a trapezoid $A \subseteq \mathrm{VR}(p, R)$ interferes with t, we use the Basic Operation to determine if $\overline{\mathcal{T}}$ intersects the edges bounding A in more than finitely many points. To check if the vertical line segments bounding A are intersected by $\overline{\mathcal{T}}$, it is enough to check if the bisector $J(p, t)$ intersects the vertical line segment, and if not, whether it lies in $D(p, t)$ or $D(t, p)$. Because of our assumption, a vertical line test takes constant time.

Now we compute E_t by walking through $\mathcal{H}(R)$ by breadth-first-search along the nodes interfering with t. Let A be a trapezoid-node created during the insertion of r_i. If t interferes with A, then we also test recursively the at most 4 trapezoids of $V^*(R_i)$, which are adjacent to A by a vertical line segment (we find them via $V^*(R_i)$), for interference with t.

Lemma 23. *By walking through $\mathcal{H}(R)$ as described above, we will reach all leaves of $\mathcal{H}(R)$ interfering with t.*

Proof. Let $A \in V^*(R)$ be a leaf of $\mathcal{H}(R)$ interfering with t. W.l.o.g., we assume that A has not been a trapezoid of $V^*(R \setminus \{p_j\})$. If A is contained in a q-region for a $q \in R \setminus \{p_j\}$, then $\overline{A} \subseteq \bigcup_{A' \text{ predecessor of } A} \overline{A'}$, and hence, at least one predecessor A' of A interferes with t. Otherwise A is a trapezoid contained in a face F of the region of p_j. Each face of $\mathcal{T}$ intersecting $\overline{A}$ in more than finitely many points also intersects an edge e on the boundary of F and a trapezoid $A' \subseteq F$ bounded by e in a nonfinite set. If e bounds A, then e is a predecessor edge-node of A interfering with t, otherwise we find A by testing the neighbors of A' recursively for interference with t. □

The next question is how much time we need to compute E_t. Of course we need at least as much time as there are nodes in $\mathcal{H}(R)$ interfering with t. But additionally for all these nodes, we need to test all their successor-nodes for interference with t. Fortunately, for each edge-node e interfering with t, all its successors are in conflict with t. This holds, because all successors are trapezoids bounded by e. Thus, it suffices to bound the outdegree of trapezoid-nodes.

Lemma 24. *The outdegree of each trapezoid-node of $\mathcal{H}(R)$ is $O(s)$.*

Proof. Each trapezoid $A \subseteq \mathrm{VR}(p, R)$ of $V^*(R)$ has at most 2 edges and 2 vertical line segments on its boundary. An edge e on the boundary of A is part of $V(R)$ and can be intersected in at most $O(s)$ many components by the region of a new site t, see Lemma 20 and 22. Further, at most constantly many points of the boundary of the region of t with vertical tangency can appear in A, because

$J(p,t)$ has constantly many points of vertical tangency. Altogether there can be at most $O(s)$ many edges of the boundary of $\mathcal{T}$ intersecting $\overline{A}$ and $O(s)$ many trapezoids of $V^*(R \cup \{t\})$ contained in the same region as A and intersecting A. Thus, at most $O(s)$ edges and trapezoids of $V^*(R \cup \{t\})$ can have A as a predecessor in the history graph. □

Lemma 25. *The set E_t can be computed in time $O(s^2c)$.*

Proof. To test if a site t interferes with a trapezoid A, we have to test if t interferes with an edge or a vertical line segment on the boundary of A. Since an edge- or vertical-line-test takes time $O(s)$ using the Basic Operation, and each trapezoid has at most 4 edges and vertical lines on its boundary, this takes time $O(s)$. If an edge-node of $\mathcal{H}(R_i)$, for minimal i, interferes with t, all of its successors are in conflict with t, at least their descriptions do not appear in $V(R_i \cup \{t\})$. Together with the fact that the outdegree of each trapezoid-node of $\mathcal{H}(R)$ is $O(s)$, the set E_t can be computed in time $O(s^2c)$. □

3.5.2 Construction of $V^*(R \cup \{t\})$ and $\mathcal{H}(R \cup \{t\})$

Now that the set E_t of all trapezoids of $V^*(R)$ interfering with t is known, it remains to compute $V^*(R \cup \{t\})$ and $\mathcal{H}(R \cup \{t\})$ from $V^*(R)$ and $\mathcal{H}(R)$. Like Section 3.4.1, this is a rather technical issue with few unexpected insights. However, it can not be avoided if we want to show the correctness of our algorithm. We follow [53] in most cases, only the new phenomena of touch points needs a special treatment.

At first, we discuss the vertices that have to be updated to construct $V(R \cup \{t\})$ from $V(R)$ and use the notation from Definition 3 in [53], which says that t clips an edge e at its endpoint v, iff $e \cap \overline{\mathcal{T}}$ contains a component incident to v. There are new vertices V_{new} that need to be added to $V(R)$, vertices V_{chang} already existing in $V(R)$ but getting a new adjacency list in $V(R \cup \{t\})$, and vertices V_{del} that have to be deleted from $V(R)$. We can define these vertices as follows and show that they indeed form a partition of all vertices of $V(R \cup \{t\})$.

- The set of vertices which remain unchanged V_{unch}, are the vertices v of $V(R)$ such that $v \notin \overline{\mathcal{T}}$.
- The set of vertices already existing in $V(R)$ but getting a new adjacency list in $V(R \cup \{t\})$, V_{chang}, are the vertices v of $V(R)$ such that $v \in \partial\mathcal{T}$ and not all edges incident to v are clipped at v by t, or v is a touch point.
- The set of vertices which have to be deleted V_{del}, are the vertices v of $V(R)$, where all edges incident to v are clipped at v by t, and v is not a touch point.
- The set of vertices which are new V_{new}, are the endpoints of $e - \overline{\mathcal{T}}$ for an edge $e \in V(R)$ intersected by $\overline{\mathcal{T}}$ (no endpoints of e, because e is open), or touch points with respect to t on e.

Lemma 26. *The set of vertices of $V(R \cup \{t\})$ is equal to $V_{unch} \cup V_{chang} \cup V_{new}$.*

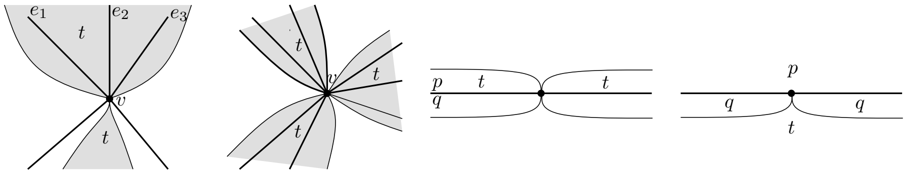
Proof. Let v be a vertex of $V(R \cup \{t\})$. If v was not yet a vertex of $V(R)$, it must lie on the boundary of $\mathcal{T}$, and on an edge e bounding a trapezoid $A \in E_t$.

If a part of e is still incident to v in $V(R \cup \{t\})$, then v is the endpoint of $e - \overline{\mathcal{T}}$. Otherwise v is a touch point with respect to t. In both cases v lies in V_{new}.

If v was already a vertex of $V(R)$, it is either not intersected by $\overline{\mathcal{T}}$, and hence, it lies in V_{unch}, or v lies on the boundary of $\mathcal{T}$. In that case, if a part of an edge e of $V(R)$ is still incident to v in $V(R \cup \{t\})$, then not all edges incident to v are clipped at v by t, otherwise v is a touch point. In both cases v lies in V_{chang}.

If a vertex v of $V(R)$ is not part of $V(R \cup \{t\})$ any more, then it lies in the interior of $\mathcal{T}$, and hence, all edges incident to v are clipped at v by t, and v lies in V_{del}. □

The vertices in V_{chang} are assigned new adjacency lists in $V(R \cup \{t\})$. Let in a clockwise order the vertex v of $V(R)$ be incident to the edges $e_1, \ldots, e_k$. To update the incident edges of v in $V(R \cup \{t\})$, all edges e_i clipped at v by t are removed. If e_i is clipped but not e_j, where $j \in \{i-1, i+1\}$, and e_i, e_j border a p-region, then an edge $\subseteq J(p,t)$ separating t and p region is inserted instead of e_i. For all adjacent faces $P \subseteq \mathrm{VR}(p,R)$, for which $J(p,t)$ intersects v and is contained in P in an ε-neighborhood around v, two edges separating p and t, and t and p region are inserted between the two edges incident to v bordering P, see Figure 3.11a,b.



(a) (b) (c) (d)

Figure 3.11: In (a) and (b) $v \in V_{chang}$, but in (a) not all edges are clipped at v by t and in (b) all edges are clipped at v by t and v is a touch point. In (c) and (d) $v \in V_{new}$ and v is of degree 4, but in (c) it borders two connected components of the new region $\mathrm{VR}(t, R \cup \{t\})$ and in (d) it borders two connected components of an old region $\mathrm{VR}(q, R \cup \{t\})$.

To handle the vertices in V_{new}, we have the following lemma, which follows directly from Lemma 14, see also Figure 3.11c and 3.11d.

Lemma 27. *Let $v \in V_{new}$ and let v lie on an edge e of $V(R)$, which is separating a p- and q-region. Then v is a vertex of degree 3 or 4 of $V(R \cup \{t\})$.*

If v is of degree 3, the adjacent edges are separating the p- and q-, q- and t- and t- and p-region.

If v is of degree 4, then v is a touch point with respect to the new site t, or an old site $q \in R$. In the first case, the adjacent edges are separating the p- and t-, t- and q-, q- and t-, and t- and p-region. In the second case, the adjacent edges separate p- and q-, q- and t-, t- and q-, and q- and p-region.

Now that we know which structural changes are necessary, we will discuss how to implement them. For each face of $\mathcal{T}$ we need to traverse its boundary through the trapezoidal decomposition of $V(R)$. Clearly, these trapezoids interfere with the new site, t, but there may be other interfered trapezoids through

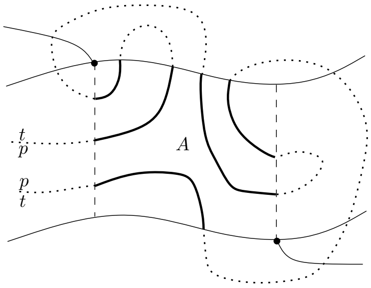


Figure 3.12: A trapezoid $A \subseteq \mathrm{VR}(p, R)$ is visited several times by the bisector $J(p, t)$. Bold segments correspond to $A \cap J(p, t)$.

which the boundary of $\mathcal{T}$ does not pass, because they are contained in the interior of a face of $\mathcal{T}$.

Starting from an arbitrary interfered trapezoid we go through the set E_t and test each trapezoid $A \in E_t$ if it is traversed by $\partial\mathcal{T}$, using the Basic Operation. If so, and if A is contained in a face of $\mathrm{VR}(p, R)$, we compute $\partial A \cap J(p, t)$, which may consist of $O(s)$ components. By our assumptions we can now in time $O(s \log s)$ sort these components according to their occurrence on $J(p, t)$ and obtain a list L_A of all segments of $\overline{A} \cap J(p, t)$ with their start- and endpoints on ∂A, see Figure 3.12.

After these preparations we have a list L of all trapezoids of $V^*(R)$ that are intersected by the boundary of the new region $\mathcal{T}$, and for each trapezoid $A \subseteq \mathrm{VR}(p, R)$ in L, the list L_A containing the segments of $\overline{A} \cap J(p, t)$. While list L is not empty we proceed in the following way. We pick some A of list L and, assuming $A \subseteq \mathrm{VR}(p, R)$, a segment z of the list L_A. Segment z bounds a face F of $\mathcal{T}$. Starting from z we trace ∂F through the trapezoids of list L, using the structure of $V^*(R)$. On the way, we update the Voronoi diagram according to our previous structural analysis. Also, we delete all segments of t-bisectors encountered in their trapezoids from the list L_A. Upon completion of the cycle ∂F at z we remove from L all trapezoids A having an empty list L_A. This way we can be sure that no face of $\mathcal{T}$ will be ignored.

Some care has to be taken, when a segment $z \in L_A$ lies on the boundary of A bordering another trapezoid B. Then z may be contained in both L_A and L_B and it must be deleted from both lists when encountered during the traversal.

After updating the Voronoi diagram we can update the trapezoidal decomposition in a natural way; compare [68].

The following lemma account for the cost of this incremental step.

Lemma 28. *If E_t is known, then $V^*(R \cup \{t\})$ and $\mathcal{H}(R \cup \{t\})$ can be constructed from $V^*(R)$ and $\mathcal{H}(R)$ in time $O(s^2|\overline{E_t}|)$.*

Proof. See the discussion above. To test all interfered trapezoids for intersection with $\partial\mathcal{T}$ takes time $O(s|E_t|)$. From this it takes time $O(s \log s\ |E_t|)$ to compute the lists L_A for all A traversed by $\partial\mathcal{T}$. During the traversal of the boundaries of all faces F of $\mathcal{T}$ one determines the sets $V_{del}, V_{chang}, V_{new}$ and update their cyclic adjacency lists. Further, each time a new segment z from $\overline{A} \cap J(p, t)$ is reached, one has to find it in the corresponding list L_A, which can be done in time $O(s)$. (By sorting the lists in advance one may also be able to find it in time $O(\log s)$, but the computation of the set E_t already took time $O(s^2c)$ and $c \geq |E_t|$, so we would not gain anything from it.)

Finally, the trapezoids of $V^*(R \cup \{t\})$ have to be updated. All trapezoids whose closure (minus the corners) is intersected by $\overline{\mathcal{T}}$ have to be deleted and new trapezoids along the boundary of $\mathcal{T}$ have to be inserted, which compares with, e. g., [68]. Further, for all these trapezoids we have to test recursively, if the at most four adjacent trapezoids get a new description, because an edge bounding them may have been changed. This can be done in time $O(s|\overline{E_t}|)$. In the same time $\mathcal{H}(R \cup \{t\})$ can be updated. Because $|E_t| \leq |\overline{E_t}|$, altogether the whole procedure takes time $O(s^2|\overline{E_t}|)$. □

3.6 Analysis

Finally, we can analyse the running time and space consumption of the algorithm. The sets E_t for all $t \in S$ can be computed in time $O(s^2c)$, Lemma 25, and $V^*(R \cup \{t\})$ and $\mathcal{H}(R \cup \{t\})$ in time $O(s^2|\overline{E_t}|)$, Lemma 28. Further, the space consumption corresponds to the size of $\mathcal{H}(S)$.

Thus we need to upper bound the following variables:

- $c := \sum_{j=3}^{i} \#$ trapezoids of $V^*(R_i)$ not in $V^*(R_{i-1})$ in conflict with t;
- Size of $\overline{E_t} := \{A \mid A$ is a trapezoid of $V^*(R)$ in conflict with $t\} \supseteq E_t$;
- Size of $\mathcal{H}(S)$ to estimate the space of the algorithm.

Because worst cases may be very inefficient for AVD's, we randomize the insertion order of the sites, and compute an expected running time and space consumption. For this purpose we use the analysis for randomized incremental constructions by Clarkson, Mehlhorn, Seidel [29] and apply it to our history graph based on trapezoids. Unlike AVD's with connected regions, where the size of the diagram is linear in the number of sites, for disconnected regions the size may vary a lot depending on the instance. To take this dependency into account we make the following definition.

Definition 6. *Let m_j denote the average number of faces per region, over all AVD's of j sites from S.*

Now let us start with an easy but important observation.

Lemma 29. *The complexity of $V^*(R)$ is in $O(|V(R)|)$.*

Proof. In $V^*(R)$ the number of trapezoids a face F of $V(R)$ is decomposed into is upper bounded by the number of vertices plus two times the number of points of vertical tangency on the boundary of F. This implies:

$$|V^*(R)| \leq \sum_{F \text{ face of } \mathrm{V}(R)} |F| + 2 \text{ \#points of vertical tangency of } V(R)$$

Since at most $O(|V(R)|)$ many bisectors contribute to $\mathrm{V}(R)$, and the number of points of vertical tangency is constant for each bisector, this sum is $O(|V(R)|)$. □

Now we can start with the run time analysis. Remember that the insertion order of the sites is randomized.

Lemma 30. *For $R = \{r_1, \ldots, r_i\}$, the expectation of c is $O(\sum_{j=3}^{i} \frac{m_j}{j})$.*

Proof. Let A be a trapezoid-node of the history graph $\mathcal{H}(r_1, \ldots, r_i)$, where the sites $r_1, \ldots, r_i$ are inserted in this order. Now let $t \in S \setminus R$ randomly chosen. If A is in conflict with t, then A with its description $D_R(A)$ would never have been constructed, if t would have been inserted as the first site, i. e., $A \notin \mathcal{H}(t, r_1, \ldots, r_i)$. By elementary set calculus, it follows:

$$\begin{aligned} &|\mathcal{H}(r_1, \ldots, r_i) \setminus \mathcal{H}(t, r_1, \ldots, r_i)| \\ =&|\mathcal{H}(r_1, \ldots, r_i)| - |\mathcal{H}(t, r_1, \ldots, r_i)| + |\mathcal{H}(t, r_1, \ldots, r_i) \setminus \mathcal{H}(r_1, \ldots, r_i)|. \end{aligned}$$

The expectation of $|\mathcal{H}(r_1, \ldots, r_i)| - |\mathcal{H}(t, r_1, \ldots, r_i)|$ is ≤ 0, because $|\{r_1, \ldots, r_i\}| < |\{t, r_1, \ldots, r_i\}|$ implies for randomized inputs the expectation $|\mathcal{H}(r_1, \ldots, r_i)| \leq |\mathcal{H}(t, r_1, \ldots, r_i)|$. Thus the equation above has an expected value

$$\leq E(\underbrace{|\mathcal{H}(t, r_1, \ldots, r_i) \setminus \mathcal{H}(r_1, \ldots, r_i)|}_{=:X}).$$

If $A \in |\mathcal{H}(t, r_1, \ldots, r_i) \setminus \mathcal{H}(r_1, \ldots, r_i)|$, then $t \in \mathrm{set}(D_{R \cup \{t\}}(A))$. Let j be minimal with $A \in V^*(t, r_1, \ldots, r_j)$ and thus $r_j \in \mathrm{set}(D_{R \cup \{t\}}(A))$.
Then the size of X is upper bounded by

$$\leq \sum_{j=2}^{i} \underbrace{|\{A \in V^*(t, r_1, \ldots, r_j) | t, r_j \in \mathrm{set}(D_{R_j \cup \{t\}}(A))\}|}_{=:Y},$$

where $R_j = \{r_1, \ldots, r_j\}$. By choosing t and r_j randomly from $\{t, r_1, \ldots, r_j\}$ we can estimate the expectation of Y by

$$\leq \frac{1}{(j+1)j} \sum_{(x,y) \in \{t, r_1, \ldots, r_j\}^2, x \neq y} |\{A \in V^*(t, r_1, \ldots, r_j) | x, y \in \mathrm{set}(D_{R_j \cup \{t\}}(A))\}|$$

and since $\mathrm{set}(D_R(A))$ consists of at most 16 different sites it follows

$$\leq \frac{1}{(j+1)j} \cdot 16 \cdot 15 \cdot \underbrace{|V^*(t, r_1, \ldots, r_j)|}_{\leq C|\mathrm{V}(t, r_1, \ldots, r_j)| \leq C m_{j+1}(j+1)} \leq C \frac{m_{j+1}}{j}$$

for a constant C. Thus the expectation of X is

$$\leq \sum_{j=3}^{i} C \frac{m_{j+1}}{j} \in O\left(\sum_{j=3}^{i} \frac{m_j}{j}\right).$$

This implies $c \in O(\sum_{j=3}^{i} \frac{m_j}{j})$. □

Lemma 31. *The expected size of the history graph $\mathcal{H}(S)$ is $O(\sum_{j=4}^{n} m_j)$.*

Proof. Let $R_j = \{r_1, \ldots, r_j\}$ be the set of the first j sites inserted in this order. The size of $\mathcal{H}(S)$ equals $\sum_{j=4}^{n} \mathcal{H}(R_j) \setminus \mathcal{H}(R_{j-1})$. If $x \in \mathcal{H}(R_j) \setminus \mathcal{H}(R_{j-1})$, then by the definition of the history graph, either x is an edge on the boundary of

$\mathrm{VR}(r_j, R_j)$ or x is a trapezoid of $V^*(R_j) \setminus V^*(R_{j-1})$. In both cases $r_j \in D_{R_j}(x)$. This implies:

$$\begin{aligned}
&E(\mathcal{H}(R_j) \setminus \mathcal{H}(R_{j-1})) \\
=&\frac{1}{j} \sum_{r \in R_j} \sum_{e \in \mathrm{edges}(V(R_j))} \delta(e, r) + \frac{1}{j} \sum_{r \in R_j} \sum_{A \in \mathrm{trapezoids}(V^*(R_j))} \delta(A, r), \\
&\text{where } \delta(x, r) = \begin{cases} 1, & \text{if } r \in \mathrm{set}(D_{R_j}(x)) \\ 0, & \text{else.} \end{cases} \\
\leq&\frac{1}{j}(6|V(R_j)| + 16|V^*(R_j)|) \in O(m_j).
\end{aligned}$$

Summing up over all j from $j = 4$ to n, we get $E(|\mathcal{H}(S)|) \in O(\sum_{j=4}^{n} m_j)$. □

Lemma 32. *The expected size of $\overline{E_t}$ is $O(m_j)$.*

Proof. The set $\overline{E_t}$ equals the set of all trapezoids of $V^*(R) \setminus V^*(R \cup \{t\})$. With the same arguments as in the two previous proofs one can show that the expected size of this set is $O(m_j)$. □

Finally, we obtain our main theorem.

Theorem 6. *$V(S)$ can be computed in expected time $O(s^2 n \sum_{j=3}^{n} \frac{m_j}{j})$ and expected space $O(\sum_{j=4}^{n} m_j)$.*

Proof. The sets E_t for all $t \in S$ can be computed in expected time $O(s^2 n \sum_{j=3}^{n} \frac{m_j}{j})$, see Lemma 25 and 30. $\mathcal{H}(S)$ and $V^*(S)$ can be computed in time

$$O(s^2 \sum_{t \in S} |\overline{E_t}|)$$

and because the expectation of $|\overline{E_t}|$ is $O(m_j)$, the expected value is in

$$O(s^2 \sum_{j=3}^{n} m_j) \in O(s^2 n \sum_{j=3}^{n} \frac{m_j}{j}).$$

Lemma 31 shows the expected size of $\mathcal{H}(S)$. □

3.7 Discussion

One question is how m_j affects the performance of our algorithm. For Voronoi diagrams, where $m_j = j$, $|V(S)| \in O(n^2)$, and s is constant, the algorithm is output sensitive.

If though there is one more site p, such that $V(S) \subset D(p, q)$ for all $q \in S$, then the expected running time is still the same, but the size of $V(S \cup \{p\})$ is constant. For such examples randomization does not help.

Also, axiom (A5) and our general position assumption are crucial in this chapter. But how to deal with applications where these fail to hold?

Another still open problem is to find a tight upper bound for the size of $V(S)$. Because of the transitivity, Lemma 12, we can define a total order on

S for all $x \in \mathbb{R}^2$ by $p \leq_x q$ iff $x \in D(p,q)$. This leads to the conjecture that the complexity of $V(S)$ equals the complexity of the lower envelope of surfaces. If each pair of bisectors $J(p,q)$ and $J(r,t)$ intersect in at most s components, and s is constant, this is $O(n^{2+\varepsilon})$, [68]. But axiom (A2') implies the at most s intersections only for bisectors sharing a site, whereas other bisectors can intersect in any finite number of components.

Chapter 4

Closed Bisectors

In the previous chapter we showed how to handle abstract Voronoi diagrams with disconnected regions, but with unbounded bisecting curves. Unbounded bisectors were strongly required to be able to locate the trapezoids of the current diagram intersected by the new region to be inserted. Otherwise the new region could be an island completely contained in the interior of a single region of the current diagram, without intersecting any edges, and the history graph from the previous chapter would fail to detect the intersected trapezoids.

In this chapter we overcome this problem and allow bisectors to be closed Jordan curves, a phenomenon occurring in many applications. We show some new facts about this wider type of abstract Voronoi diagrams and present the first algorithm for constructing them. It runs in expected

$$O\big(s^2 n \log(\max\{s,n\}) \sum_{j=2}^{n} m_j/j\big)$$

many steps and $O(\sum_{j=3}^{n} m_j)$ space, where m_j denotes the average number of faces per Voronoi region in any diagram of a subset of j sites and s is the maximum number of intersections between any two related bisectors $J(p,q)$ and $J(p,r)$.

4.1 Introduction

So far, only abstract Voronoi diagrams with unbounded bisecting curves were studied, leaving many interesting Voronoi diagrams uncovered. In this chapter we consider, for the first time, abstract Voronoi diagrams built from arbitrary Jordan curves, unbounded or closed.

The difficulty with closed bisecting curves is that Voronoi regions are no longer simply-connected. That is, the Voronoi region of an abstract site p can be an island inside the region of q if $J(p,q)$ is a closed curve. But how to detect islands, in constructing Voronoi diagrams, is not obvious. For example, the classical divide&conquer algorithm by Shamos and Hoey [67] relies heavily on the fact that endpoints of the polygonal chain bisecting two sub-diagrams can be picked up at infinity; compare [48]. Also, the randomized incremental construction method by Clarkson and Shor [30], as applied to AVDs in [53, 49], makes use of the fact that a Voronoi region to be newly inserted must intersect an edge of the Voronoi diagram so far constructed, and that such an edge can be found efficiently by means of a suitable history graph. None of these facts holds in the presence of islands. It seems unavoidable to search the existing Voronoi regions for the occurrence of an island. The difficulty is in doing this efficiently.

Throughout this chapter we require the following properties. Bisecting curves intersect vertical lines only constantly often. Any two bisecting curves intersect finitely often, and at most s times if both bisectors are related to the same site. The closures of the Voronoi regions of any Voronoi diagram of three sites cover the plane. Under these assumptions we can construct the abstract Voronoi diagram in expected

$$O\Big(s^2 n \log(\max\{s,n\}) \sum_{j=2}^{n} \frac{m_j}{j}\Big) \tag{4.1}$$

many steps and in $O(\sum_{j=3}^{n} m_j)$ space, where m_j denotes the average number of faces per Voronoi region in any diagram of a subset of j sites. Let us look at some applications.

If the bisecting curves are pseudo-circles, Voronoi regions are of linear complexity, as we will see in Section 4.5. Thus, $m_j \leq j$, and (4.1) yields the first $O(n^2 \log n)$ randomized algorithm for this class of Voronoi diagrams, which is nearly optimal since their size can be quadratic.

An important special case are points with multiplicative weights; here bisectors are lines or Euclidean circles. The deterministic algorithm by Aurenhammer and Edelsbrunner [12] runs in time $\Theta(n^2)$ which is optimal in the worst-case where the diagram is of quadratic size. In this case, our algorithm takes a $\log n$ factor longer. But if, e.g., points are fixed, and weights drawn independently from some distribution (or just permuted at random) then the multiplicatively weighted Voronoi diagram has expected size only $O(n \log^2 n)$, as was recently shown by Har-Peled and Raichel [44], and it can be constructed in this time. Our algorithm is sensitive to the smaller output size: Since $m_j \in O(\log^2 j)$ holds in this case, it runs in expected time $O(n \log^4 n)$.

For the farthest Voronoi diagram of n simple polygons of constant complexity, formula (4.1) yields a randomized $O(n \log^2 n)$ algorithm. Indeed, s is constant, and the Voronoi diagram is of linear size, as was shown by Cheong et al. [24]; thus, the average number of faces per Voronoi region is constant. In comparison, the algorithm in [24] runs on polygons of total complexity n in deterministic time $O(n \log^3 n)$.

Our algorithm for abstract Voronoi diagrams is fairly general. It uses a randomized incremental construction [30, 29] based on a new history graph and search procedure that is quite different from the one used in Chapter 3, and [53, 49], and [17]. Furthermore, it uses an application of Chazelle's linear triangulation algorithm [23] to compute the trapezoidal decomposition of a given face. At a later stage this may be replaced by a simpler randomized method, e. g., by Amato et. al. [9], which though would require a new randomized analysis.

We will discuss the new history graph and search procedure in Section 4.3, and upper bound the expected number of conflicts, and the effort invested in finding them in Section 4.4.

4.2 Preliminaries

For completeness let us explicitly name the axioms used in this chapter. They are claimed to hold for each subset S' of S.

(A1') By stereographic projection to the sphere, each curve $J(p, q)$ is a closed Jordan curve on the sphere.

Curve $J(p, q)$ is unbounded iff its projection runs (in the limit) through the north pole of the sphere; otherwise, $J(p, q)$ is a closed Jordan curve in the plane. One of the dominance regions $D(p, q)$ or $D(q, p)$ may now be bounded by $J(p, q)$.

Compared to the previous chapter we will use a different version of axiom (A2'). Lemmata 15 and 16 from Section 3.3 show that they are equivalent in the sense that if any two related bisectors intersect in $O(s)$ components, then any Voronoi region in a diagram of a subset of 3 sites consists of $O(s)$ many faces.

(A2") The intersection of any two bisecting curves $J(p, q)$ and $J(p, r)$ related to the same site p has at most s connected components.

(A3) Each point of the plane belongs to the closure of a Voronoi region $\overline{VR(p, S')}$.

(A4) For any two curves $J(p, q)$ and $J(r, t)$, their intersection has only finitely many connected components.

Because of (A3) the useful Transitivity Lemma [49] is still valid.

Lemma 33. *For any p, q, r in S we have $D(p, q) \cap D(q, r) \subseteq D(p, r)$.*

Like in the previous chapter property (A4) allows us to zoom in on any point v situated on one or more bisecting curves, and find a neighborhood U of v where the curves passing through v either stay disjoint or coincide, on either side of v. As a consequence one obtains a "piece-of-pie" Lemma, compare Lemma 14 from Section 3.2, stating that each point v of $V(S)$ is either an interior point of a Voronoi edge, if v lies on the common boundary of exactly two Voronoi regions of different sites, or neighborhood U contains pieces of Voronoi regions of three different sites or more, making v a Voronoi vertex.

An example of an abstract Voronoi diagram based on (A1'), (A2"), (A3), and (A4) is shown in Figure 4.1. The main differences to the abstract Voronoi diagrams introduced so far are as follows. There can be islands inside a Voronoi region. The boundary of an island need not contain a Voronoi vertex and it equals a closed bisector $J(p, q)$. In the neighborhood of a Voronoi vertex, the same face of a Voronoi region can be represented several times.

Property (A2") implies that, in a Voronoi diagram of three sites, the number of vertices and faces of a Voronoi region is in $O(s)$. For the overall complexity of the Voronoi diagram, one obtains from (A2") the following, compare Lemmata 18 and 19 from Section 3.3.

Lemma 34. *A single Voronoi region is of complexity $O(s\,n^2)$, and this bound can be attained. The whole Voronoi diagram $V(S)$ is of size $O(s\,n^3)$ [17].*

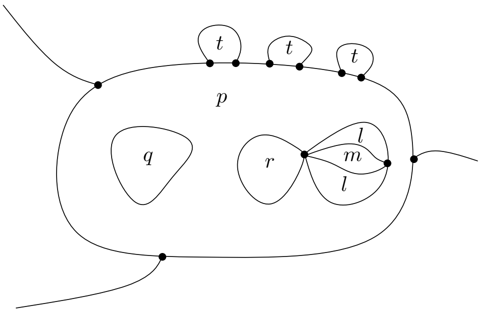


Figure 4.1: An abstract Voronoi diagram based on axioms (A1'), (A2"), and (A3).

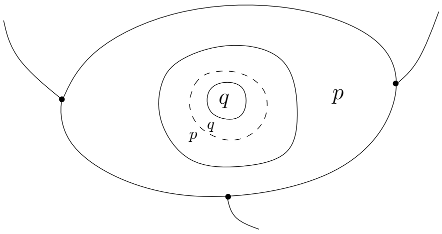


Figure 4.2: The Voronoi region of q is confined to the island by the dashed bisector $J(q, p)$.

The proof is by vertex counting. We observe that islands can be nested, but their total number is in $O(n)$. Namely, if an island is situated in a face of the Voronoi region of p, and contains points of the region of some site $q \neq p$, then the whole Voronoi region of q is confined to this island by the bisector $J(q, p)$; see Figure 4.2.

Since we are going to subdivide Voronoi regions into trapezoids [66] we require one last property which is quite common in arrangement theory [68].

(A5) Each bisector has constantly many points of vertical tangency. All such points have pairwise different x-coordinates.

Property (A5) implies that each bisecting curve has at most a constant number of points of vertical tangency. The trapezoidal decomposition $V^*(R)$, for a subset R of S, results from shooting vertical rays from all vertices and points of vertical tangency in both directions, until they hit another point of $V(R)$; see Figure 4.3b. Note that $V^*(R)$ is still of the same size as $V(R)$.

Voronoi edges and trapezoids of $V^*(R)$ are referred to by unique *descriptions.* A general edge description is shown in the leftmost drawing of Figure 4.4. Associated with vertex v is a 5-tuple containing the names of the two sites p, q separated by edge e, the names of the sites of the adjacent regions, r_q and r_p, and a tuple containing the coordinates of v. The sites are listed in clockwise order around v. A similar 5-tuple is associated with w.

Figure 4.5 depicts a general trapezoid. Its description consists of the descriptions of the upper and lower Voronoi edges, and of the vertical segments bounding the trapezoid on either side.

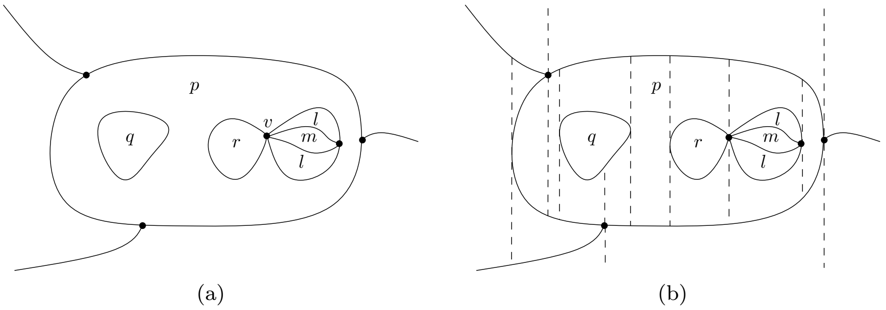


Figure 4.3: (a) An example of a Voronoi diagram with closed bisectors. (b) The trapezoidal decomposition of the same diagram.

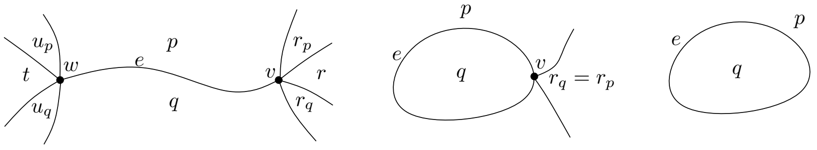


Figure 4.4: Edges e with descriptions $D_R(e) = \{(r_q, q, p, r_p, \text{coor}(v)), (u_p, p, q, u_q, \text{coor(w)})\}$, $D_R(e) = \{(r_q, p, q, r_p), \text{coor}(v)\}$ and $D_R(e) = \{(p, q)\}$.

Similar like in Chapter 3, as a *basic operation* we assume that the intersections of two related bisectors can be computed in time $O(s)$, two related bisectors can intersect in at most $O(s)$ components, see (A2"). The intersection between a Voronoi edge of $V(R)$ and the region of a new site $\text{VR}(t, R \cup \{t\})$ for $t \in S \setminus R$ can also be computed in time $O(s)$, because this can be decided in the diagram of 7 sites, the site t and the sites contained in the description of e, and e can be intersected in at most $O(s)$ components. The intersection of a vertical line and a bisector can be computed in constant time, there can be at most a constant number of intersection points, see (A5), and hence, the intersection of a trapezoid and a new region can be computed in time $O(s)$. Furthermore, we assume that for a bisector $J(p, q)$ and a point x in the plane, we can in constant time compute the points of vertical tangency of $J(p, q)$, decide if x is on

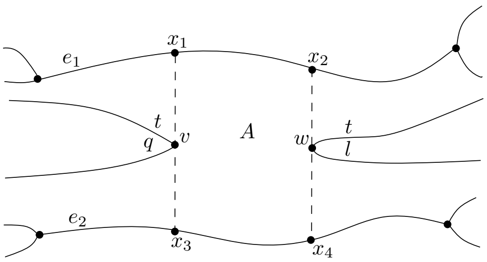


Figure 4.5: The description of trapezoid A contained in the Voronoi region of t is given by the descriptions of its upper and lower Voronoi edges, e_1 and e_2, the sites q and l whose bisectors $J(q, t)$ and $J(l, t)$ determine the tangency points v and w, and the coordinates of x_1 to x_4.

$J(p,q)$, or in $D(p,q)$, or $D(q,p)$. Also, for two points x and y on $J(p,q)$, given a direction of $J(p,q)$, we can decide in constant time if x or y comes first.

4.3 Searching for Intersections

For the algorithm we restrict our attention to the finite part of the Voronoi diagram and assume that a large closed curve Γ is given, encircling the diagram such that it encloses all closed bisectors, it intersects each unbounded bisecting curve exactly twice, and outside of Γ each pair of bisecting curves is either equal or disjunct.

Our algorithm is a randomized incremental construction, where the insertion order $\{r_1,\dots,r_n\} = S$ of the sites is given randomized, and we insert the Voronoi regions successively to the current Voronoi diagram. Let $R = \{r_1,\dots,r_i\}$, $V^*(R)$ the trapezoidal decomposition of the current diagram $V(R)$, and $t \in S \setminus R$ the next site to be inserted. In order to incrementally construct the augmented Voronoi diagram $V^*(R \cup \{t\})$ from $V^*(R)$ we need to determine which parts of $V^*(R)$—edges or trapezoids—are intersected by the Voronoi region $\mathrm{VR}(t, R \cup \{t\})$ of the new site t (if this region is a small island it could well be contained in the interior of a single trapezoid of $V^*(R)$, without intersecting any edges).

We define edges, trapezoids, and Voronoi regions to be open, i.e., an edge does not contain its endpoints, and trapezoids and regions do not contain their boundaries. Thus, all intersections are "proper", in that they contain a subset of full dimension. If r were the last site inserted to the Voronoi diagram shown in Figure 4.3a, its region would not intersect the previous Voronoi diagram, but one of its trapezoids, and Voronoi vertex v would be found at a later stage, when tracing the boundary of the Voronoi region of r through the trapezoidal decomposition.

Definition 7. *We say that a trapezoid A or an edge e of $V^*(R)$ is*

(i) intersected *by t, if $VR(t, R \cup \{t\})$ intersects A or e respectively;*

(ii) in conflict *with t, if A or e with their descriptions $D_R(A)$ and $D_R(e)$ do not appear in $V^*(R \cup \{t\})$.*

Observe that if a trapezoid or an edge is intersected by t, then it is also in conflict with t but not necessarily vice versa, see Figure 4.6. Recall also the very important fact that if $R' \subseteq R$, then $\mathrm{VR}(t, R \cup \{t\}) \subseteq \mathrm{VR}(t, R' \cup \{t\})$, which follows quite directly from the definition of Voronoi regions.

Finding all proper intersections of $\mathrm{VR}(t, R \cup \{t\})$ with $V^*(R)$ is facilitated by a *history graph* $\mathcal{H}(R)$, a DAG with a single source, a node for each face of the inserted regions, and a node for each trapezoid ever constructed during the incremental process. More precisely, let $R = R_i = \{r_1,\dots,r_i\}$ be the insertion order, then the node set of $\mathcal{H}(R)$ equals

$$\begin{aligned}\{\text{source}\} \cup \{D_{R_j}(A) \mid A \text{ is a trapezoid of } V^*(R_j) \text{ for } j \in \{2,\dots,i\}\} \\ \cup \ \{F \mid F \text{ is a face of } \mathrm{VR}(r_j, R_j) \text{ for } j \in \{3,\dots,i\}\}.\end{aligned}$$

Note that a trapezoid that exists in a sequence of Voronoi diagrams $V^*(R_j)$ is represented in $\mathcal{H}(R)$ by a single node. Faces are represented by a node only

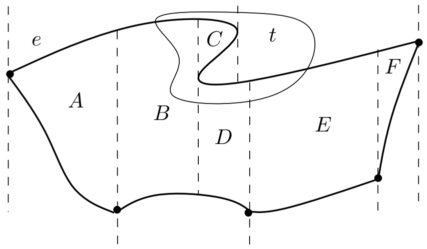


Figure 4.6: All trapezoids A to F are in conflict with t, because t intersects edge e, but only the trapezoids B to E are intersected by t.

when contained in a new region, thus, in contradistinction to trapezoids, not all faces ever constructed are represented by a node in $\mathcal{H}(R)$. Let a node of $\mathcal{H}(R)$ be called *face-node* if it refers to a face and *trapezoid-node* if it refers to a trapezoid. The edges of $\mathcal{H}(R)$ are defined incrementally, maintaining the invariant that the leaves of $\mathcal{H}(R)$ correspond to the trapezoids of $V^*(R)$. Also each trapezoid and face of the current diagram $V^*(R)$ is linked to its corresponding trapezoid- and face-node of $\mathcal{H}(R)$ and vice versa.

The top of the history graph is an artificial source and its children are the trapezoids of the diagram of the first two sites r_1 and r_2 of the insertion order. Now let $t \in S \setminus R$ be the next site to be inserted. The history graph $\mathcal{H}(R \cup \{t\})$ is obtained by updating $\mathcal{H}(R)$ as follows.

If F is a face of $\mathrm{VR}(t, R \cup \{t\})$, then F is made a successor of all trapezoids A' of $V^*(R)$ (these trapezoids are leaves of $\mathcal{H}(R)$) intersected by F, compare face F in Figure 4.7.

If A is a trapezoid of $V^*(R \cup \{t\})$, which was not yet part of $V^*(R)$, then there are two cases:

1. Trapezoid A is contained in the region of an old site $r \in R$. Then A is made a successor of all trapezoids A' of $V^*(R)$ intersected by A. Compare trapezoids $B_1, D_1, E_1, H_1, H_2, H_3, I_1, J_1$ in Figure 4.7.

2. Trapezoid A is contained in a face F of $\mathrm{VR}(t, R \cup \{t\})$. Then A is made a successor of F, compare trapezoids K, L, M, N, O, P in Figure 4.7.

In addition to the history graph which contains only the descriptions of the trapezoids ever constructed and the "names" of the new faces, for each face-node F of $\mathcal{H}(R)$ we store its geometric structure. More specifically, let F be a face of $\mathrm{VR}(t, R \cup \{t\})$. We store its trapezoidal decomposition F^* together with a *search-structure* $\mathrm{SS}(F)$ that allows us to perform fast *point location* for a given query point x of the plane. We link the face-node F of $\mathcal{H}(R)$ to its geometric structures F^* and $\mathrm{SS}(F)$, and the trapezoids of F^* and $\mathrm{SS}(F)$ to their corresponding trapezoid-nodes, which are successors of F. Furthermore, for each trapezoid A of $V^*(R \cup \{t\})$ not contained in F but bordering from the outside on an edge e of the boundary of F (this trapezoid must have been constructed while inserting F to $V^*(R)$), we link the trapezoid-node A to the edge e in F^*, and vice versa, compare trapezoids $E_1, H_1, H_2, H_3, I_1, J_1$ in Figure 4.7. Note that trapezoid A may border on up to two different edges of the boundary of F, compare Figure 4.8. If this is the case, we link A to both of them.

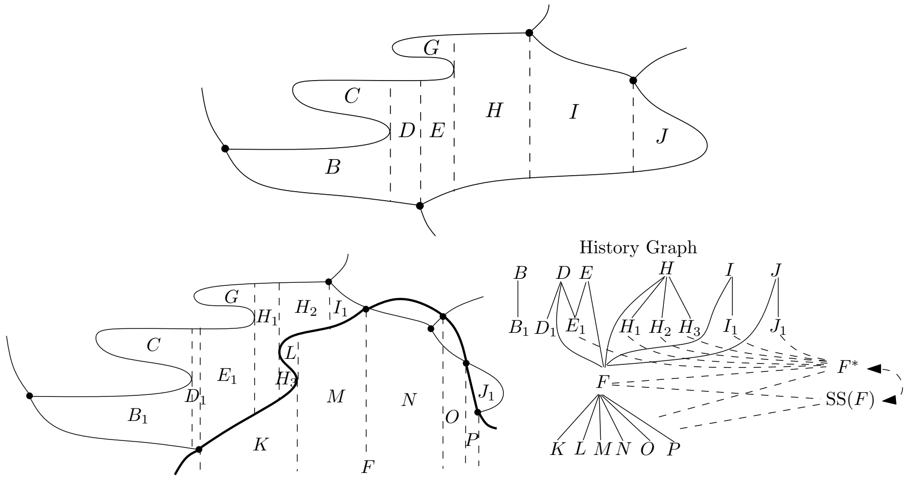


Figure 4.7: Face $F_t \subseteq \text{VR}(t, R \cup \{t\})$ is inserted in the trapezoidal decomposition of the upper diagram and the history graph is updated.

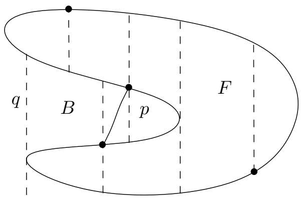


Figure 4.8: Trapezoid B borders on two different edges of F.

Now let us discuss how to compute the geometric structures of F from scratch. Each edge on the boundary of F can be split up into a constant number of x-monotone arcs, by putting an additional vertex on each point of vertical tangency (A5). By doing this, in time $O(|F|)$, we obtain a simple polygon with monotone curved edges whose trapezoidal decomposition can be computed in time $O(|F|)$, see [23]. Observe that we consider F without the rest of the diagram and we compute the trapezoidal decomposition both in the inside and outside of F. The trapezoids in the outside are needed to obtain a monotone subdivision F^* of the whole plane. For this subdivision we can now build a *search-structure* SS(F), based on the structure for monotone subdivisions introduced by Edelsbrunner et. al. in [38], which again takes time and storage $O(|F|)$, and it allows us to perform *point location* for a given query point x of the plane in time $O(\log |F|)$.

Both in [23] and [38] the edges are straight lines, but the same ideas can be adapted to instances where edges are monotone curves, and the intersection between an edge and a vertical line can be computed in constant time.

Next, we describe how to find all trapezoids intersected by the new site t. We walk through the history graph along its trapezoid-nodes intersected by the region of t as follows, starting with the successors of the source. If a trapezoid-node A of $V^*(R_j)$ is intersected by $\text{VR}(t, R_j \cup \{t\})$, then we test recursively all

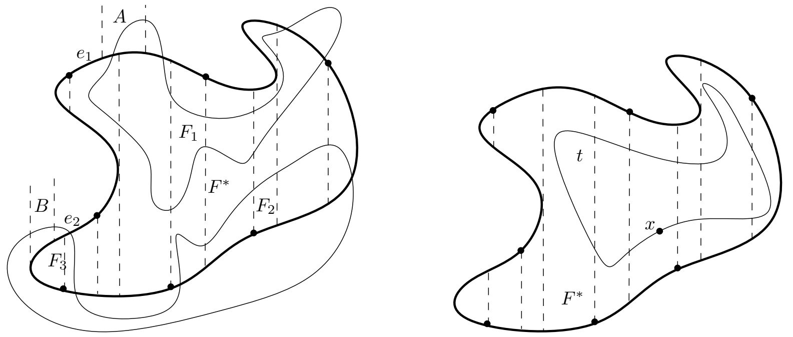


Figure 4.9: The region of t (faces with normal boundary) is traced through the trapezoidal decomposition F^* (face with fat boundary), in the left figure from two starting points – one from A trough e_1, and one from B through e_2, and in the right figure from a point x found using $\mathrm{SS}(F)$.

its succeeding trapezoid-nodes for intersection with t; for the succeeding face-nodes we discuss later how to proceed. First, if A is additionally linked to an edge e of the trapezoidal decomposition F^* of a face-node F (by construction this means that A borders on e from the outside of F), we test if e is intersected by t and if yes, we test all trapezoids bordering on e from the inside of F for intersection with t; these trapezoids are found in F^*. Afterwards we set a flag on e saying that e already has been tested for intersection with t, thus if e is reached from another trapezoid, we do not test it and its adjacent trapezoid in F^* for intersection with t again. For the running time analysis later on we note already now that if edge e is intersected by t, then all trapezoids bordering on e (there may be many!) are in conflict with t.

For each trapezoid bordering on e from the inside of F, having been successfully tested for intersection with t, we recursively test also its at most 4 neighbors in F for intersection with t. Doing this we trace the region of t through F. Of course the region of t may intersect F and other faces of the same region in several disconnected components, but we will later see that we will find a starting point for tracing each component. Again we set a flag with respect to t on each tested trapezoid to prevent it from having to be tested again.

In the left drawing of Figure 4.9, the region of t intersects F^* in 3 components, F_1, F_2, and F_3. If we enter F^* from A through e_1, then all trapezoids of F^* intersected by F_1 and F_2 are found. To detect the trapezoids intersected by F_3, we need another starting point, e. g., from B through e_2.

We may now have detected additional trapezoids in the interior of F that are intersected by t, they are linked to their corresponding trapezoid (succeeding F) in the history graph and we recursively test also their successors for intersection with t.

For a face-node $F_{r_j} \subseteq \mathrm{VR}(r_j, R_j)$ reached in the history graph, we choose an arbitrary point x of the bisector $J(t, r_j)$ and use $\mathrm{SS}(F_{r_j})$ to determine if x is contained in F_{r_j} and if yes, in which trapezoid. If no trapezoid of F_{r_j} containing x is detected, i. e., x lies outside of F_{r_j}, then nothing has to be done, as will become clear later. Otherwise a trapezoid $A \subseteq F_{r_j}$ is detected with $x \in \overline{A}$. Then we know that A is intersected by t, and like before we can trace the region of t through F_{r_j} and mark the tested trapezoids with flags with respect to t (unless they already have a flag with respect to t, then we are done), compare the right drawing of Figure 4.9. Again we may have detected additional trapezoids intersected by t and we recursively test also their successors in the history graph for intersection.

Lemma 35. *By walking through the history graph $\mathcal{H}(R)$ as described above, we reach all trapezoids of $V^*(R)$ that are intersected by $VR(t, R \cup \{t\})$.*

Proof. We prove that if a trapezoid A of $\mathcal{H}(R)$ is intersected by t, then either a predecessor of A in $\mathcal{H}(R)$ is intersected by t, or A is detected in the search structure $\mathrm{SS}(F)$ of a face F, or found while tracing the region of t through a face F. In all these cases, we will detect A in the history graph.

So, assume that A is a trapezoid which was constructed during the insertion of a site $r_j \in R$, i. e., A is in $V^*(R_j)$ but not in $V^*(R_j \setminus \{r_j\})$. Now there are two cases.

Case 1: $A \subseteq \mathrm{VR}(r, R_j)$ and $r \neq r_j$.

Let $A_1, \ldots, A_k$ be the predecessors of A in $\mathcal{H}(R_j)$. Then $A \subseteq \bigcup_{j\in\{1,\ldots,k\}} \overline{A_j}$. Thus if A is intersected by t, then also at least one of its predecessors must be intersected by t.

Case 2: $A \subseteq \mathrm{VR}(r_j, R_j)$.

Let F be the face of $\mathrm{VR}(r_j, R_j)$ containing A.

If the region of t is not completely contained in F, then no face of the region of t may be an island in F. Thus, there is a corridor contained in the region of t starting in A , crossing an edge e on the boundary of F and reaching into an adjacent trapezoid B. The trapezoid B must have been constructed during the insertion of r_j, thus, the intersection of t and B is detected like in Case 1. Because the boundaries of B and F share a common edge e which is intersected by t, the region of t is traced through F^* and A is detected.

Now let the region of t be completely contained in F. Lets first observe that F is a face-node in $\mathcal{H}(R_j)$ and if $A_1, \ldots, A_k$ are its predecessors, then F is contained in $\bigcup_{j\in\{1,\ldots,k\}} \overline{A_j}$. Thus, at least one of the predecessors of F is intersected by t and we reach F together with its search structure $\mathrm{SS}(F)$ while walking through $\mathcal{H}(R)$. Furthermore, because the region of t is contained in F the whole bisector $J(r_j, t)$ must be contained in F, thus, for any point x on the bisector we will detect a trapezoid B contained in F, which is intersected by t. From there we start to trace the boundary of the region of t and reach A. □

Let E_t be the set of all trapezoids of $V^*(R)$ intersected by t. Once this set has been computed, we can update $V^*(R)$ like described in Section 3.5.2 and $\mathcal{H}(R)$ like described above. This takes time $O(s^2|E_t|)$, see Section 3.5.2 and [17].

4.4 Analysis

To organize the discussion of the running time of our algorithm we define the following variables. Let $R = R_i = \{r_1, \ldots, r_i\} \subset S$, the sites are inserted in this order, and $t \in S \setminus R$ is the next site to be inserted. Furthermore,

$$E_t := \{A \mid A \text{ is a trapezoid of } V^*(R) \text{ intersected by } t\},$$

and

$$c := \sum_{j=2}^{i} \#\text{ trapezoids of } V^*(R_j) \text{ not in } V^*(R_{j-1}) \text{ in conflict with } t.$$

We start our discussion with the following lemma.

Lemma 36. *The set E_t can be computed in time $O(s^2 c \log(\max\{s,n\}))$.*

Proof. To test if a trapezoid A or edge e is intersected by t takes time $O(s)$, using our basic operations. Each trapezoid-node of $\mathcal{H}(R)$ has an outdegree of $O(s)$, because each trapezoid can be intersected in at most $O(s)$ components by a new Voronoi region. Thus, for each intersected trapezoid-node at most $O(s)$ succeeding trapezoids are unsuccessfully tested for intersection with t. This takes time $O(s^2)$ per intersected trapezoid-node.

Further, each trapezoid-node is linked to at most two edges e of the trapezoidal decomposition F^* for a face-node F. To test them for intersection with t or to detect that they already have a flag with respect to t takes time $O(s)$ per intersected trapezoid-node.

If such an edge e has been successfully tested for intersection with t, we have to test all trapezoids bordering on e from inside of F for intersection with t. There may be many such trapezoids, but fortunately, if e is intersected by t, then all trapezoids bordering on e are in conflict with t and thus their number can be upper bounded by c. Within the face F, each trapezoid has at most 4 neighbors, thus, while continuing the tracing within the face F, for each intersected trapezoid at most four adjacent trapezoids may be tested unsuccessfully for intersection with t. The flags prevent us from testing edges and trapezoids several times for intersection with t. Thus altogether this process takes time $O(sc)$.

For each face-node F of $\mathcal{H}(R)$ reached, we have to perform a point location searching in $\mathrm{SS}(F)$ which takes time $O(\log(F))$. A very rough (but sufficient for O-notations) estimation of $|F|$ is $O(sn^2)$, which gives us $O(\log(F)) \in O(\log(\max\{s,n\}))$. If the searching was successful, we trace the region of t and again for each intersected trapezoid we test at most 4 additional trapezoids unsuccessfully.

Each trapezoid intersected by t is also in conflict with t, thus, the overall running time to compute E_t is in $O(s^2 c \log(\max\{s,n\}))$. □

Similar to the previous chapter, we have a randomized analysis which uses the techniques by Clarkson, Mehlhorn, and Seidel [29]. The proofs are quite similar to the ones in Section 3.6.

To upper bound the number of conflicts during the insertion process in the algorithm, we define m_j as the average number of faces per region, over all diagrams of j sites from S. Furthermore, with $\mathrm{set}(D_R(A))$ we denote the set of all sites contributing to the description $D_R(A)$ of a trapezoid A, i. e., if $r \in \mathrm{set}(D_R(A))$, then either $r \in D_R(A)$, or $r \in D_R(e)$ for an edge e with $D_R(e) \in D_R(A)$.

Lemma 37. *For $R = \{r_1, \ldots, r_i\}$, the expectation of c is $O(\sum_{j=2}^{i} \frac{m_j}{j})$.*

Proof. See the proof of Lemma 30 in Section 3.6. □

Lemma 38. *The expected size of the history graph $\mathcal{H}(S)$ is $O(\sum_{j=3}^{n} m_j)$.*

Proof. Let $R_i = \{r_1, \ldots, r_i\}$ be the set of the first i sites inserted in this order. The size of $\mathcal{H}(S)$ equals $\sum_{j=3}^{n} \mathcal{H}(R_j) \setminus \mathcal{H}(R_{j-1})$. If $x \in \mathcal{H}(R_j) \setminus \mathcal{H}(R_{j-1})$, then by the definition of the history graph, either x is a face of $\mathrm{VR}(r_j, R_j)$ or x is a trapezoid of $V^*(R_j) \setminus V^*(R_{j-1})$. Because the number of faces in the history graph is less than the number of trapezoids (each face-node has at least one unique succeeding trapezoid-node), the number of all vertices of the history graph is $\leq$ two times the number of trapezoid-nodes. Now similar to the proof of Lemma 31 in Section 3.6 we have the following. If x is a trapezoid-node, then $r_j \in D_{R_j}(x)$ which implies:

$$E(\mathcal{H}(R_j) \setminus \mathcal{H}(R_{j-1})) \leq \frac{2}{j} \sum_{r \in R_j} \sum_{A \in \{\text{trapezoids of } V^*(R_j)\}} \delta(A, r),$$

$$\text{where } \delta(A, r) = \begin{cases} 1, & \text{if } r \in \mathrm{set}(D_{R_j}(A)) \\ 0, & \text{else.} \end{cases}$$

$$\leq \frac{2}{j}(16|V^*(R_j)|) \in O(m_j).$$

Summing up over all j from $j = 3$ to n, we get $E(|\mathcal{H}(S)|) \in O(\sum_{j=3}^{n} m_j)$. □

Lemma 39. *The expected size of E_t is $O(m_j)$.*

Proof. See the proof of Lemma 32 in Section 3.6. □

Finally, we obtain our main theorem.

Theorem 7. *Let $\{J(p, q) : p \neq q \in S\}$ be a bisecting curve system fulfilling axioms (A1'), (A2"), (A3), (A4) and (A5). Then $V(S)$ can be computed in expected time*

$$O\Big(s^2 n \log(\max\{s, n\}) \sum_{j=2}^{n} \frac{m_j}{j}\Big),$$

and expected space $O(\sum_{j=3}^{n} m_j)$.

Proof. The sets E_t for all $t \in S$ can be computed in expected time $O(s^2 c \log(\max\{s, n\}))$, see Lemma 36, which is in

$$O\Big(s^2 n \log(\max\{s, n\}) \sum_{j=2}^{n} \frac{m_j}{j}\Big),$$

see Lemma 37. Afterwards $\mathcal{H}(S)$ and $V^*(S)$ can be computed in time

$$O(s^2 \sum_{t \in S} |E_t|),$$

see the end of the previous section, and because the expectation of $|E_t|$ is $O(m_j)$, Lemma 39, the expected value is in

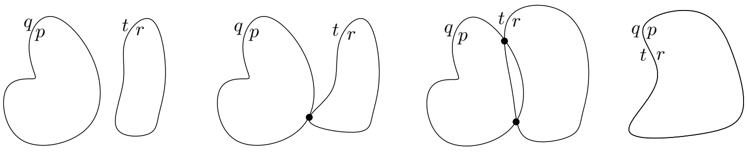
$$O(s^2 \sum_{j=2}^{n} m_j) \in O\Big(s^2 n \log(\max\{s, n\}) \sum_{j=2}^{n} \frac{m_j}{j}\Big).$$

Lemma 38 shows the expected size of $\mathcal{H}(S)$, and the size of the trapezoidal decomposition and the search structures for the faces F is linear in the size of F, see [38], which gives us the expected space. □

4.5 Pseudo-circles: an Application

Let S be a set of n sites and let the set of bisectors $\mathcal{J} := \{J(p,q) : p \neq q \in S\}$ be a set of pseudocircles. This means that each bisector is a simply closed curve and for each pair of bisectors $J(p,q)$ and $J(r,t)$ there are four cases to describe their intersection, see Figure 4.10:

1. $J(p,q)$ and $J(r,t)$ do not intersect at all;
2. $J(p,q)$ and $J(r,t)$ intersect in exactly one point and this intersection is nontransversal;
3. $J(p,q)$ and $J(r,t)$ intersect in exactly two points and both intersection points are transversal;
4. $J(p,q)$ and $J(r,t)$ are equal.



Case 1 Case 2 Case 3 Case 4

Figure 4.10: Pseudocircles.

This curve system fulfills axiom (A1'), (A2"), and (A4). In addition we assume that also (A3) and (A5) are fulfilled. Because all bisectors are bounded, there is exactly one unbounded region in $V(S)$.

Lemma 40. *If $|S| = 3$, then all Voronoi regions are connected.*

Proof. Let $S = \{p, q, r\}$. Then the region of p equals the intersection of the two dominance regions $D(p,q) \cap D(p,r)$. A dominance region is either an open pseudodisc or the open complement of a pseudodisc. It is easy to see that the intersection of two dominance regions is always connected. □

For unbounded bisecting curves we have the property that if for all subsets S' of S of size 3 the Voronoi regions in $V(S')$ are connected, then all Voronoi regions in $V(S)$ are connected, see [49, Lemma 14]. This is something which is no longer true, if bisectors may be closed curves, as has been shown already for multiplicatively weighted Voronoi diagrams [12]. Here one region may consist of $O(n)$ faces and the total complexity of $V(S)$ is $O(n^2)$. Interestingly there is no connection at all between the property that for all subsets S' of S of size k, $1 \leq k \leq n-1$, all regions in $V(S')$ are connected and the property that all regions in $V(S)$ are connected.

Lemma 41. *Let S be a set of n sites. We can choose a set of bisecting curves such that for all proper subsets $S' \subsetneq S$ all Voronoi regions in $V(S')$ are connected but there is a region in $V(S)$ which is disconnected.*

Proof. An example for such a system of bisecting curves is depicted in Figure 4.11. Let $S = \{p_1, \ldots, p_n\}$. The uppermost circle is the bisector $J(p_1, p_2)$, $J(p_2, p_3), \ldots, J(p_2, p_n)$, where $D(p_1, p_2), D(p_3, p_2) \ldots, D(p_n, p_2)$ equal the outer face of the circle. The circle to the right of the uppermost one is the bisector $J(p_1, p_3)$, $J(p_3, p_4), \ldots, J(p_3, p_n)$, where $D(p_1, p_3), D(p_4, p_3), \ldots, D(p_n, p_3)$ equal the outer face of it. Finally, the circle to the left of the uppermost one is the bisector $J(p_1, p_n)$, where $D(p_1, p_n)$ is the outer face.

Now the region of p_1 equals the plane minus the union of the circles. The uppermost disc is the region of p_2, the disc to the right minus its left one is the region of p_3 and so on. Thus, in $V(S)$, where all circles are present, the region of p_1 is disconnected into two faces. But as soon as one circle is missing, i. e., when we have a subset S' not containing one of the sites $p_2, \ldots, p_n$, then the region of p_1 grows together through this circle and becomes connected in $V(S')$. All other regions are connected in all $V(S')$, where S' is a subset of S. □

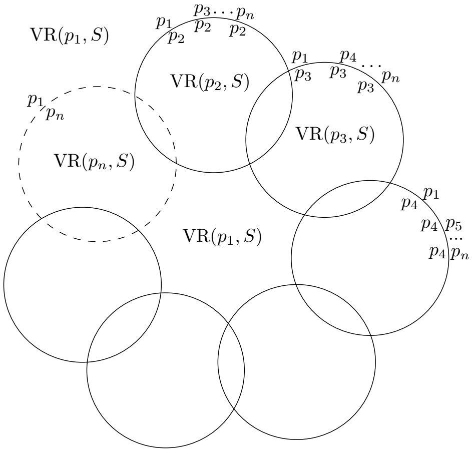


Figure 4.11: An example where the region of p_1 is disconnected in $V(S)$, but for proper subsets $S' \subsetneq S$ all regions in $V(S')$ are connected.

Nevertheless, we have the following complexity bound on the size of the Voronoi diagram. Agarwal et. al. [3] showed that the boundary of the union of n pseudodiscs, for $n \geq 3$, consist of at most $6n - 12$ elementary arcs. In our case we are interested in the complexity of the intersection of pseudodiscs and complements of pseudodiscs, thus the result by Agarwal et. al. is not directly applicable, but the technique used in the last part of our proof is similar.

Theorem 8. *Let S be a set of n sites together with a set of bisecting curves fulfilling our axioms. Then each Voronoi region in $V(S)$ is of size $O(n)$ and $V(S)$ is of size $O(n^2)$. These bounds are tight in the worst case.*

Proof. Let p be a site in S. Its Voronoi region is the intersection of open pseudodiscs and complements of pseudodiscs.

Claim 1. The intersection of n open pseudodiscs is simply connected[1] (or empty).

[1] We call a set *simply connected* if it is both connected and has no holes.

Proof. It is clear that the intersection has no holes, otherwise one of the pseudodiscs would have a hole. Thus, it remains to show that it is connected, which follows from [49, Lemma 14], because if all Voronoi regions in any diagram of three sites are connected, then related unbounded bisectors may intersect in at most two points, and these intersections are transversal. Thus, the bisectors can be connected to closed ones at infinity giving us a system of pseudocircles. Now the claim follows from the lemma. □

Because of Claim 1, the boundary of the intersection of n pseudodiscs is a closed curve consisting of pseudocircle segments. We call these segments edges.

Claim 2. If each edge has the name of the pseudocircle which it is part of, then the edges along the boundary of the intersection of n pseudodiscs is a Davenport-Schinzel-Sequence of order 2.

Proof. Let an edge have the name q if it belongs to the pseudocircle $J(p,q)$. Now suppose there are edges q and r alternating three times in the sequence, i. e., $q \dots r \dots q \dots r$. Then the two bisectors $J(p,q)$ and $J(p,r)$ must intersect in more than two points, a contradiction. □

This claim shows that there can be at most $2n-1$ edges on the boundary of the intersection of n pseudodiscs.

Now recall that $\mathrm{VR}(p,S)$ is the intersection of both open pseudodiscs and complements of pseudodiscs. Let F be the intersection of all $D(p,q)$, where $q \in S \setminus \{p\}$ and $D(p,q)$ is a pseudodiscs. By Claim 2, F is connected and has $O(n)$ edges on its boundary. It is clear that $\mathrm{VR}(p,S)$ is contained in F. Place a vertex on each edge on the boundary of F and for each pseudodisc $D(q,p)$, $q \in S \setminus \{p\}$, (whose complement is a dominance region of p) intersecting F, place a vertex q in $D(q,p) \cap F$; observe that each pseudodisc can intersect F in at most one connected component.

Connect two vertices q and r belonging to $D(q,p)$ and $D(r,p)$ by an edge iff $\partial D(q,p)$ and $\partial D(r,p)$ intersect in a point in F not contained in any other $D(s,p)$, i. e., $J(q,p)$ and $J(r,p)$ intersect in a vertex on the boundary of the region of p. Further, connected a vertex q belonging to $D(q,p)$ and a vertex v belonging to an edge e on the boundary of F iff $\partial D(q,p)$ intersects e in a point not contained in any other $D(r,p)$. Finally, connect two vertices belonging to edges e and e' on the boundary of F, if e and e' are incident.

Now consider the resulting graph G, see Figure 4.12. Because two pseudocircles can intersect in at most two points, G is a planar graph without parallel edges. Further, we can draw the edges e between two vertices $q \in D(q,p)$ and $r \in D(r,p)$ such that $e \subseteq D(q,p) \cup D(r,p)$, which is possible, because by definition e exists only if $D(q,p) \cap D(r,p) \neq \emptyset$. Then it is clear that each connected component of $\mathrm{VR}(p,S)$ corresponds to a unique face of G (but G may have more faces than $\mathrm{VR}(p,S)$). Because G has $O(n)$ vertices it also has $O(n)$ faces and edges, which proves the upper bound of the Theorem.

An example, where the bounds are tight is the Voronoi diagram of multiplicatively weighted points, see [12]. □

Because the size of each Voronoi region is $O(n)$, we have the following theorem.

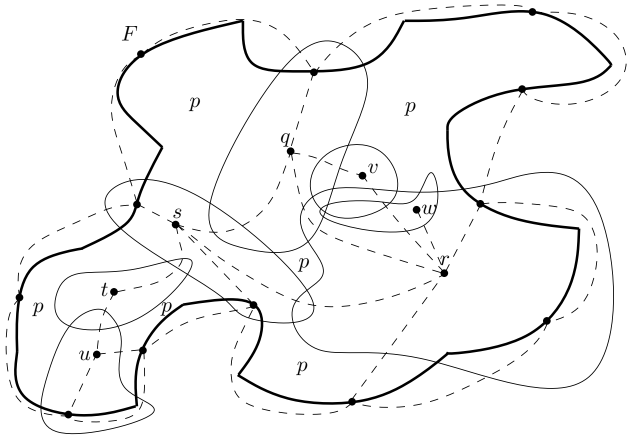


Figure 4.12: Proof of the last part of Theorem 8. Fat curves depict the boundary of F, normal curves the boundaries of the pseudodiscs $D(q,p)$ and dashed curves edges of the graph G.

Theorem 9. *The Voronoi diagram based on a bisector system of pseudocircles fulfilling axioms (A3) and (A5) can be computed in expected time*

$$O(n^2 \log n).$$

The size of the Voronoi diagram is $\Theta(n^2)$, thus in this case the algorithm is nearly optimal.

4.6 Conclusion

Our algorithm covers many concrete Voronoi diagrams, named in the introduction, for which yet no efficient algorithm existed. The run time strongly depends on the size of m_j, the average number of components of a Voronoi region in a diagram of j sites. As we have seen, for pseudocircles this works very nicely and we obtain a nearly optimal run time, but there are examples were m_j is higher than the total complexity of the final diagram, thus in these cases our algorithm would be less efficient.

Also, if one wants to implement the algorithm one may not want to use Chazelle's linear triangulation algorithm [23], but an easier approach, e. g., a randomized one [9]. This may be possible but it would require a new randomized analysis of the run time.

Chapter 5

Higher Order

In this Chapter we study *abstract Voronoi diagrams of arbitrary order k*. In a concrete order-k Voronoi diagram, all points are placed into the same region that have the same k nearest neighbors among the given sites, compare Section 2.2.7. Like for standard *nearest* Voronoi diagrams, this notion can be defined also for the abstract setting.

We prove that their complexity in the plane is upper bounded by $2k(n-k)$. So far, an $O(k(n-k))$ bound has been shown only for point sites in the Euclidean and L_p plane [57, 59], and, recently, for line segments, in the L_p metric [63]. These proofs made extensive use of the geometry of the sites.

Our result on AVD's implies a $2k(n-k)$ upper bound for a wide range of cases for which only trivial upper complexity bounds were previously known, and a slightly sharper bound for the known cases. Also, our proof shows that the reasons for this bound are combinatorial properties of certain permutation sequences.

A preliminary version of this chapter has been published at ICALP'13 [16].

5.1 Introduction

Let S be a set of n sites. For an integer k between 1 and $n-1$, the order-k Voronoi diagram is defined to consist of Voronoi regions, where points belonging to the same region have got the same k nearest sites. For $k=1$ the standard *nearest* Voronoi diagram results, while for $k=n-1$ the *farthest* Voronoi diagram is obtained, where all points of M having the same farthest site belong in the same Voronoi region. Observe that the order of the Voronoi diagram has nothing to do with the dimension of the space of its embedding. We consider Voronoi diagrams in the 2-dimensional plane only.

Recall that in abstract Voronoi diagrams for each pair of sites p and q a curve is given, separating the plane into the two dominance regions $D(p,q)$ on one side and $D(q,p)$ on the other side. *Nearest (Order-1) abstract Voronoi regions* are then defined by

$$\mathrm{VR}(p,S) := \bigcap_{q\in S\setminus\{p\}} D(p,q).$$

To define *higher order abstract Voronoi regions* let $P \subset S$ be a set of k sites. Then the order-k region of P is defined by

$$\mathrm{VR}^k(P,S) := \bigcap_{p\in P,\ q\in S\setminus P} D(p,q),$$

The order-k abstract Voronoi diagram $V^k(S)$ is defined to be the complement of all order-k Voronoi regions in the plane; it equals the collection of all edges that separate order-k Voronoi regions (Lemma 44).

If $D(p,q)$ is the set of all points that are nearer to p than to q with respect to a distance measure, $\mathrm{VR}(P,S)$ equals the set of points having P has their k-nearest sites, thus our definition makes completely sense.

Farthest abstract Voronoi diagrams consist of regions

$$\mathrm{VR}^{-1}(p,S) := \bigcap_{q\in S\setminus\{p\}} D(q,p).$$

They have been shown to be trees of complexity $O(n)$, computable in expected $O(n\log n)$ many steps [61].

Of course we still have to require some properties about the abstract Voronoi diagrams. Because it is the first time to consider higher order AVD's and we want to keep our proofs simple, we restrict to the basic axiomatic system. We require axioms (A1) to (A3), compare Definition 3 in Section 2.3, saying that for each nonempty subset S' of S we have the following.

(A1) By stereographic projection to the sphere, each curve $J(p,q)$ can be completed to a closed Jordan curve through the northpole.

(A2) Each nearest Voronoi region $V(p,S')$ is path-connected.

(A3) Each point of the plane belongs to the closure of a nearest Voronoi region $\overline{VR(p,S')}$.

In addition we shall assume the following for each nonempty subset S' of S.

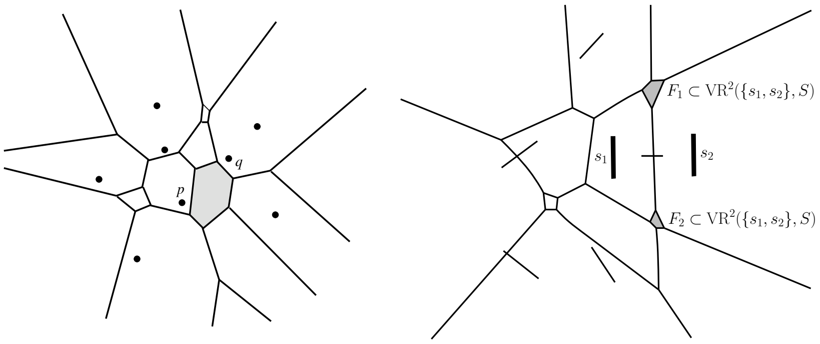
(A4) Any two curves $J(p,q)$ and $J(r,t)$ have only finitely many intersection points, and these intersections are transversal.

(A5) No nearest Voronoi region $VR(p,S')$ is empty.

Observe that our axioms are claimed to hold only for the *nearest*-, i. e., for the order-1 Voronoi diagram. In Lemma 42 we prove that property (A5) need only be tested for all subsets S' of size four. Clearly, (A5) holds in all concrete cases where each nearest region contains its site.

Figure 5.1 shows two concrete order-2 diagrams of points and line segments under the Euclidean metric. We observe that the order-2 Voronoi region of line segments s_1, s_2 is disconnected, whereas for points the higher-order regions are still connected. In general, a Voronoi region in $V^2(S)$ can have $n-1$ connected components [63]. Figure 5.2 depicts a curve system fulfilling all properties required, and the resulting abstract Voronoi diagrams of orders 1 to 4. An index p placed next to a curve indicates the side of the curve where $D(p,q)$ lies. The dashed curves represent arcs of the bisectors that do not appear as Voronoi edges in the diagram. The order-2 region of p_1 and p_2 consists of four connected components.

In this chapter we are proving the following result on the number of 2-dimensional faces of order-k abstract Voronoi diagrams.



p
q
$F_1 \subset \mathrm{VR}^2(\{s_1, s_2\}, S)$
s_1
s_2
$F_2 \subset \mathrm{VR}^2(\{s_1, s_2\}, S)$

Figure 5.1: Order-2 diagrams of points and line segments. The shadowed region in the left picture belongs to the sites p and q and is connected, in the right picture the shadowed faces belong to the region of s_1 and s_2 which is disconnected.

Theorem 10. *The abstract order-k Voronoi diagram $V^k(S)$ has at most $2k(n-k)$ many faces.*

So far, an $O(k(n-k))$ bound has been shown only for points [57], and recently for line-segments [63], in the L_p metric.[1] The structural properties of the diagram for line segments turned out surprisingly different from those for points, including disconnected Voronoi regions and a lack of symmetry between the number of unbounded regions in the order-k and order-$(n-k)$ diagrams. The differences clearly propagate in the abstract setting. The proofs of these results are based on geometric arguments, including results on k-sets[2], point-line duality and $\leq k$-levels in arrangements. None of these arguments applies to abstract Voronoi diagrams.

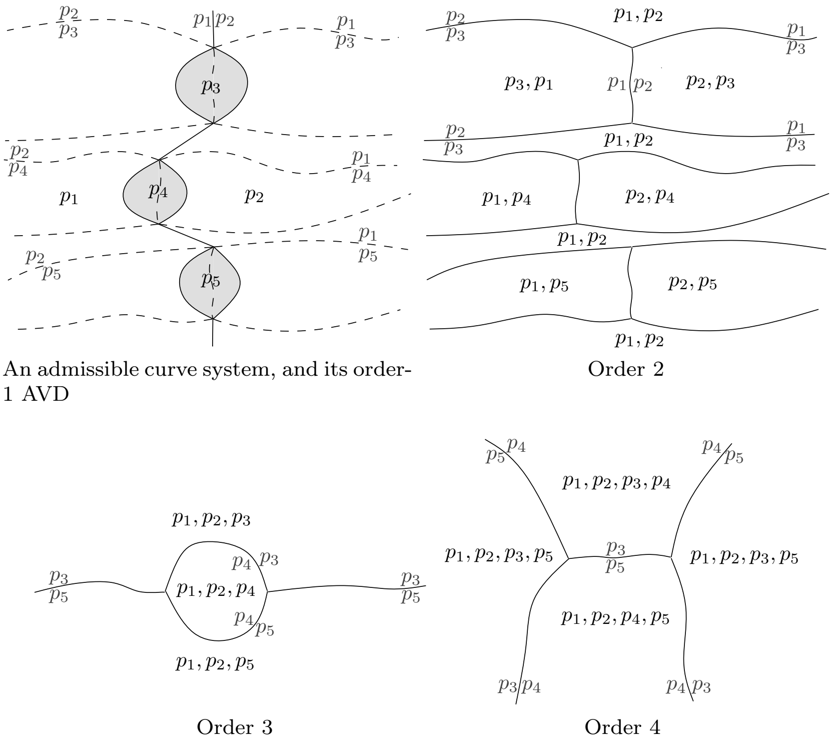
However, the upper bound on k-sets established in [8] had a combinatorial proof; it was obtained by analyzing the cyclic permutation sequences that result when projecting n point sites onto a rotating line. In such a sequence, consecutive permutations differ by a switch of adjacent elements, and permutations at distance $\binom{n}{2}$ are inverse to each other.

In this chapter we prove Theorem 10 using combinatorial arguments. We traverse the unbounded edges of higher order AVD's, and obtain a strictly larger class of cyclic permutation sequences, where consecutive permutations differ by switches and any two elements switch exactly twice. Our proof is based on a tight upper bound to the number of switches that can occur among the first $k+1$ elements; see Lemma 49. It is interesting to observe that in our class, each permutation sequence can be realised by an AVD (Lemma 50), while this is not the case for the sequences obtained by point projection [42].

It is tempting to think that one could easily use the techniques from Clarkson and Shor [30] to prove these bounds. But for abstract Voronoi diagrams there appear some special phenomena which complicate this task, as will be discussed in the conclusion of this chapter, Section 5.6.

[1] In the L_1 and L_∞ metrics, the bound is slightly tighter, i.e., $O((n-k)^2)$ for $k > n/2$ [63].

[2] We call a subset of size k of n points a *k-set* if it can be separated by a line passing through two other points. Such k-sets correspond to unbounded order-$(k+1)$ Voronoi edges.



An admissible curve system, and its order-1 AVD

Order 2

Order 3

Order 4

Figure 5.2: AVD of 5 sites in all orders.

To avoid technical complications we are assuming, in the first 4 sections of this chapter, that any two input curves $J(p, q)$ intersect in a finite number of points, these intersections are transversal, and that Voronoi vertices are of degree 3. How to get rid of the first assumption has been shown for the case of nearest AVD's in [49]. In Section 5.5 we show that also vertices of higher degree than 3 can be allowed.

Theorem 10 implies a $2k(n-k)$ upper complexity bound on a wide range of order-k Voronoi diagrams for which no good bounds were previously known. For example, sites may be disjoint convex objects of constant complexity in L_2 or under the Hausdorff metric. For point sites, distance can be measured by any metric d satisfying the following conditions: points in general position have unbounded bisector curves; d-circles are of constant algebraic complexity; each d-circle contains an L_2-circle and vice versa; for any two points $a \neq c$ there is a third point $b \neq a, c$ such that $d(a, c) = d(a, b) + d(b, c)$ holds. This includes all convex distance functions of constant complexity, but also the Karlsruhe metric where motions are constrained to radial or circular segments with respect to a fixed center point. A third example are point sites with additive weights a_p, a_q that satisfy $|a_p - a_q| < |p - q|$, for any two sites $p \neq q$; see [13] for a discussion of these examples.

Using Theorem 10, the first algorithm computing the abstract order-k Voronoi diagram has been presented in [20]. It runs in expected time $O(kn^{1+\epsilon})$.

The rest of this chapter is organized as follows. In Section 5.2 we present some basic facts about AVD's. Then, in Section 5.3, permutation sequences will be studied, in order to establish an upper bound to the number of unbounded Voronoi edges of order at most k. This will lead, in Section 5.4, to a tight upper bound for the number of faces of order k. Finally, in Section 5.5 we show how to get rid of the general position assumption and maintain the same result for the number of order-k faces.

5.2 Preliminaries

In this section we present some basic facts on abstract Voronoi diagrams of various orders.

Each transversal intersection v between two related bisectors $J(p,q)$ and $J(p,r)$ is a Voronoi vertex in the order-1 diagram of $\{p,q,r\}$, and $J(q,r)$ runs also through v, compare [48]. Together with the *Euler formula* and our axioms we obtain that any two related bisectors $J(p,q)$ and $J(p,r)$ may intersect in at most two points.

Fortunately, verification of our axioms can be based on constant size examples. Pairs and quadruples of sites are clearly sufficient to verify axioms (A1) and (A4). Axioms (A2) and (A3) can be verified by checking all subsets of S of size 3, see [49, Section 4.3]. For axiom (A5) we need subsets of size 4 as shown in the following lemma, to check all subsets of size 3 would not be enough, as shown in Figure 5.3.

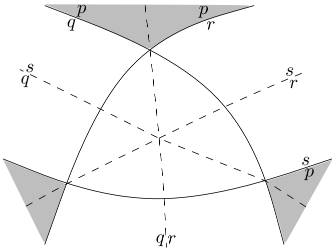


Figure 5.3: An AVD of 4 sites p,q,r,s. In each diagram of three of these sites no Voronoi region is empty (the regions of p are shaded) but in the diagram of all 4 sites the region of p is empty.

Lemma 42. *To verify axiom (A5), assuming that axioms (A2) and (A3) hold, it is sufficient to check all subsets S' of S of size 4.*

Proof. If all bisecting curves are straight lines, (A5) follows from Helly's theorem on convex sets, stating that for a finite collection of convex sets in the plane, if each intersection of 3 of these sets is nonempty, then the intersection of the whole collection is nonempty.

Our bisecting curves $J(p,q)$ do not necessarily define convex domains $D(p,q)$ from which the Voronoi region $\mathrm{VR}(p,S) = \bigcap_{q \in S \setminus \{p\}} D(p,q)$ is constructed. This is why a new proof is needed.

So, assume that $\mathrm{VR}(p,S')$ is nonempty for all subsets S' of size 4. Let $|S| > 4$ and for the sake of a contradiction assume that $\mathrm{VR}(p,S) = \emptyset$. Let $q_1, q_2, q_3 \in S \setminus \{p\}$ be pairwise different. By induction on the size of S, there exist points $x_i \in \mathrm{VR}(p, S \setminus \{q_i\})$, $i \in \{1, 2, 3\}$. Since no point lies in $\mathrm{VR}(p, S)$, we have $x_i \in D(q_i, p)$. By (A2), there are paths P_{ij} connecting x_i and x_j in $\mathrm{VR}(p, S \setminus \{q_i, q_j\}) \subseteq D(p, q_k)$, where $\{i, j, k\} = \{1, 2, 3\}$. Because $\mathrm{VR}(p, S)$ may not be empty, P_{ij} has to be contained in $D(q_i, p) \cup D(q_j, p)$, thus when traversing P_{ij} from x_i to x_j one must cross $J(p, q_j)$ first, before one crosses $J(p, q_i)$. W.l.o.g. we can assume that P_{ij} intersects $J(p, q_i)$ and $J(p, q_j)$ exactly once each.

Let x'_1 be the first point of P_{13} one meets, when traversing P_{12} from its intersection with $J(p, q_2)$ in the direction of x_1, let x'_2 be the first point on P_{23} one meets, when traversing P_{12} from its intersection with $J(p, q_1)$ in the direction of x_2, and let x'_3 be the first point on P_{23} one meets, when traversing P_{13} from its intersection with $J(p, q_1)$ in the direction of x_3. Because P_{23} can not intersect P_{12} in $D(q_1, p)$, P_{13} can not intersect P_{12} in $D(q_2, p)$, and P_{12} can not intersect P_{23} in $D(q_3, p)$, the points x'_1, x'_2 and x'_2 together with the segments of P_{12}, P_{23}, and P_{13} in between form a simple closed curve which bounds a domain D.

We have x'_i in $D(q_i, p)$ and in $D(p, q)$ for all $q \in S \setminus \{q_i\}$ and $J(p, q_i)$ may not intersect P_{jk} for $\{i, j, k\} = \{1, 2, 3\}$. Further, because $P_{12} \subseteq D(q_1, p) \cup D(q_2, p)$ the two bisectors $J(p, q_1)$ and $J(p, q_2)$ must intersect transversally in a point w contained in the domain D; see Figure 5.4. The two bisectors $J(p, q_1)$ and $J(p, q_2)$ divide the domain D into four subdomains around w, $A \subseteq D(q_1, p) \cap D(q_2, p)$, $B \subseteq D(p, q_1) \cap D(q_2, p)$, $C \subseteq D(p, q_2) \cap D(p, q_1)$, and $E \subseteq D(p, q_2) \cap D(q_1, p)$.

The bisector $J(p, q_3)$ enters D through P_{13} and E such that the segment of P_{13} in the direction of x'_1 is in $D(p, q_3)$ and the segment in the direction of x'_3 is in $D(q_3, p)$. $J(p, q_3)$ leaves D through B and P_{23} such that the segment of P_{23} in the direction of x'_2 is in $D(p, q_3)$ and the segment in the direction of x'_3 is in $D(q_3, p)$. Because $J(p, q_3)$ may intersect $J(p, q_1)$ and $J(p, q_2)$ in at most two points, if it traverses A then it can not traverse C and vice versa, and it can traverse A and C in at most one segment.

Now there are three cases, see Figure 5.5.

Case1: $J(p, q_3)$ traverses A. Then it forms a bounded domain of the farthest region $\mathrm{VR}^{-1}(p, \{p, q_1, q_2, q_3\})$ in A incident to w. This contradicts Lemma 47 (given later) whose proof is independent from this lemma.

Case2: $J(p, q_3)$ traverses C. Then it bounds $\mathrm{VR}(p, \{p, q_1, q_2, q_3\})$ in C incident to w. Let $x \in \mathrm{VR}(p, \{p, q_1, q_2, q_3\})$. Because $\mathrm{VR}(p, S) = \emptyset$ there must be $q \in S \setminus \{p, q_1, q_2, q_3\}$ such that $x \in D(q, p)$. But the boundary of the domain D is contained in $D(p, q)$, implying that $J(p, q)$ is closed—a contradiction to (A5).

Case3: $J(p, q_3)$ traverses neither A nor C and must hence run through w. Since $\mathrm{VR}(p, \{p, q_1, q_2, q_3\})$ must not be empty, by (A3), $J(p, q_3)$ has to traverse through $D(p, q_1) \cap D(p, q_2)$, but then it must intersect $J(p, q_1)$ or $J(p, q_2)$ in another point, resulting in a disconnection of $\mathrm{VR}(p, \{p, q_1, q_3\})$ or $\mathrm{VR}(p, \{p, q_2, q_3\})$ that contradicts (A2).

□

Remember also the very useful fact that for all p, q, r in S, $D(p, q) \cap D(q, r) \subseteq D(p, r)$ holds, compare Lemma 7 in Section 2.3.

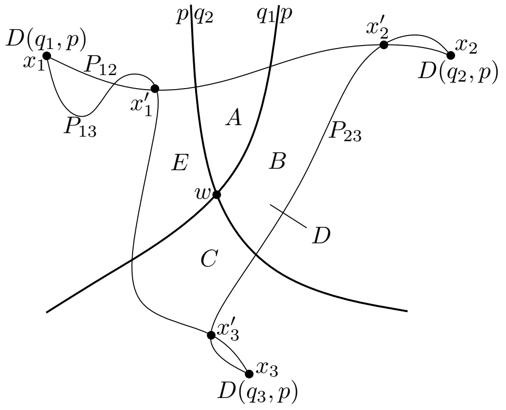


Figure 5.4: In the proof of Lemma 42, curves $J(p,q_1)$ and $J(p,q_2)$ meet at point w.

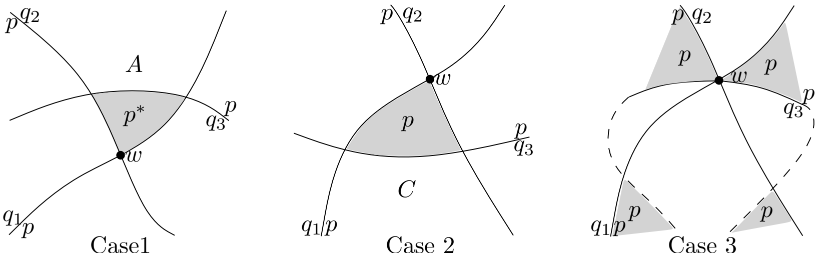


Figure 5.5: Discussion of three cases.

Consequently, a total ordering of the set S is possible for $x \notin \bigcup_{p,q\in S} J(p,q)$, where

$$p <_x q :\Longleftrightarrow x \in D(p,q).$$

Informally, one can interpret $p <_x q$ as "x is closer to p than to q". We will write $p < q$ if it is clear which $x \in \mathbb{R}^2$ we are referring to.

As a direct consequence we show that property (A3) holds also for abstract order-k Voronoi regions.

Lemma 43. *Let $J = \{J(p,q) : p \neq q \in S\}$ be an admissible curve system. Then for each $k \in \{1,\ldots,n-1\}$*

$$\mathbb{R}^2 = \bigcup_{P\subseteq S, |P|=k} \overline{VR^k(P,S)}.$$

Proof. Let $x \in \mathbb{R}^2$. If x is not contained in any bisecting curve $J(p,q)$ then it belongs to the order-k region $\mathrm{VR}^k(P,S)$, where $P = \{p_1,\ldots,p_k\}$ are the k smallest elements of S with respect to the ordering $<_x$. Otherwise, x lies on the boundary of a domain $D \subset \mathbb{R}^2 \setminus \bigcup_{p\neq q\in S} J(p,q)$, and D fully belongs to an order-k region. □

The proofs of the following Lemmata 44 and 45 are similar to the proof of Lemma 43. Lemma 45 indicates that two neighboring regions differ in exactly one site.

Lemma 44.
$$V^k(S) = \bigcup_{\substack{P \neq P' \subset S \\ |P|=|P'|=k}} \overline{VR^k(P,S)} \cap \overline{VR^k(P',S)}$$

Lemma 45. *If the intersection $E := \overline{VR^k(P,S)} \cap \overline{VR^k(P',S)}$ is not empty, there are sites $p \in P$ and $p' \in P'$ such that $P \setminus \{p\} = P' \setminus \{p'\}$, and $E \subseteq J(p,p')$ holds. For each point $x \in VR^k(P,S)$ near E, index p is the k-th with respect to $<_x$, while for points x' in $VR^k(P',S)$ index p' appears at position k.*

In particular, $D(p,p')$ is on the same side of $J(p,p')$ as $\mathrm{VR}^k(P,S)$.

If F, F' are connected components (*faces*) of $\mathrm{VR}^k(P,S)$ and $\mathrm{VR}^k(P',S)$, respectively, the intersection $\overline{F} \cap \overline{F'}$ can be empty, or otherwise be of dimension 0 (*Voronoi vertices*) or 1 (*Voronoi edges*).

For the next lemma we assume that all vertices are of degree 3. Recall that if two related bisectors $J(p,q)$ and $J(p,r)$ intersect in a point v, then $J(q,r)$ runs also through v, [48], and v is a Voronoi vertex of the order-1 Voronoi diagram of $\{p,q,r\}$. Thus, when assuming that all Voronoi vertices are of degree 3, no other bisector related to one of the sites p,q,r can run through v.

Let $P_1, P_2, P_3 \subset S$ be the sets defining the adjacent to v order-k Voronoi regions in clockwise order, see Figure 5.6. As with the concrete order-k Voronoi diagrams, vertex v can be of two types, depending on the nature of sets P_1, P_2, P_3 [57]. There are two cases. In the first case, there exists a set $H \subset S$ of size $k-1$ and three more sites $p,q,r \in S$ satisfying

$$P_1 = H \cup \{p\}, \quad P_2 = H \cup \{q\}, \quad P_3 = H \cup \{r\}.$$

Then vertex v is called *new* in $V^k(S)$, or *of nearest type.* In the second case, there is a subset $K \subset S$ of size $k-2$ and three more sites $p,q,r \in S$ such that

$$P_1 = K \cup \{p,r\}, \quad P_2 = K \cup \{q,r\}, \quad P_3 = K \cup \{p,q\}.$$

Then vertex v is called *old* in $V^k(S)$, or *of farthest type.*

To see that these are indeed the only two cases, we walk around v in clockwise order along the boundary of an ϵ-neighborhood, see Figure 5.6. Let the edge between P_1 and P_2 belong to the bisector $J(p,q)$, such that $P_1 \subseteq D(p,q)$ and $P_2 \subseteq D(q,p)$. If q is still last in the ordering of the sites when we reach the edge between P_2 and P_3, then this edge must belong to the bisector $J(q,r)$ for an $r \in S \setminus \{p,q\}$. Thus, bisector $J(r,p)$ must run through v, and because no other bisector related to r may contain v, a segment of $J(r,p)$ defines the edge between P_2 and P_3. Thus, we have the first case.

If q is no longer last in this ordering, then we must have crossed a bisector $J(q,r)$ related to q while traversing P_2. Again, $J(p,r)$ must contain v, and thus, no other bisector related to r may run through v. Thus, r is in last position when we reach the edge between P_2 and P_3 and the edge must belong to $J(r,p)$. Next, because all bisector intersections are transversal (A4), we cross the bisector $J(p,q)$, and p changes its position in the ordering with q, which is now in last position. The next bisector is $J(q,r)$, and because q is in last position, this defines the edge between P_2 and P_3. This implies the second case and no other configuration is possible.

The proof of the following lemma follows quite directly from these definitions.

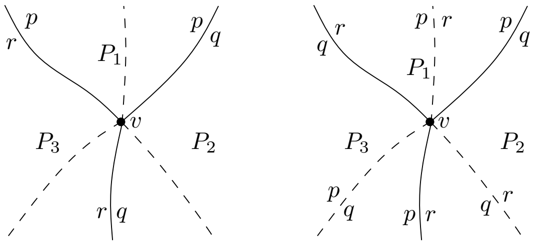


Figure 5.6: A new vertex v to the left and an old vertex to the right. Solid curves indicate the Voronoi edges and dashed ones the prolongations of the bisectors.

Lemma 46. *Let v be a new vertex in $V^k(S)$. Then v is an old vertex of $V^{k+1}(S)$, and v lies in the interior of a face of $V^{k+2}(S)$, i. e., v is not a vertex of $V^{k+2}(S)$. Furthermore, every edge of $V^k(S)$ is enclosed by a face of $V^{k+1}(S)$.*

Already in [61] it has been shown that farthest abstract Voronoi diagrams are trees, under a slightly different definition of admissible curves. In this chapter we give a short alternative proof of this fact based on our axioms (A1)–(A5).

Lemma 47. *The farthest abstract Voronoi diagram $V^{-1}(S)$ is a tree.*

Proof. Let $|S| = n$. Suppose some farthest region $\mathrm{VR}^{-1}(p, S)$ has a face F that is bounded. Let its boundary consist of edges $e_1, \ldots, e_i$ in this order, such that each edge e_j is a segment of a bisector $J(p, q_j)$. The sites q_j need not be pairwise different, but consecutive edges belong to different bisecting curves. Let x_j be the intersection point between e_j and e_{j+1} on the boundary of F, and x_i the intersection point between e_i and e_1 on F, see Figure 5.7.

If $n = 1$, then there exists no bounded Voronoi region. If $n = 2$, there can be at most one edge e_1 on the boundary of $V^{-1}(p, S)$, implying that e_1 and $J(p, q_1)$ would be closed a contradiction to (A1).

Now let $n > 2$, and let $i \geq 2$. By induction on n, $\mathrm{VR}^{-1}(p, S \setminus \{q_i\})$ is unbounded (it still contains F and is thus not empty). Let z be a point in F, then there exists an unbounded arc $L \subseteq VR^{-1}(p, S \setminus \{q_i\})$ emanating from z to infinity. L may not intersect any $J(p, q_1), \ldots, J(p, q_i)$ except for $J(p, q_i)$, thus it must leave F through $J(p, q_i)$, w.l.o.g. let its last intersection with the boundary of F be on e_i.

Because of (A4), there is a point x in the ϵ-neighborhood of x_{i-1} contained in $D(p, q_{i-1}) \cap D(p, q_i)$ and a point y in the ϵ-neighborhood of x_i contained in $D(p, q_1) \cap D(p, q_i)$. We claim that any path π_{xy} from x to y intersect $J(p, q_i)$ or L. This is immediate, because by construction x and y are on the same side of the unbounded curve $J(p, q_i)$, but on opposite sides of L.

Because of (A5), there exists a point $w \in \mathrm{VR}(p, \{p, q_1, q_{i-1}, q_i\})$ (q_1 may be equal to q_i). The point w is also contained in both $\mathrm{VR}(p, \{p, q_1, q_i\})$ and $\mathrm{VR}(p, \{p, q_{i-1}, q_i\})$. The point y is contained in $\mathrm{VR}(p, \{p, q_1, q_i\})$ and because of (A1) there exists a path π_{yw} from y to w in $\mathrm{VR}(p, \{p, q_1, q_i\})$ and thus intersecting neither L nor $J(p, q_i)$. The point x is contained in $\mathrm{VR}(p, \{p, q_{i-1}, q_i\})$ and thus there exists a path π_{xw} from x to w in $\mathrm{VR}(p, \{p, q_{i-1}, q_i\})$ also intersecting neither L nor $J(p, q_i)$. But then the concatenation of π_{xw} and π_{yw} is a path connecting x and y without intersecting L and $J(p, q_i)$, a contradiction.

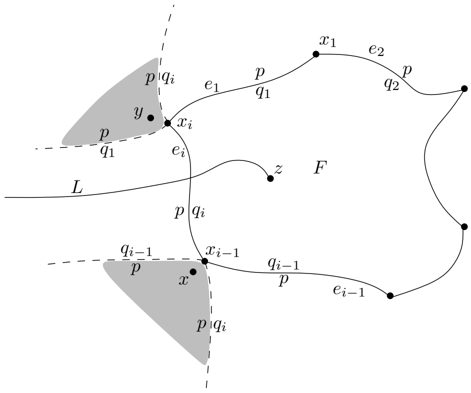


Figure 5.7: Farthest AVD's cannot contain bounded regions.

It remains to show that $V^{-1}(S)$ is connected. Suppose there is a curve C separating parts of $V^{-1}(S)$. Then $C \subset \mathrm{VR}^{-1}(p, S)$ for a $p \in S$, $C \cap D(p,q) = \emptyset$ for all $q \in S \setminus \{p\}$ and there are $q \neq r \in S$ such that $D(p,q)$ lies on one side of C and $D(p,r)$ on the other side. But then $\mathrm{VR}(p, \{p,q,r\})$ would be empty. □

5.3 Bounding the number of unbounded edges of $V^{\leq k}(S)$

Let Γ be a closed Jordan curve in $\mathbb{R}^2$ large enough such that no pair of bisectors cross on or outside of Γ (axiom (A4)), each bisector crosses Γ exactly twice and these intersections are transversal (axiom (A1)).
If we traverse Γ around the Voronoi diagram, the ordering $<_x$ on S changes whenever we cross a bisector $J(p,q)$. Here indices p and q change their relation according to $<_x$.

Lemma 48. *When we cross a bisector $J(p,q)$, then p and q change their places in the ordering $<_x$ along Γ. Further, they are adjacent to each other just before and after the crossing.*

Proof. Let $p_i < p_j$, and assume that we cross $J(p_i,p_j)$, which means that $p_i < p_j$ changes to $p_j < p_i$. Let $p_k < p_i, p_j$ before we cross $J(p_i,p_j)$. Because of the construction of Γ no other bisector has been crossed at the same time, thus $p_k < p_i, p_j$ remains. The same happens for a $p_k > p_i, p_j$. Now let $p_i < p_k < p_j$ before we cross $J(p_i,p_j)$. Then we still have $p_i < p_k < p_j$ afterwards, but now $p_j < p_i$, a contradiction to the transitivity. Thus p_i and p_j must have been adjacent right before and after we cross $J(p_i,p_j)$ and hence they change their places in the ordering. □

We call such a change in the ordering of S a *switch* between the two sites p and q, which are adjacent. There can be only one switch at a time and each pair of sites switches exactly two times while walking one round around Γ, resulting in $n(n-1)$ switches altogether.

Every time a switch among the first $k+1$ elements of the ordering occurs, there is an unbounded edge of a Voronoi diagram of order $\leq k$. This means that *the maximum number of unbounded edges of all diagrams of order $\leq k$ is equal to the maximum number of switches among the first $k+1$ elements in the ordering.*

Permutation sequences and estimates for the maximum number of switches among the first k elements have been used in [8] to bound the number of k-sets of n points in the plane. These sequences resulted from projecting n points in general position onto a rotating line. Hence, they were of length $2N$, where $N = \binom{n}{2}$, and they had the following properties. Adjacent permutations differ by a transposition of adjacent elements, and any two permutations a distance N apart are inverse to each other. It has been shown in [42] that not every permutation of this type can be realized by a point set.

In the following lemma we introduce a larger class of permutation sequences that fits the AVD framework.

Lemma 49. *Let $P(S)$ be a cyclic sequence of permutations $P_0, \ldots, P_N = P_0$ such that*

(i) P_{i+1} differs from P_i by a (adjacent) switch;

(ii) each pair of sites $p, q \in S$ switches exactly two times in $P(S)$.

Then the number of switches occuring in $P(S)$ among the first $k+1$ sites is upper bounded by $k(2n-k-1)$. Furthermore, this bound is tight.

Proof. Call a switch *good* if it involves at least one of the k first sites of a permutation; otherwise call it *bad*. Let $S = \{p_1, \ldots, p_n\}$ and the initial ordering of the sites in the first permutation be $p_1 < \ldots < p_n$. For $i \in \{k+2, \ldots, n\}$, define B_i as the set of bad switches where p_i is switching with a site in $\{p_1, \ldots, p_{i-1}\}$. We remark that the sets B_i, for $i \in \{k+2, \ldots, n\}$, are pairwise disjoint. If p_i is not involved in a good switch, then all its $2i-2$ switches with sites in $\{p_1, \ldots, p_{i-1}\}$ are bad. Otherwise, for p_i to be involved in a good switch, it must first be involved in at least $i-k-1$ bad switches with sites in $\{p_1, \ldots, p_{i-1}\}$, in order to reach the first $k+1$ positions, and since $P_0 = P_N$, p_i has to be involved in as many bad switches in order to return to its original place in the ordering. In both cases, $|B_i| \geq 2(i-k-1)$. Because of (ii), the total number of switches is $N = 2\binom{n}{2}$. Therefore the number of good switches is at most

$$2\binom{n}{2} - \sum_{i=k+2}^{n} |B_i| \leq 2\binom{n}{2} - 2\sum_{j=1}^{n-k-1} j = k(2n-k-1),$$

where $j = i-k-1$.

To show that the bound is tight, let again the initial ordering of the first permutation be $p_1 < \ldots < p_n$. We switch each p_i with $p_{i-1}, \ldots, p_1$ in decreasing index order such that p_i now is the first element and then in inverse order back to its original position. Start with $i = 2$ and continue until $i = n$. Then the number of switches among the first $k+1$ sites is exactly $2\binom{n}{2} - 2\sum_{j=1}^{n-k-1} j$. □

In contradistinction to the result in [42], each such permutation sequence can be realized by an AVD. The following Lemma 50 will be used for proving that the upper bound shown in Lemma 51 is tight.

Lemma 50. *Let $P(S)$ be a sequence of permutations as in Lemma 49. There exists an abstract Voronoi diagram where the ordering of the sites along Γ changes according to $P(S)$.*

Proof. We show that for $|S| = 3$ each $P(S)$ fulfilling the above properties can be realized by an AVD. If $|S| \geq 3$, we can consider $V(S)$ such that each triple p, q, r of sites changes its ordering on Γ according to $P(S)$. This is possible because if the curve system of each triple of sites is admissible then the curve system of S is admissible, too; see [49]. Now if there are two bisectors $J(p, q)$ and $J(r, t)$ having a different order on Γ than p, q, r, t have in $P(S)$, then p, q, r, t are pairwise different, and neither of the bisectors $J(p, r), J(p, t), J(q, r)$ or $J(q, t)$ can occur between the two bisectors $J(p, q)$ and $J(r, t)$ on Γ. Otherwise, suppose w. l. o. g. that $J(p, r)$ occurs between $J(p, q)$ and $J(r, t)$; then $\{p, q, r\}$ would not have the same ordering on Γ as in $P(S)$, a contradiction to our assumption. Thus the bisectors $J(p, q)$ and $J(r, t)$ can be deformed such that their ordering on Γ changes, but the structure of $V(S)$ remains the same.

Now let $S = \{p, q, r\}$. Then there are three different cases:

(1) Each site switches into the first position exactly once.

(2) One site switches into the first position exactly twice; it cannot do so more often because then it would have to switch with one of the other sites more often than twice. Further, it implies that all other sites must switch themselves into first position exactly once.

(3) One site never moves to first position. This implies that both the other sites switch to first position exactly once; otherwise, either one site would remain in first position during the whole permutation, but then it would never switch with any other site, or the two other sites would have to switch more than twice.

Let $P_0 = (p, q, r)$. Then there are two possibilities for P_1 in case (1): Either $P_1 = (q, p, r)$, which leads to the sequence
$P_0 = (p, q, r)$
$P_1 = (q, p, r)$
$P_2 = (q, r, p)$, otherwise r never switches into first position or p switches into first position twice
$P_3 = (r, q, p)$, otherwise r never switches into first position
$P_4 = (r, p, q)$, otherwise q switches into first position a second time
$P_5 = (p, r, q)$, otherwise p and q switch more than twice
$P_6 = (p, q, r)$, otherwise $P_0 \neq P_6$;
or $P_1 = (p, r, q)$, which leads to the same permutation sequence in inverse order.

Assume that p is the site that switches into first position twice in case (2). Then we get the following permutation sequence:
$P_0 = (p, q, r)$
$P_1 = (q, p, r)$, then
$P_2 = (p, q, r)$, otherwise $P_2 = (q, r, p)$ which leads to the permutation sequence as in case (1)
$P_3 = (p, r, q)$, otherwise p and q switch more than twice
$P_4 = (r, p, q)$, otherwise r never switches into first position
$P_5 = (p, r, q)$, otherwise p and q switch more than twice

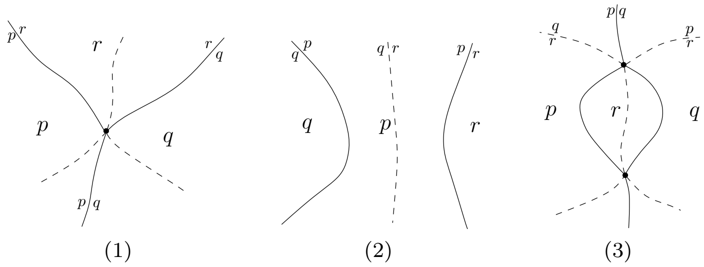


Figure 5.8: Illustrations of cases (1) to (3) in the proof of Lemma 50.

$P_6 = (p, q, r)$, otherwise $P_0 \neq P_6$;
or in inverse order.

Assume that r is the site that never switches into first position in case (3). Then we get the following permutation sequence:
$P_0 = (p, q, r)$
$P_1 = (q, p, r)$, then
$P_2 = (q, r, p)$, otherwise p switches into first position twice
$P_3 = (q, p, r)$, otherwise r switches into first position
$P_4 = (p, q, r)$, otherwise p and r switch more than twice
$P_5 = (p, r, q)$, otherwise p and q switch more than twice
$P_6 = (p, q, r)$, otherwise $P_0 \neq P_6$;
or in inverse order.

These permutation sequences can be realized by the AVD's depicted in Figure 5.8.

□

Let S_i be the number of unbounded edges in $V^i(S)$. If an edge e has got two unbounded endpieces, i. e., edge e bounding a p- and q-region is the whole bisector $J(p, q)$, then e is counted twice as an unbounded edge.

Lemma 51. *Let $k \in \{1, \ldots, n-1\}$. Then,*

$$k(k+1) \leq \sum_{i=1}^{k} S_i \leq k(2n-k-1).$$

Both bounds can be attained.

Proof. The second bound follows directly from Lemma 49. The first bound follows from the fact that the minimum number of switches among the first $(k+1)$ sites is greater or equal to the total number of switches, $n(n-1)$, minus the maximum number of switches among the last $(n-k)$ sites, which again is equal to the maximum number of switches among the first $(n-k)$ sites. Using Lemma 49 this implies

$$\sum_{i=1}^{k} S_i \geq n(n-1) - (n-k-1)(2n-(n-k-1)-1) = k(k+1).$$

The tightness of the bounds follows from Lemma 50. □

5.4 Bounding the number of faces of $V^k(S)$

In the following, we assume that each Voronoi vertex is of degree 3. The following two lemmata give combinatorial proofs for facts that were previously shown by geometric arguments [57, 63].

Lemma 52. *Let H be a subset of S of size $k+1$ and F a face of $VR^{k+1}(H,S)$. The portion of $V^k(S)$ enclosed in F is exactly the farthest Voronoi diagram $V^{-1}(H)$ intersected with F.*

Proof. "$\Rightarrow$": Let $x \in F$ and suppose $x \in \mathrm{VR}^k(H',S)$ for $H' \subset S$ of size k. Since $F \subseteq \mathrm{VR}^{k+1}(H,S)$ it follows that $x \in D(p,q)$ for all $p \in H$ and $q \in S \setminus H$, implying $H' \subset H$. Let $H \setminus H' = \{r\}$, then $x \in D(p,r)$ for all $p \in H'$ and hence $x \in \mathrm{VR}^{-1}(r,H)$.
"$\Leftarrow$": Let $x \in F$ and $x \in \mathrm{VR}^{-1}(r,H)$. Then $x \in D(p,q)$ for all $p \in H$ and $q \in S \setminus H$ and $x \in D(p,r)$ for all $p \in H \setminus \{r\}$. This implies $x \in \mathrm{VR}^k(H \setminus \{r\}, S)$. □

Lemma 53. *Let F be a face of $VR^k(H,S)$, $H \subseteq S$, $|H| = k \geq 2$. Then $V^{-1}(H) \cap F$ is a nonempty tree.*

Proof. First we show that $V^{-1}(H) \cap F$ is not empty by assuming the opposite. Then there is a $p \in H$ such that $F \subseteq \mathrm{VR}^{-1}(p,H)$. Let $F' \subseteq \mathrm{VR}^k(H',S)$ be a face of $V^k(S)$ adjacent to F along an edge e. By Lemma 45, we have $H = U \cup \{q\}$ and $H' = U \cup \{q'\}$, where q, q' are different and not contained in U. Also, $e \subseteq J(q,q')$ holds. If p were in U, we would obtain $F' \subseteq D(p,q)$ and $F \subseteq VR^{-1}(p,H) \subseteq D(q,p)$, hence $e \subseteq J(p,q)$—a contradiction to axiom (A4). Thus, $p \notin U$, which means $p = q$. Now Lemma 45 implies that each edge on the boundary of F has to be a segment of a curve $J(p,q_j)$ such that $D(p,q_j)$ lies on the F-side. Let $q_1, \ldots, q_i$ be the sites for which there is such an edge e on the boundary of F. Then $\mathrm{VR}^1(p, \{p, q_1, \ldots, q_i\}) = F$, because nearest Voronoi regions are connected thanks to axiom (A1). But from $F \subseteq VR^{-1}(p,H)$ it follows that $\mathrm{VR}^1(p,H) \subseteq \mathbb{R}^2 \setminus F$, and hence, $\mathrm{VR}^1(p,S) \subseteq F \cap \mathbb{R}^2 \setminus F = \emptyset$, a contradiction to axiom (A5).

Next we show that $V^{-1}(H) \cap F$ is a tree. Because of Lemma 47 it is clear that it is a forest. So it remains to prove that it is connected. Otherwise, there would be a domain $D \subset F$, bounded by two paths $P_1, P_2 \subset F$ of $V^{-1}(H)$ and two disconnected segments e_1 and e_2 of the boundary of F. There is an index $p \in H$ such that $D \subseteq \mathrm{VR}^{-1}(p,H)$. Since $V^{-1}(H)$ is a tree, by Lemma 47, the upper (or: the lower) two endpoints of P_1 and P_2 must be connected by a path P in $V^{-1}(H)$ that belongs to the boundary of $\mathrm{VR}^{-1}(p,H)$; see Figure 5.9. Here path P connects the endpoints of e_1; both curves together encircle a domain D', which is part of $\mathrm{VR}^{-1}(p,H)$. By definition of the farthest Voronoi diagram and because e_1 is on the boundary of F and contained in $\mathrm{VR}^{-1}(p,H)$, there are $q_1, \ldots, q_i$, such that $e_1 \cup P$ consists of segments of $J(p,q_1), \ldots, J(p,q_i)$, and all $D(p,q_j)$ are situated outside of D'; compare Lemma 45. But then $\mathrm{VR}^{-1}(p, \{p, q_1, \ldots, q_i\})$ would be bounded, a contradiction to Lemma 47. □

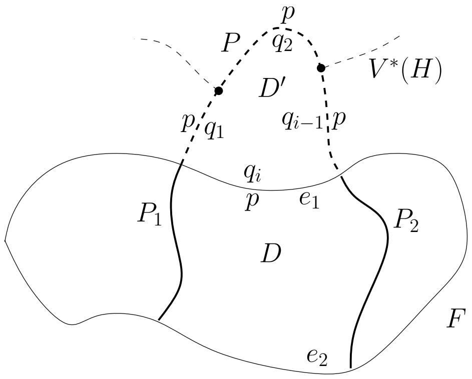


Figure 5.9: The intersection of an order-k face F and the farthest Voronoi diagram of its defining sites must be a tree.

Lemma 54. *Let F be a face of $VR^{k+1}(H,S)$ and m the number of Voronoi vertices of $V^k(S)$ enclosed in its interior. Then F encloses $2m+1$ Voronoi edges of $V^k(S)$.*

Proof. See Lemmata 52 and 53. □

The formulae in the next two lemmata originate in [57]. Below we include their proofs from [63] for completeness.

Lemma 55. *Let F_k, E_k, V_k and S_k denote, respectively, the number of faces, edges, vertices, and unbounded edges in $V^k(S)$. Then,*

$$E_k = 3(F_k - 1) - S_k \tag{5.1}$$

$$V_k = 2(F_k - 1) - S_k. \tag{5.2}$$

Proof. Consider $V^k(S) \cup \Gamma$, cut off all edges outside of Γ, and let G be the resulting graph. Then G is a connected planar graph and for its number of faces, f, of vertices, v, and edges, e, we have $f = F_k + 1$, $v = V_k + S_k$, $e = E_k + S_k$. Because of the general position assumption each vertex is of degree 3 and hence $2e = 3v$. Now the Euler formula $v - e + f = c + 1$ implies the lemma. □

Lemma 56. *The number of faces in an AVD of order k is*

$$F_k = 2kn - k^2 - n + 1 - \sum_{i=1}^{k-1} S_i.$$

Proof. Let V_k, V_k' and V_k'' be the number of Voronoi vertices, *new* Voronoi vertices and *old* Voronoi vertices in $V^k(S)$, respectively. Then because of Lemma 46 we have $V_k = V_k' + V_k'' = V_k' + V_{k-1}'$.

Claim 1: $F_{k+2} = E_{k+1} - 2V_k'$.

Because of Lemma 46, every *old* vertex of $V^{k+1}(S)$ lies in the interior of a face of $V^{k+2}(S)$. Consider a face F_i of $V^{k+2}(S)$. Let m_i be the number of *old* vertices of $V^{k+1}(S)$ enclosed in its interior. Then F_i encloses $e_i = 2m_i + 1$ edges of $V^{k+1}(S)$; see Lemma 54. If we sum up through all the faces in $V^{k+2}(S)$, we obtain

$$\sum_{i=1}^{F_{k+2}} e_i = 2\sum_{i=1}^{F_{k+2}} m_i + F_{k+2}.$$

Note that $\sum_{j=1}^{F_{k+2}} m_j = V''_{k+1} = V'_k$ and $\sum_{j=1}^{F_{k+2}} e_j = E_{k+1}$, hence $F_{k+2} = E_{k+1} - 2V'_k$.

Claim 2: The number of faces in $V^1(S)$ is $F_1 = n$ and the number of faces in $V^2(S)$ is $F_2 = 3(n-1) - S_1$.

The first part follows from axioms (A2) and (A5). To prove the second part, consider a face of $V^2(S)$. There are no *old* vertices in $V^1(S)$, therefore the face encloses exactly one edge of $V^1(S)$ and hence $F_2 = E_1$. Equation (5.1) implies $F_2 = 3(n-1) - S_1$.

Now we sum up F_{k+2} and F_{k+3} to obtain

$$F_{k+3} = E_{k+2} + E_{k+1} - F_{k+2} - 2V'_{k+1} - 2V'_k = E_{k+2} + E_{k+1} - F_{k+2} - 2V_{k+1};$$

(see Claim 1). Substituting (5.1) and (5.2) of Lemma 55 into it results in

$$F_{k+3} = 2F_{k+2} - F_{k+1} - 2 - S_{k+2} + S_{k+1}.$$

Using the iterative formula, the base cases $F_1 = n$ and $F_2 = 3(n-1) - S_1$, we derive the lemma by strong induction. □

Theorem 11. *The number of faces F_k in an AVD of order k is bounded as follows*

$$\begin{aligned} n - k + 1 \leq \quad F_k &\leq 2k(n-k) + k + 1 - n \\ &\leq 2k(n-k) \\ &\in O(k(n-k)). \end{aligned}$$

Both the upper bound $2k(n-k) + k + 1 - n$ and the lower bound $n - k + 1$ can be attained.

Proof. Lemma 51 implies tight bounds $k(k-1) \leq \sum_{i=1}^{k-1} S_i \leq (k-1)(2n-k)$. Together with Lemma 56 this proves the theorem. □

5.5 Generalizations

In [49] it has been shown, with some technical effort, that it is not necessary to require only finitely many intersection points between each pair of bisecting curves $J(p,q)$ and $J(r,t)$. The abstract Voronoi diagram is still a finite planar graph with finitely many vertices even if bisectors intersect in infinitely many points. With the same proofs one can show that this is also true for higher order AVD's. Observe that bisector intersections have to be considered only when resulting in a Voronoi vertex of some order. Thus in axiom (A4) it would suffice to require only transversal intersections.

Another natural question is, whether the general position assumption, where each Voronoi vertex is of degree 3, is really necessary. We can show that it is not and the number of faces of an order k AVD can only decrease if vertices have higher degree than 3 and all bisectors intersect transversally. First we prove that all "pieces of pie" around an order k vertex are from different order k Voronoi regions.

Lemma 57. *Let v be a vertex of $V^k(S)$. All faces adjacent to v are from pairwise different Voronoi regions and if we walk around v, no face appears more than once.*

Proof. The proof that no face appears more than once is analogous to the proof of the simple connectivity of order-1 Voronoi regions, see [49].

Suppose there are two different faces F_1 and F_2 incident to v, both belonging to the same order k Voronoi region of $H \subseteq S$. For $k = 1$ the regions are connected and we have a contradiction.

So let $k \geq 2$. Let the left border-edge of F_1 incident to v belong to the bisector $J(p_1, q_1)$ and the right one to $J(p_2, q_2)$ such that $p_1, p_2 \in H$ and $q_1, q_2 \notin H$, in other words $\mathrm{VR}^k(H, S) \subseteq D(p_1, q_1)$ and $\subseteq D(p_2, q_2)$, see Figure 5.10. Because F_2 is also a face of the region of H, the bisector $J(p_1, q_1)$ has to pass on the left side between F_1 and F_2 and $J(p_2, q_2)$ has to pass on the right side between F_1 and F_2. But then $\{p_1, q_1\} \neq \{p_2, q_2\}$ and the two bisectors $J(p_1, q_1)$ and $J(p_2, q_2)$ intersect non-transversally in v contradicting axiom (A4). □

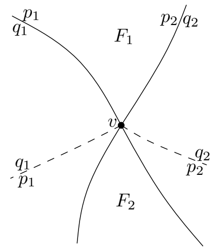


Figure 5.10: Proof of Lemma 57.

Now we can prove a more relaxed version of Theorem 11.

Theorem 12. *Let $V^k(S)$ be an AVD of order k, satisfying axioms (A1) to (A5), with vertex degree ≥ 3 for all Voronoi vertices. Then the number of faces of $V^k(S)$ is bounded from above by $2k(n-k)$.*

Proof. For each vertex v of $V^k(S)$ with degree ≥ 4 consider a sufficiently small ε-neighborhood $U_\varepsilon(v)$ around v, such that no other intersection between bisectors lies in $U_\varepsilon(v)$. Use the perturbation technique of [48] to obtain degree 3 vertices in $U_\varepsilon(v)$. Because of Lemma 57 the modified Voronoi diagram $\overline{V^k}(S)$ has got at least as many faces as $V^k(S)$. Now Theorem 11 implies that $V^k(S)$ has got at most $2k(n-k)$ many faces. □

5.6 Concluding remarks

One may wonder if the Clarkson-Shor technique [30] can be applied to get the same complexity bound. The Clarkson-Shor technique proves that for point sites in the Euclidean metric $V_k(S)$ has at most $2k(n-k-1)$ new Voronoi vertices, leading to at most $4kn - 4k^2 - 2n$ Voronoi vertices. However, there are three problems in applying this technique to the abstract setting. Let $V_{k,S}$, $V'_{k,S}$, and $V''_{k,S}$ be the numbers of Voronoi vertices, new Voronoi vertices, and old Voronoi vertices of $V_k(S)$, respectively, and let $F_{k,S}$ and $S_{k,S}$ be the numbers of faces and unbounded faces of $V_k(S)$, respectively.

First, it is shown that $V'_{k,S} + V'_{n-k,S}$ is $2k(n-k-1)$. This proof depends on the fact that in the Euclidean metric, $V'_{1,R} + V'_{r-1,R}$ is $2r-4$ for any r-element subset R of S. However, the equation $V'_{1,R} + V'_{r-1,R} = 2r-4$ does not hold in the abstract version since $S_{1,R}$ is not necessarily $S_{r-1,R}$. Actually, $r-2 \leq V'_{1,R} + V'_{r-1,R} \leq 2r-4$. It is not clear how to make use of this inequality and their technique to derive an upper bound. Second, the Clarkson-Shor technique mainly focuses on the upper bound. No bound for the minimum number of faces in $V_k(S)$ is derived. Last but not least, if we assume that $V'_{1,R} + V'_{r-1,R} = 2r-4$ holds, by the Euler formula, we have an upper bound of $2k(n-k) + S_{k,S}/2 + 1 - n$, but our derived upper bound is $2k(n-k)+k+1-n$. In the abstract version, it is trivial to find a case in which $S_{k,S}/2 > k$. Moreover, since we prove the existence of an instance satisfying the bound $2k(n-k)+k+1-n$, our bound is tight.

Another natural question is whether axioms weaker than (A1)–(A5) can still imply Theorem 11. In Section 5.5, we showed that axiom (A4) can be simplified, requiring only transversal intersections, and the general position assumption can be omitted. However, what about disconnected and empty Voronoi regions?

Chapter 6

Forest-Like

It is well known that the construction of Voronoi diagrams requires time $\Omega(n \log n)$, already for the euclidean Voronoi diagram of point sites in the plane, see Section 2.1. But what if some more information about the Voronoi diagram is given in beforehand, e. g., if the sites already are sorted in some way. It turned out to be very difficult to answer this question, because the computation of Voronoi diagrams is a 2-dimensional problem.

Djidjev and Lingas [35] showed that if a set of point sites is sorted according to one coordinate, then one still needs time $\Omega(n \log n)$ to compute the Voronoi diagram. But if the point sites are vertices of a monotone histogram, given in sorted order, then it is possible to compute the Voronoi diagram in linear time.

Another way to attack this problem is to assume that the Voronoi diagram has some special structure, like a tree structure, which occurs, e. g., for points in convex position. Aggarwal et. al. [5] showed that if the ordering of the points along the convex hull is given, then their euclidean Voronoi diagram can be computed in linear time. The same technique applies for computing the Voronoi diagram of line segments forming a convex polygon, because these diagrams also have a tree structure. It equals the medial axis in the interior of the polygon, compare Section 2.2.4.

The natural question ist whether this technique can be generalized to abstract Voronoi diagrams, having a special structure. Klein and Lingas [50] showed that, given an unbounded Hamiltonian path passing through each Voronoi region exactly once for all diagrams of a subset S' of S, together with the ordering of the Voronoi regions along the path, the abstract Voronoi diagram can be computed in linear time. This also applies to Voronoi diagrams having a tree structure for all subsets S', and where the ordering of the unbounded regions at infinity is known.

In this chapter we go one step further and allow the AVD to be disconnected, thus having a forest structure for real subsets $S' \subsetneq S$. Now, given the ordering of the unbounded regions in $V(S)$, the diagram can still be computed in linear time, assuming that bisectors are unbounded and Voronoi regions are connected and nonempty.

An extended abstract of this chapter appeared at EuroCG'14 [18] and at CCCG'14 [19].

6.1 Introduction

General abstract Voronoi diagrams can not be computed faster than in $\Omega(n \log n)$ time. However, certain practical applications require only a specific substructure of the entire diagram or a special kind of Voronoi diagram. That is why it is worthwhile to consider special types of Voronoi diagrams and to come up with faster algorithms for their computation. Aggarwal et. al. [5] developed a linear-time algorithm for euclidean Voronoi diagrams of point-sites in convex position, see Figure 6.1. Their algorithm further allows to delete a site from euclidean Voronoi diagrams in time linear to the structural changes, and also speeds up the algorithm for the k^{th}-order Voronoi diagram in [57] by a $O(\log n)$ factor.

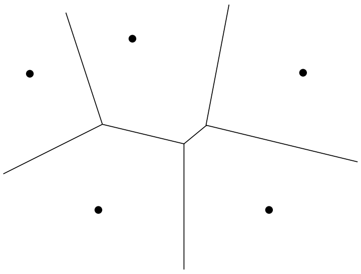


Figure 6.1: The euclidean Voronoi diagram of points in convex position is a tree.

Moreover, there are two kinds of Voronoi diagrams of a simple polygon, both consisting of a tree or forest structure, who have received considerable attention. First, the medial axis of a simple polygon is the Voronoi diagram of its polygonal edges. Lee [56] first proposed an $O(n \log n)$-time algorithm for this medial axis, and Chin et al. [27] later developed a linear-time algorithm. Second, the constrained Voronoi diagram of a simple polygon is the Voronoi diagram of its polygonal vertices constrained by its polygonal edges. Lee and Lin [58] first derived an $O(n \log n)$-time algorithm for these kinds of constrained Voronoi diagrams, and then Klein and Lingas [51] proposed a linear-time algorithm in the L_1 metric. Furthermore, in the euclidean metric, Klein and Lingas [52] later developed a randomized linear-time algorithm, and Chin and Wang [28] finally gave a deterministic linear-time algorithm.

However, for many other geometric objects and distance measures the convex position is not applicable. Besides, the linear-time algorithms [52, 27, 28] for the medial axis and the constrained Voronoi diagram of a simple polygon depend on a decomposition of a simple polygon, which prevents them from being extended to a more general setting.

The first approach towards abstract Voronoi diagrams was taken by Klein and Lingas [50] by defining the so called *Hamiltonian abstract Voronoi diagram* and proposing a linear-time algorithm. Here an unbounded Hamiltonian path is given, which for all subsets S' of S runs through each Voronoi region of $V(S')$ exactly once, see Figure 6.2. Furthermore, the usual axioms were required for each subset $S' \subseteq S$.

(A1) *By stereographic projection to the sphere, each curve $J(p,q)$ can be completed to a closed Jordan curve through the northpole.*

(A2) Each Voronoi region $V(p, S')$ is path-connected.

(A3) Each point of the plane belongs to the closure of a Voronoi region $\overline{VR(p, S')}$.

(A4) Any two curves $J(p, q)$ and $J(r, t)$ have only finitely many intersection points, and these intersections are transversal.

(A5) No nearest Voronoi region $VR(p, S')$ is empty.

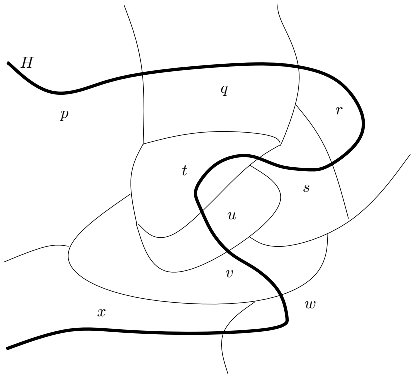


Figure 6.2: A Hamiltonian abstract Voronoi diagram. The sites are ordered $p, q, r, s, t, u, v, w, x$ along the Hamiltonian path H.

Axiom (A4) is assumed for technical reasons, it can be overcome as shown in [49] and axiom (A5) is needed, because otherwise by definition there would not exist a Hamiltonian path.

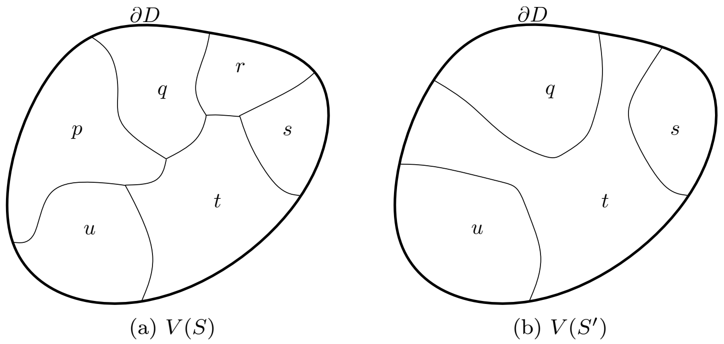
Unfortunately, computing a Hamiltonian path for a given AVD is NP-complete, which will be proven by Theorem 14 in Section 6.2. Therefore, let us now consider the abstract Voronoi diagram restricted to a domain D where its structure is a forest and each site has exactly one face, see Figure 6.3. Let $D \subseteq \mathbb{R}^2$ be a bounded domain, e. g., a domain bounded by Γ, where Γ is a simple closed curve intersecting each bisector exactly twice, such that no two bisectors intersect in a connected component entirely enclosed by the outer domain of Γ. In the following, without explicit indication, $V(S')$ means $V(S') \cap D$ and $\mathrm{VR}(p, S')$ means $\mathrm{VR}(p, S') \cap D$.

To be more specific, instead of requiring a Hamiltonian path, we have the additional axiom:

(A6) $V(S)$ is a tree and for or all $S' \subseteq S$, $V(S')$ is a forest and each Voronoi region has exactly one face.

This axiom implies that each bisector traverses D exactly once, any two related bisectors, $J(p, q)$ and $J(p, r)$ having one site in common, cross *at most* once in D and, together with axioms (A2) and (A5), ∂D runs through each Voronoi region of $V(S)$ *exactly* once. Based on our axioms we prove the following result.

Theorem 13. *Given a domain D together with the ordering of the Voronoi regions along ∂D we can compute $V(S)$ in time $O(n)$.*



(a) $V(S)$

(b) $V(S')$

Figure 6.3: (a) Abstract Voronoi diagram $V(S)$ in a domain D, the ordering of the regions along ∂D is p, q, r, s, t, u.
(b) For a subset $S' \subset S$, $V(S')$ may be a forest. Here, the ordering of the regions along ∂D is t, q, t, s, t, u.

On the other hand, if each bisector crosses D exactly once, no two related bisectors intersect in more than one point and each Voronoi region of $V(S)$ intersects the boundary of D, meaning that no region is completely contained in the interior of D and no region is empty, then we know that $V(S')$ is a forest for all subsets S' of S. $V(S)$ would be a tree only if in addition each Voronoi region intersects the boundary of D in exactly one component. If we would know in advance which regions of $V(S)$ are intersected by ∂D more than once, we would know how to separate $V(S)$ into trees and could adapt the theorem for each tree. Otherwise this would already be an element-uniqueness-test which would need time $\Omega(n \log n)$.

There is also a possibility of normalizing the bisector system in the sense that afterwards each pair of related bisectors cross *exactly* once, see Section 6.3. Then $V(S')$ would be a tree for all $S' \subseteq S$. But there are $\binom{n}{3}$ pairs of related bisectors, and none of these pairs must cross from the beginning. Thus it takes time $\Omega(n^3)$ to normalize them. And even afterwards ∂D may be a curve intersecting each region of $V(S)$ exactly once but it is unclear whether this is also true for subsets S' of S, because only related bisectors are claimed to cross exactly once.

That is why in this chapter we choose a different problem definition than in [50]. Compared to the algorithm for Hamiltonian abstract Voronoi diagrams our algorithm has two major differences in the coloring (Section 6.4.1) and selection (Section 6.4.2). For the coloring, our algorithm needs to consider two more subcases, and two consecutive sites in the sequence can be both colored red, while no consecutive sites in [50] are colored red. For the selection, our algorithm needs to modify $V(S')$ into a tree for applying Aggarwal's selecting lemma [5].

6.2 NP-completeness

A Hamiltonian path with respect to an AVD is a simple unbounded curve, homeomorphic to a line, that passes through each Voronoi region exactly once. We show that it is NP-complete to decide whether such a curve exists or not.

Let $V(S)$ denote an arbitrary AVD. In [48], Theorem 2.7.3, it has been shown that $V(S)$ together with the large curve Γ around the diagram is a biconnected planar graph with vertex-degree ≥ 3, the vertices on Γ are of degree 3. Also the opposite is true, namely each graph fulfilling these properties represents an AVD. In [48] a slightly different definition of AVD's was used, but it is easy to see that the Theorem is still true for our setting based on axioms (A1) to (A5). This means that the dual of $V(S)$, the dual of the graph structure of $V(S)$ inside Γ, is a biconnected planar graph and vice versa. Thus a Hamiltonian curve with respect to $V(S)$ is equivalent to a Hamiltonian path, with its endpoints on the outer face, in a biconnected planar graph.

To show that it is NP-hard to even decide the existence of such a path, we reduce the problem of deciding whether a Hamiltonian cycle exits in a Delaunay triangulation. In [33] it has been shown that this problem is NP-complete.

Lemma 58. *Let G be a biconnected planar graph. The problem to determine whether a Hamiltonian path P with endpoints on the outer face of G exists is NP-complete.*

Proof. It is clear that the problem is in NP. So, let G be a Delaunay triangulation. A Hamiltonian cycle C exists in G iff C visits each vertex v of G exactly once. Thus, if there exists a Hamiltonian path P with endpoints v and w such that there is an edge from v to w in G. Let v be an arbitrary vertex of G. Because G is a triangulation, there are $O(n)$ triangles adjacent to v. The boundary of each triangle consists of 3 vertices which are pairwise connected by an edge. If G contains a Hamiltonian cycle, then there must be a Hamiltonian path having its endpoints on one of the triangles. For each triangle T adjacent to v turn the graph inside out, such that T becomes the outer face and the outer face becomes a bounded face. The resulting graph G' is biconnected and planar. Now a Hamiltonian cycle exists in G iff for a triangle adjacent to v a Hamiltonian path, with its endpoints on the outer face, exists in G'. This proves the lemma. □

From this lemma we get our theorem.

Theorem 14. *It is NP-complete to determine if for a given abstract Voronoi diagram there exists an unbounded simple curve visiting each Voronoi region exactly once.*

Corollary 1. *For a given system of bisecting curves it is NP-comlete to determine if there exists an unbounded simple curve crossing each bisecting curve exactly once.*

Proof. An unbounded simple curve crossing each bisecting curve exactly once visits each Voronoi region in $V(S)$ exactly once, see Lemma 3 in [50]. □

6.3 Normalizing a Bisector System

In this section we show that it is possible to normalize a given bisector system, where each pair of bisectors cross at most once, such that afterwards each pair of bisectors cross exactly once and our axioms are still fulfilled.

Let $\{J(p,q) \mid p,q \in S\}$ be a system of bisecting curves in general position fulfilling axioms (A1) to (A5) and a relaxed version of (A6):

(A6') $V(S')$ is a forest for all $S' \subseteq S$.

Axiom (A6') implies that any two related bisectors cross *at most* once. Would all pairs of related bisectors cross *exactly* once then $V(S')$ would be a tree for all subsets S' of S. Let Γ be a closed curve encircling all intersection points of bisecting curves, such that each bisecting curve consists of two unbounded segments outside of Γ and a bounded segment inside.

Notation: We shall write $p|q$ to denote a segment of $J(p,q)$ that has $D(p,q)$ to its left and $D(q,p)$ to its right, if no confusion can arise.

Theorem 15. *By introducing crossings of neighboring bisecting curves outside of Γ (and afterwards enlarging Γ to include all new intersection points) we can transform $\{J(p,q) \mid p,q \in S\}$ into a* normal *system $\{J'(p,q) \mid p,q \in S\}$ of bisecting curves such that*

(i) axioms (A1) to (A5) and (A6') are fulfilled,

(ii) each pair of related bisectors cross exactly *once,*

(iii) $V(S) = V'(S) \cap int(\Gamma)$.

Property (ii) is equivalent to saying that the Voronoi diagram of any three sites contains exactly one Voronoi vertex. Namely, because of general position, we know that two related bisectors $J(q,p), J(p,r)$ can only cross in a point, where they intersect, and $J(q,r)$ must pass through this point, too. Such triplet crossing points correspond to Voronoi vertices in the diagrams of the three sites involved. From the original bisector system, all of these $\binom{n}{3}$ many crossings may be missing.

Proof. If the bisector system is not normal there exist three different sites p,q,r in S where $J(q,p), J(q,r), J(p,r)$ fail to cross but instead run as shown in Figure 6.4. By axiom (A5'), the Voronoi region of p in $V(S)$ extends to infinity, w.l.o.g. through the northern part of the strip (the southern part may be blocked by other sites in $V(S)$). Let m denote the number of unbounded southern bisector segments between $q|p$ and $p|r$. Clearly, $m \geq 1$ because $q|r$ is situated between $q|p$ and $p|r$. We call $q|p$ and $p|r$ a "strip of width m".

The theorem follows by applying the following lemma repeatedly.

Lemma 59. *If there is a strip of width $m \geq 1$ we can introduce another triplet crossing point while maintaining properties (i) to (iii).*

(We observe that the crossing point introduced need not be the one of the strip boundaries!)

Proof. By induction on m. If $m = 1$ then $q|p, q|r, p|r$ are direct neighbors and can be "braided" to obtain a triplet crossing point, v. We enlarge Γ to include v, and have obtained the new ordering $p|r, q|r, q|p$. Axioms (A1) to (A6') are still fulfilled.

Now let $m > 1$, and let $t|u$ be the right hand side neighbor of $q|p$, as shown in Figure 6.4. If $t|u = q|r$ then we are done with moving $q|p$ to the right, and start to move $p|r$ to the left towards $q|r$ in a symmetric way.

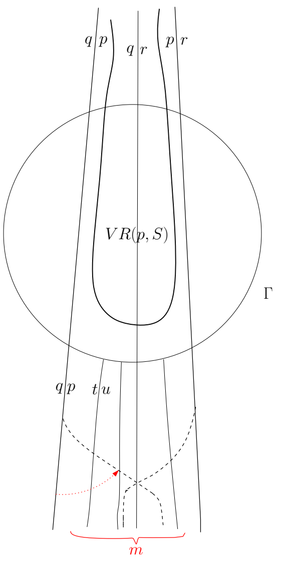


Figure 6.4: In principle we want to move $q|p$ and $p|r$ together such that they cross on $q|r$. This may require re-ordering the bisector segments in between, as we must not cause related bisectors to cross more than once.

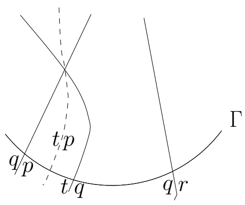


Figure 6.5: Illustration of the case $t|u = t|q$.

If t, u are different from q, p we can simply make $q|p$ cross $t|u$ without difficulty. This reduces m to $m - 1$, and the claim follows by induction. Otherwise, we analyze the following cases.

$t|u = p|u$. Impossible, because $q|u$ must appear between $q|p$ and $p|u$.

$t|u = t|p$. Here $t|p$ and $p|r$ form a strip of width $m-1$, so that induction applies.

$t|u = t|q$. We observe that $t \neq p$ because $q|p$ runs to the north. Similarly, we have $t \neq r$. Since the region of q is nonempty, $q|p$ and $t|q$ must intersect, as shown in Figure 6.5. But then $t|p$ must appear in between—a contradiction.

$t|u = q|u$. This case splits into three subcases, depending on the intersection behavior of $q|u$.

First, if $q|p$ and $q|u$ cross we have the situation shown in Figure 6.6. We observe that $q|u$ cannot cross $q|r$, too, because it would need to cross it twice in order to run to $q|p$. Since the region of p is unbounded to the north, $u|p$ and $p|r$ cannot cross, so $u|r$ must run between them to the north. If $u|r$ were situated, on the southern part of Γ, between $q|p$ and $q|u$, as shown in Figure 6.6, it could not run to the north because it could not cross $q|u$ (the crossing would have to lie on $q|r$, too, which is impossible as $q|u$ and $q|r$ are disjoint). Thus,

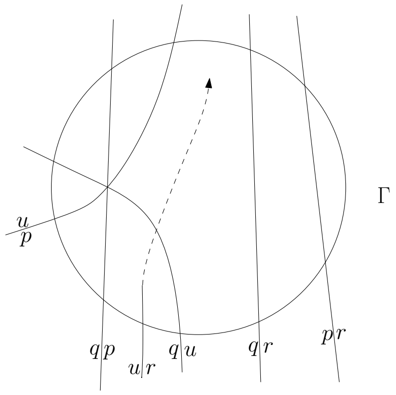


Figure 6.6: An impossible situation.

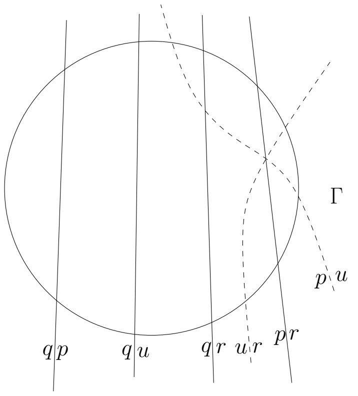


Figure 6.7: Bisectors $q|u$ and $u|r$ form a strip of width $< m$.

$u|r$ appears between $q|u$ and $p|r$, and $q|u$ and $u|r$ form a strip of width $< m$. Induction applies because the region of u is unbounded to the north.

Second, let us assume that $q|u$ intersects neither $q|p$ nor $q|r$, as shown in Figure 6.7. In the Voronoi diagram of p, q, u, bisector $q|u$ must run between $q|p$ and $p|u$ without crossing, because the region of p runs to the north. If, on the southern boundary of Γ, bisector $p|u$ appears to the left of $p|r$ then $q|p$ and $p|u$ form a strip of width $< m$, and induction applies. Otherwise, we have the situation depicted in Figure 6.7. Since $q|u$ and $q|r$ are disjoint by assumption, $q|r$ and $u|r$ cannot cross. Thus, $q|u$ and $u|r$ form a strip of width $< m$, and we can apply induction since the region of u is unbounded to the north.

Third, we assume that $q|u$ intersects $q|r$ but not $q|p$; see Figure 6.8. As in the previous case, bisector $q|u$ must run between $q|p$ and $p|u$ without crossing, and if $p|u$ appears to the left of $p|r$ we can apply induction to the strip formed by $q|p$ and $p|u$. Let us assume that $p|u$ is situated to the right of $p|r$. If $p|r$ and $p|u$ were disjoint, either the region of r or of u would be empty in the Voronoi diagram of p, r, u, because $r|u$ would run to the left of $p|r$; see Figure 6.9. Thus, there must be a crossing, as shown in Figure 6.8. But there is no way the two segments of $r|u$ can be connected, since multiple crossings are not allowed—a

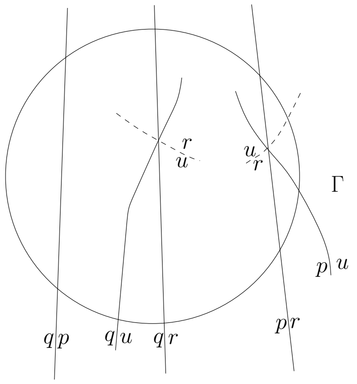


Figure 6.8: An impossible situation, since the two segments of $J(r,u)$ cannot be connected.

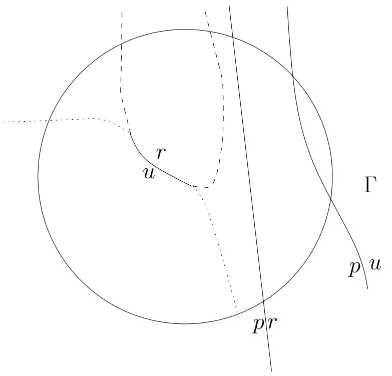


Figure 6.9: The Voronoi region of r or of u would be empty.

contradiction. This concludes the proof of the Lemma and of the Theorem. □

□

Unfortunately there are $\binom{n}{3}$ pairs of related bisectors and no pair of them must cross from the beginning. In this case all pairs have to be considered and it would take time $\Omega(n^3)$ to normalize the system.

Thus the algorithm discussed in the following does not require a normalized bisector system, it needs only axiom (A6).

6.4 The Algorithm

Let us get back to our actual problem. Here a bounded domain D together with the ordering of the Voronoi regions along its boundary is given and axioms (A1) to (A6) are required.

Definition 8. *For each set of sites $S' \subseteq S$ let $\pi(S')$ be the sequence of regions of $V(S')$ along ∂D. Since $V(S)$ partitions ∂D into $|\pi(S)|$ pieces, each element of $\pi(S)$ corresponds to a unique piece. For each element p of $\pi(S)$, $d(p)$ is a point on its corresponding piece.*

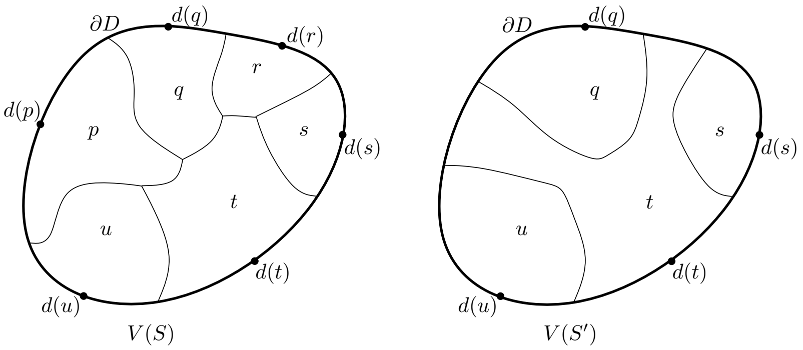


Figure 6.10: Illustration of Definition 8 and 9. Here $\pi(S) = (p, q, r, s, t, u)$, $\pi(S') = (t, q, t, s, t, u)$, and $\pi'(S') = (q, s, t, u)$.

Remark that $\mathrm{VR}(p, S) \subseteq \mathrm{VR}(p, S')$, thus $d(p) \in \mathrm{VR}(p, S')$ for all subsets S' of S, see Figure 6.10. Actually, $\pi(S)$ depends on the starting point and the direction of a traversal along ∂D. W. l. o. g., we assume the starting point is known and the direction is **clockwise**. Axiom (A6) implies that each element in $\pi(S)$ occurs only once. For subsets S' of S we have the following observation.

Lemma 60. *$\pi(S')$ is a Davenport-Schinzel-Sequence of order 2.*

Proof. By definition no element of the sequence appears twice without another site in between. So, suppose there are $p \neq q \in S'$ such that p, q, p, q occur in this ordering in $\pi(S')$. Then either the two p's or the two q's can not be connected in D, a contradiction to axiom (A2). □

We want to use a recursive algorithm to compute $V(S)$. To be able to recursively compute $V(S')$ from $V(S)$ it is important that the input, the sequence of sites $\pi(S')$, fulfills the same properties as the sequence $\pi(S)$. But $\pi(S)$ is a *Davenport-Schinzel-Sequence (DSS)* of order 1, whereas $\pi(S')$ may be a DSS of order 2. For this purpose we will use the following definition.

Definition 9. *Let $\pi'(S')$ be the subsequence of $\pi(S)$ containing all elements from S', i. e., $\pi'(S')$ is a DSS of order 1.*

In the example drawn in Figure 6.10, $\pi'(S')$ would be (q, s, t, u).

In the following we show that it indeed suffices to consider the subsequence $\pi'(S')$ in order to compute $V(S')$. Now our algorithm can be summarized as follows.

Algorithm

Input: The order $\pi(S)$ of the regions of $V(S)$ along the boundary of D.
Output: The Voronoi diagram within the domain D.

S1. Color each element of $\pi(S)$ either *blue* or *red*, i.e., π is partitioned into $\pi'(B)$ and $\pi'(R)$, and S is partitioned into B and R, such that both $|B|$ and $|R|$ are a constant fraction of $|S|$, and for each two consecutive red sites, r_1, and r_2, in π, $\mathrm{VR}(r_1, B \cup \{r_1, r_2\})$ and $\mathrm{VR}(r_2, B \cup \{r_1, r_2\})$ are not adjacent. See Section 6.4.1 for details.

S2. Compute $V(B)$ from $\pi'(B)$ recursively.

S3. Select a subset C from R such that $|C|$ is a constant fraction of $|R|$, and for any two sites, c_1 and c_2, $\mathrm{VR}(c_1, B \cup \{c_1, c_2\})$ and $\mathrm{VR}(c_2, B \cup \{c_1, c_2\})$ are not adjacent. See Section 6.4.2 for details.

- Add artificial Voronoi edges to $V(B)$ to obtain a tree structure $V^*(B)$.
- Apply Aggarwal et al.'s selecting Lemma [5] on $V^*(B)$.

S4. Compute $V(B \cup C)$ by sequentially inserting each element of C into $V(B)$.

S5. Compute $V(G)$ from $\pi'(G)$ recursively, where $G = R \setminus C$ and $\pi'(G)$ is obtained from $\pi'(R)$ by removing all elements in C.

S6. Merge $V(B \cup C)$ and $V(G)$.

Step 1 can be carried out in linear time according to Section 6.4.1, Step 3 and Step 4 can be completed in linear time according to Section 6.4.2 and 6.4.3, and Step 6 can be implemented in linear time using the general merge method described in [48]. Since $|B|$ and $|G|$ are both a constant fraction of $|S|$, the above claims conclude Theorem 13.

In the following we will make use of two more definitions.

Definition 10. *For a set S of sites, a subset S' of S, and a site p of S', a connected intersection between $VR(p, S')$ and ∂D is* redundant *if it does not contain the connected intersection between $VR(p, S)$ and ∂D. From the viewpoint of $\pi'(S')$, which is a subsequence of $\pi(S)$, a connected intersection between $VR(p, S')$ and ∂D is* redundant *if it does not contain $d(p)$ for p in $\pi'(S')$.*

In Figure 6.10 the two intersections of $\mathrm{VR}(t, S')$ with ∂D between the regions of u and q and between q and s are redundant.

Definition 11. *For all $S' \subseteq S$, a pqr-vertex of $V(S')$ is a Voronoi vertex adjacent to $VR(p, S')$, $VR(q, S')$, and $VR(r, S')$ clockwise. If $VR(p, S')$ is the only region bordering on $VR(q, S')$, $VR(p, S')$ encloses $VR(q, S')$, for brevity we say p* encloses *q* in *$V(S')$, compare Figure 6.11.*

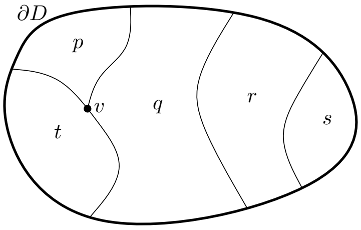


Figure 6.11: Vertex v is a pqt-vertex and r encloses s.

6.4.1 Red-Blue-Coloring Scheme

The red-blue-coloring scheme consists of two steps, where a site is blue as long as it is not colored red. See also Figure 6.12.

1. For every 5 consecutive sites along $\pi(S)$, (l, m, p, q, r), p is colored *red* if one of the following conditions holds. Let T be $\{l, m, p, q, r\}$.

   (i) There is an mpq-vertex in $V(T)$.

   (ii) $\mathrm{VR}(m, T)$ encloses $\mathrm{VR}(p, T)$.

   (iii) $\mathrm{VR}(q, T)$ encloses $\mathrm{VR}(p, T)$.

2. For every 3 consecutive sites along $\pi(S)$ that are all blue, the middle one is colored *red*.

Let R be the set of red sites, and B be the set of blue sites. Observe that the final diagram $V(S)$ is a tree, but in the recursion $V(S)$ may be a forest, e. g., if $S = B$. Then we use the sequence $\pi'(S)$ instead of $\pi(S)$.

Lets first observe that not too many consecutive sites are colored red by our coloring scheme.

Lemma 61. *No 3 consecutive sites in $\pi(S)$ are all colored red.*

Proof. For the sake of a contradiction assume that three consecutive sites r_1, r_2, r_3 are all red. Let s_1 and s_2 be the two consecutive sites previous to r_1, and s_3 and s_4 the two consecutive sites after r_3. By definition r_1, r_2 and r_3 can not be colored red by Step 2. Thus we need to consider only Step 1. There are three different cases for r_1 to be colored red.

Case 1: There is an $s_2r_1r_2$-vertex in $V(\{s_1, s_2, r_1, r_2, r_3\})$. This vertex is still an $s_2r_1r_2$-vertex in $V(\{s_2, r_1, r_2, r_3\})$ implying that there can not exist an $r_1r_2r_3$-vertex in $V(\{s_2, r_1, r_2, r_3\})$ and hence also no $r_1r_2r_3$-vertex in $V(\{s_2, r_1, r_2, r_3, s_3\})$. This means that r_2 must be colored red because r_1 or r_3

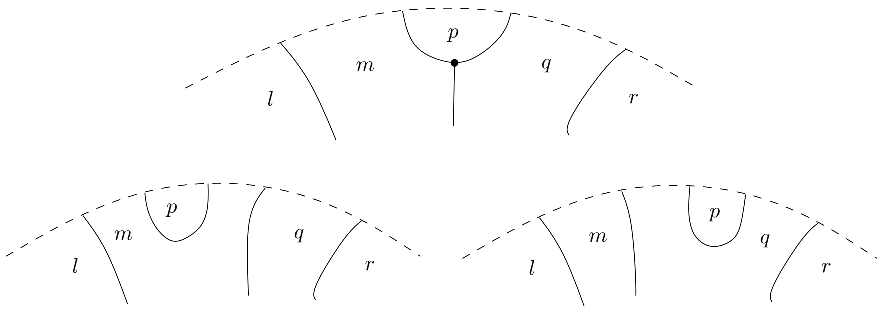


Figure 6.12: Three cases where p is colored red.

encloses it in $V(\{s_2, r_1, r_2, r_3, s_3\})$. But then r_2 can not be adjacent to a vertex in $V(\{s_2, r_1, r_2, r_3\})$, i.e., no $s_2r_1r_2$ vertex exists, a contradiction.

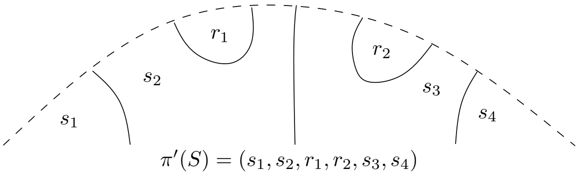
Case 2: r_1 is colored red because s_2 encloses r_1 in $V(\{s_1, s_2, r_1, r_2, r_3\})$. But then the regions of r_1 and r_2 are not adjacent in $V(\{s_2, r_1, r_2, r_3\})$ and there can be no $r_1r_2r_3$-vertex in $V(\{s_2, r_1, r_2, r_3, s_3\})$. Further r_1 can not enclose r_2. This means that r_2 must be colored red because r_3 encloses it in $V(\{s_2, r_1, r_2, r_3, s_3\})$. But then there can be no $r_2r_3s_3$-vertex in $V(\{r_1, r_2, r_3, s_3, s_4\})$ and r_3 can not be enclosed by r_2 or s_3 in $V(\{r_1, r_2, r_3, s_3, s_4\})$. Thus r_3 is not colored red, a contradiction.

Case 3: r_1 is enclosed by r_2 in $V(\{s_1, s_2, r_1, r_2, r_3\})$ but then because of the same reasons as in Case 2 r_2 is not colored red. □

It can happen that two consecutive sites are both colored red, but fortunately if this happens, their regions can not be adjacent as will be shown in the following lemma. Compare also Figure 6.13.

Corollary 2. *Let $s_1, s_2, r_1, r_2, s_3, s_4$ be 6 consecutive sites in $\pi(S)$. If r_1 and r_2 are both red, then s_2 and s_3 are both blue. Further s_2 encloses r_1 in $V(\{s_1, s_2,$ $r_1,$ $r_2, s_3\})$ and s_3 encloses r_2 in $V(\{s_2, r_1, r_2, s_3, s_4\})$. In particular s_2 encloses r_1 and s_3 encloses r_2 in $V(\{s_2, r_1, r_2, s_3\})$.*

Proof. Lemma 61 shows that s_2 and s_3 are both red. Case two in the proof of Lemma 61 is the only case where two consecutive sites r_1 and r_2 are both colored red. Here s_2 encloses r_1 in $V(\{s_1, s_2, r_1, r_2, s_3\})$ and r_3 encloses r_2 in $V(\{s_2, r_1, r_2, s_3, s_4\})$, implying that s_2 encloses r_1 and s_3 enclose r_2 in $V(\{s_2,$ $r_1,$ $r_2,$ $s_3\})$. □



$\pi'(S) = (s_1, s_2, r_1, r_2, s_3, s_4)$
$\pi(S) = (s_1, s_2, r_1, s_2, s_3, r_2, s_3, s_4)$

Figure 6.13: Two consecutive sites r_1 and r_2 in $\pi'(S)$ are both colored red, but their regions are not adjacent.

The property from above can be generalized to any two consecutive red sites, even if there are one or two blue sites in between, as the next lemma will say.

Lemma 62. *Let r_1 and r_2 be two consecutive red sites. Then $VR(r_1, B \cup \{r_1, r_2\})$ and $VR(r_2, B \cup \{r_1, r_2\})$ are not adjacent.*

Proof. Let s_1 be the site previous to r_1 and s_2 the site after r_2 in π. There are three cases.

Case 1: There is no blue site between r_1 and r_2. Because of Corollary 2, s_1 and s_2 are both blue and s_1 encloses r_1 and s_2 encloses r_2 in $V(s_1, r_1, r_2, s_2)$. Thus it follows directly that the regions of r_1 and r_2 can not be adjacent in $V(B \cup \{r_1, r_2\})$.

Case 2: There is exactly one blue site b between r_1 and r_2. For the sake of a contradiction suppose $\mathrm{VR}(r_1, B \cup \{r_1, r_2\})$ and $\mathrm{VR}(r_2, B \cup \{r_1, r_2\})$ are adjacent. Then the regions of r_1 and r_2 are the only regions that may be adjacent to the region of b in $V(B \cup \{r_1, r_2\})$. If they both are adjacent to the region of b, then there is a $r_1 b r_2$-vertex in $V(B \cup \{r_1, r_2\})$. If only the region of r_1 is adjacent to the region of b, then r_1 encloses b in $V(B \cup \{r_1, r_2\})$ and if only the region of r_2 is adjacent to the region of b, then r_2 encloses b in $V(B \cup \{r_1, r_2\})$.

Now if s_1 and s_2 are both blue, then $\{s_1, r_1, b, r_2, s_2\} \subseteq B \cup \{r_1, r_2\}$ and b would have been colored red, a contradiction to the assumption that b is blue.

Now assume s_1 is red and let s_0 be the predecessor of s_1. Corollary 2 tells us that b encloses r_1 in $V(\{s_0, s_1, r_1, b, r_2\})$, but then the regions of r_1 and r_2 can not be adjacent in $V(B \cup \{r_1, r_2\})$. The case that s_2 is red is symmetric.

Case 3: There are exactly two blue sites b_1 and b_2 between r_1 and r_2. As in case 2, for the sake of a contradiction suppose $\mathrm{VR}(r_1, B \cup \{r_1, r_2\})$ and $\mathrm{VR}(r_2, B \cup \{r_1, r_2\})$ are adjacent. Then the regions of r_1, r_2, b_1 and b_2 are the only regions that may be adjacent to the regions of b_1 and b_2 in $V(B \cup \{r_1, r_2\})$.

Now there are two subcases:

Case 3.1. The region of b_1 or b_2 is not adjacent to any of the regions of r_1 or r_2 in $V(B \cup \{r_1, r_2\})$. Then b_1 encloses b_2 or b_2 encloses b_1 in $V(B \cup \{r_1, r_2\})$. W.l.o.g. let b_1 enclose b_2, the other case is symmetric. If s_2 is blue, then b_1 also encloses b_2 in $V(\{r_1, b_1, b_2, r_2, s_2\})$, and b_2 must be colored red. But if s_2 is red, then by Corollary 2 b_2 has to enclose r_2 in $V(\{b_1, b_2, r_2, s_2)\})$, both a contradiction.

Case 3.2. Both the regions of b_1 and b_2 are adjacent to the region of r_1 or r_2. Because all Voronoi regions are connected the region of b_1 or b_2 is adjacent to the region of r_1 but not r_2 or vice versa. Let b_1 be adjacent to the region of only r_1 and b_2, the other 3 constellations are symmetric. Then there is a $r_1 b_1 b_2$-vertex v in $V(B \cup \{b_1, b_2\})$. If s_1 is blue, then v is also a $r_1 b_1 b_2$-vertex in $V(\{s_1, r_1, b_1, b_2, r_2\})$ and thus b_1 would be red, a contradiction.

If s_1 is red, then r_1 would be enclosed by b_1 in $V(\{s_0, s_1, r_1, b_1, b_2\})$, a contradiction as in Case 2. □

The definition of the coloring scheme (Step 2) ensures that the number of red sites is at least a constant fraction of $|S|$ and Corollary 2 shows that also the number of blue sites is at least a constant fraction of $|S|$. Together with the previous lemma this shows that (S1) of the Algorithm is satisfied.

6.4.2 Choosing Crimson Sites

The next step is to choose a set of crimson sites out of the set of red sites. To satisfy (S3) from the algorithm, we want to apply the following combinatorial lemma from [5], which helps us to obtain a set of crimson sites whose regions are not adjacent.

Lemma 63. *Let T be an unrooted binary tree, embedded in the plane. Assume that for each leaf l of T a subtree T_l rooted at l is defined, such that*

1. *given leaf l, one can in constant time decide if its parent node belongs to T_l,*

2. *if l, l' are consecutive leaves in the topological ordering around T, then T_l and $T_{l'}$ are disjoint.*

Then it is possible to find, in time $O(|T|)$, at least $g|T|$, $0 < g < 1$, many leaves whose subtrees have a pairwise edge distance greater than one.

The only concern is that our diagram $V(B)$, generated by the blue sites, is rather a forest than a tree. But as will be seen we can modify it by adding some edges and leaves to obtain a tree $V^*(B)$ fulfilling the claimed properties. We start with the following observation.

Lemma 64. *We can detect all redundant intersections of $V(B)$ in time $O(n)$.*

Proof. Because $V(B)$ is already computed, we know the sequence $\pi(B)$. The sequence $\pi'(B)$ is obtained by deleting all sites from R from $\pi(S)$. This takes time $O(n)$.

Recall that $\pi(B)$ is the sequence of sites along ∂D in $V(B)$. This is a Davenport-Schinzel-Sequence (DSS) of order 2, whereas $\pi'(B)$ is a DSS of order 1. Let $|B| = m \leq n$, $\pi'(B) = (p_1, \dots, p_m)$ and $\pi(B) = (q_1, \dots, q_l)$, where $q_1 = p_1$ refers to a non redundant intersection. Further $l \leq 2m - 1$, because $\pi(B)$ is a DSS of order 2.

Let $\pi(x)$ be the site at position x in $\pi(B)$, let $\pi(i) = p_j$ refer to the non redundant intersection of p_j and let i' be the smallest index greater i with $\pi(i') = p_{j+1}$. We claim that $\pi(i')$ refers to the non redundant intersection of p_{j+1}, implying that all $\pi(k)$ for $i < k < i'$ refer to redundant intersections, compare Figure 6.14.

Suppose $\pi(i') = p_{j+1}$ refers to a redundant intersection. Then there must be a $\pi(l) = p_{j+1}$, $l > i' + 1$ referring to the non redundant intersection of p_{j+1}, which implies that all $\pi(k)$ with $i' < k < l$ refer to redundant intersections. Let $\pi(k) \neq p_{j+1}$, $i' < k < l$, then there must be $k' < i'$ or $k' > l$ with $\pi(k') = \pi(k)$ referring to the non redundant intersection of $\pi(k)$, contradicting that $\pi(B)$ is a DSS or order 2.

This shows that by simultaneously walking one time along $\pi(B)$ and $\pi'(B)$, we can detect all non redundant and redundant intersections, which thus takes time $O(l) \in O(m) \in O(n)$. □

Now we construct the tree $V^*(B)$ out of the forest $V(B)$ by the following operations, compare Figure 6.15.

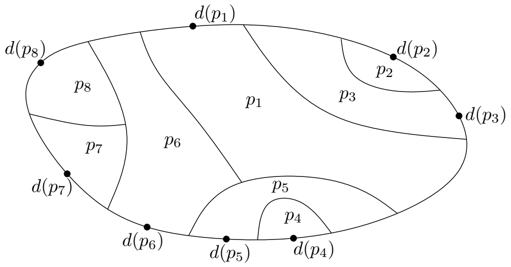


Figure 6.14: $\pi(B) = (p_1, p_3, p_2, p_3, p_1, p_5, p_4, p_5, p_6, p_7, p_8, p_6)$ and $\pi'(B) = (p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8)$. The redundant intersections of $V(B)$ are $\pi(2), \pi(5), \pi(6)$, and $\pi(12)$.

(i) For all redundant intersections on ∂D link the two leaves bounding it along ∂D, see the fat edges in Figure 6.15.
If a redundant intersection borders another redundant intersection on its right (left) end, then let the leaf between them now be a vertex in $V^*(B)$, see the vertex between the regions of t and z in Figure 6.15. Observe that this is a vertex of degree 3. Otherwise link the right (left) end of the redundant intersection to $V(B)$, deleting the vertex in between.

Next we attach some leaves to $V^*(B)$ outside of D such that between each pair of consecutive blue sites b_i and b_{i+1} having one (or two) red site(s) in between, there is exactly one (or between one and two) leaves in $V^*(B)$. If there is no red site between b_i and b_{i+1} there is also no leaf.

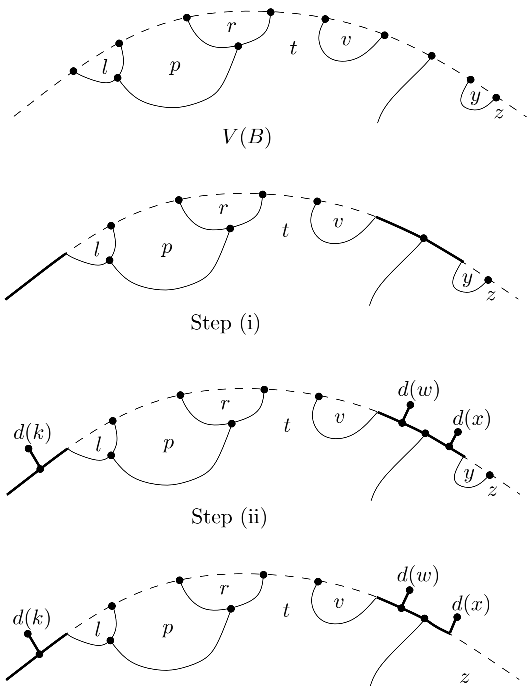
(ii) If there are one or two red sites r_1 and r_2 between two consecutive blue sites b_i and b_{i+1} but no leaf between them, then there is a connected set of redundant intersections between b_i and b_{i+1}. If $d(r_j)$, $j = 1, 2$ lies within this sequence we attach a leaf to $V^*(B)$ at $d(r_j)$, otherwise if $d(r_j)$ lies to the left (right) of the sequence we attach a leaf at the leftmost (rightmost) point of the sequence. If both $d(r_j)$, $j = 1, 2$ are to the left (right) of the redundant intersection sequence, only one edge and leaf is attached at the leftmost (rightmost) point, compare $d(k), d(w), d(x)$ in Figure 6.15.

Between two consecutive blue sites in $\pi(S)$ there are at most two red site. Thus for every connected sequence of redundant intersections at most two leaves are attached. Further, between each pair of consecutive blue sites separated by one or two red sites there is now at least one leaf.

(iii) If there is a leaf in $V(B)$ between two consecutive blue sites b_i and b_{i+1}, which are not separated by a red site, it is contracted out, see the edge and leaf between the regions of y and z in Figure 6.15. Also all vertices of degree 2 are deleted and the two adjacent edges are linked to each other.

The following lemma now proves that the constructed graph $V^*(B)$ fulfills the desired.

Lemma 65. *$V^*(B)$ is a binary tree and can be constructed in time $O(n)$.*



$V(B)$

Step (i)

Step (ii)

Step (iii): $V^*(B)$

Figure 6.15: Illustration of the construction of $V^*(B)$ out of $V(B)$, where $\pi(S) = (k, l, m, p, q, r, s, t, u, v, w, x, y, z, \ldots)$, $\pi'(B) = (l, p, r, t, v, y, z, \ldots)$, and $\pi(B) = (t, l, p, r, t, v, t, z, y, z, \ldots)$. Fat edges indicate redundant intersections and new leaves.

Proof. It is clear that $V^*(B)$ is a forest. So assume it is disconnected. Then there is a site $b \in B$ whose Voronoi region in $V(B)$ intersects ∂D in more than one component. By (i) all these components are non redundant. But then b has to appear several times in $\pi(S)$, a contradiction. Definitions (i) to (iii) imply that all internal nodes of $V^*(B)$ are of degree 3.

By lemma 64 we can detect all redundant intersections in time $O(n)$. In the same time operation (i) can be accomplished. For operation (ii) and (iii) we have to walk once around ∂D and look at consecutive blue sites.

For each pair of consecutive blue sites b_i and b_{i+1} we test if there are zero, one, or two red sites in between. If there is no red site between b_i and b_{i+1} but a leaf, we prune the leaf in constant time. If there are one or two red sites r_1 and r_2 but no leaf between b_i and b_{i+1} we test if $d(r_1)$ and $d(r_2)$ lie to the left, within or to the right of the redundant intersection sequence between the two blue sites and attach one or two leaves like described in (ii). For each redundant intersection this takes constant time and there are $O(n)$ redundant intersections altogether. □

Now that we know how to obtain a tree $V^*(B)$ out of $V(B)$, we can explain how Lemma 63 can be applied in order to choose a fixed fraction of crimson sites out of the set of red sites, such that no pair of crimson regions are adjacent in the blue diagram.

If two blue sites b_i and b_{i+1} are separated by a red site r in $\pi(S)$ but the leaf between them is not contained in $\mathrm{VR}(r, B \cup \{r\})$, then r is enclosed by the region of b_i (or b_{i+1}) in $V(B \cup \{r\})$. In this case color r crimson *with respect to* b_i (b_{i+1}) and if the leftmost (rightmost) leaf between b_i and b_{i+1} is not contained in the region of a consecutive site, associate with r the subtree containing only this leaf. If two red sites are between b_i and b_{i+1} and both are colored crimson because of b_i (b_{i+1}) associate only one of them with the leftmost (rightmost) leaf.

Up to now we may already have colored some red sites crimson. To make sure we obtain a fixed fraction of crimson sites we apply lemma 63 in the following way. For each leaf l of $V^*(B)$ contained in a red region $\mathrm{VR}(r, B \cup \{r\})$ define T_l by the subtree spanned by all vertices of $V^*(B)$ contained in $\mathrm{VR}(r, B \cup \{r\})$. The next lemma shows that this is possible.

Lemma 66. *Let r be a red site. If $VR(r, B \cup \{r\})$ intersects a leaf of $V^*(B)$, then $VR(r, B \cup \{r\}) \cap V^*(B)$ is connected. Otherwise it is empty.*

Proof. Suppose $\mathrm{VR}(r, B \cup \{r\})$ intersects a leaf of $V^*(B)$ and $\mathrm{VR}(r, B \cup \{r\}) \cap V^*(B)$ is not connected. Then $\mathrm{VR}(r, B \cup \{r\})$ would disconnect the region of a blue site in $V(B \cup \{r\})$, a contradiction.

If $\mathrm{VR}(r, B \cup \{r\})$ does not intersect a leaf of $V^*(B)$ it must be contained within a single blue region of $V(B)$, thus it can not intersect $V^*(B)$. □

Now Lemma 62 and 63 imply the requested property which shows that (S3) of the algorithm is satisfied.

Lemma 67. *A fixed fraction of sites C are colored crimson and no regions of two crimson sites are adjacent in $V(B \cup C)$.*

Proof. To test whether the parent node of a leaf is contained in the region of a red site it is enough to consider the diagram of the three sites adjacent to the node and the red site. Thus this test can be done in constant time. Further each leaf of $V^*(B)$ is associated with the region of a red site. This ensures a correct application of Lemma 63 and finishes the proof. □

6.4.3 Insertion of Crimson Sites

It remains to clarify how to insert the crimson sites into $V(B)$ in order to obtain $V(B \cup C)$, (S4). Let c be a crimson site between the two blue sites b_i and b_{i+1} in the sequence. If the region of c does not intersect a leaf of $V^*(B)$, then we know that it is enclosed by the region of a blue site b_i or b_{i+1}. Let it be b_i, then we just have to insert the part of the bisector $J(c, b_i)$ contained in $\mathrm{VR}(b_i, B)$ as a new edge in $V(B \cup C)$. The other crimson sites can be inserted along the subtrees of $V^*(B)$ associated with them. Thus also the insertion takes time $O(n)$.

Altogether we have now proven our theorem.

Theorem 16. *Given a domain D together with the ordering of the Voronoi regions along ∂D we can compute $V(S)$ in time $O(n)$.*

6.5 Discussion

A natural question is whether it is possible to relax our axioms and still have a linear time algorithm for computing the Voronoi diagram. There exist applications where Voronoi regions restricted to the domain D are disconnected. This is something that can happen for the farthest Voronoi diagram of line segments or when the domain D corresponds to a Voronoi region which is to be deleted from a given Voronoi diagram.

Chapter 7

Conclusion and Open Problems

In this thesis we have been able to study wide classes of abstract Voronoi diagrams, which have not been considered before. We managed to analyze them and find efficient algorithms to compute them. But still there are concrete Voronoi diagrams not covered by our concept. Bisectors may have a more complicated shape, e. g., for crossing line segments bisectors may consist of four halflines meeting at the crossing point. For special cases of the L_1-, L_∞-, and Karlsruhe-metric, bisectors may be even two dimensional, see Section 2.2.1 and 2.2.5. The usual way to overcome these difficulties is to assume general position, but this may not always be applicable.

And if bisectors are two dimensional, is there a way to modify them by assigning portions of the two dimensional areas to each dominance region in a suitable way, such that our axioms are fulfilled afterwards? More precisely, after modifying the bisectors no Voronoi region should consist of more connected components than before and the closures of the Voronoi regions should now cover the whole plane, i. e., the Voronoi diagram should consist of one dimensional edges and zero dimensional vertices.

Furthermore, let us consider the example described in Section 3.1 once more. Here each point site sends out an expanding circles and conquers the area where its circle arrives first. The conquered area becomes its Voronoi region. If no circle ever overruns another one, then all bisectors are unbounded, otherwise they are closed and may even consist of several closed curves. This is something not covered by the setting of Chapter 4. The problem is that now a Voronoi region may consist of several islands surrounded by different other regions and the presented algorithm would not always be able to detect all these islands when inserting a new region to the current diagram. Here a new idea is needed.

Also the example from Section 3.1 itself is in general not very well analyzed yet. What is the complexity of such Voronoi diagrams? How many intersections are there between any two bisectors, and can we use our algorithms to compute them efficiently?

Finally, there remain some implementation issues. Throughout this thesis we make use of some basic operations, like computing the intersections between two bisectors, to test if x or y comes first on a bisector for two given points x and

y, to decide if x is in $D(p, q)$, $J(p, q)$, or $D(q, p)$, and to compute the diagram of 5 sites. We claim that these operations can be performed in constant time (resp. in time $O(s)$ if two bisectors intersect in $O(s)$ points). But how do we have to provide the bisector system such that this really can be achieved?

List of Figures

Bibliography

[1] M. Abellanas, F. Hurtado, and B. Palop. Transportation Networks and Voronoi Diagrams. Proceedings International Symposium on Voronoi Diagrams in Science and Engineering, 2004.

[2] P. Agarwal. Personal Communication, 2012.

[3] P.K. Agarwal, J. Pach and M. Sharir. State of the union, of geometric objects. Proc. Joint Summer Research Conf. on Discrete and Computational Geometry: 20 Years Later. Contemp. Math. 452, AMS, pp. 9–48, 2008.

[4] P. Agarwal, O. Schwarzkopf, and M. Sharir. Overlay of Lower Envelopes and its Applications. Report CS-1994-18, Department of Computer Science, Duke University, 1994.

[5] A. Aggarwal, L. J. Guibas, J. Saxe, and P. W. Shor. A linear-time algorithm for computing the Voronoi diagram of a convex polygon. Discrete and Computational Geometry 4, pp. 591–603, 1989.

[6] H.-K. Ahn, O. Cheong, and R. v. Oostrum. Casting a Polyhedron with Directional Uncertainty. Computational Geometry: Theory and Applications 26(2), pp. 129–141, 2003.

[7] O. Aichholzer, F. Aurenhammer, and B. Palop. Quickest Paths, Straight Skeletons, and the City Voronoi Diagram. Discrete and Computational Geometry 31(7), pp. 17–35, 2004.

[8] N. Alon and E. Györi. The number of Small Semispaces of a Finite Set of Points in the Plane. Journal of Combinatorial Theory, Ser. A 41(1), pp. 154–157, 1986.

[9] N.M. Amato, M.T. Goodrich and E.A. Ramos. A Randomized Algorithm for Triangulating a Simple Polygon in Linear Time. Discrete & Computational Geometry, pp. 245–265, 2001.

[10] F. Aurenhammer. Voronoi Diagrams: A Survey of a Fundamental Geometric Data Structure. ACM Computing Surveys 23(3), pp. 345–405, 1991.

[11] F. Aurenhammer, R.L.S. Drysdale, and H. Krasser. Farthest Line Segment Voronoi Diagrams. Information Processing Letters 100, pp. 220–225, 2006.

[12] F. Aurenhammer and H. Edelsbrunner. An Optimal Algorithm for Constructing the Weighted Voronoi Diagram in the Plane. Pattern Recognition, Vol. 17, No. 2, pp. 251–257, 1983.

[13] F. Aurenhammer and R. Klein. Voronoi Diagrams. In: J.R. Sack and G. Urrutia (Eds.), Handbook on Computational Geometry, Elsevier, pp. 201–290, 1999.

[14] F. Aurenhammer, R. Klein, and D.-T. Lee. Voronoi Diagrams and Delaunay Triangulations. World Scientific Publishing Company, 2013.

[15] S. W. Bae and K.-Y. Chwa. Voronoi Diagrams for a Transportation Network on the Euclidean Plane. International Journal on Computational Geometry and Applications 16, pp. 117-144, 2006.

[16] C. Bohler, P. Cheilaris, R. Klein, C.H. Liu, E. Papadopoulou, and M. Zavershynskyi. On the Complexity of Higher Order Abstract Voronoi Diagrams. Proc. International Colloquium on Automata Languages and Programming, pp. 208-219, 2013. *To appear in:* Computational Geometry: Theory and Applications.

[17] C. Bohler, and R. Klein. Abstract Voronoi Diagrams with Disconnected Regions. Proc. 24th International Symposium on Algorithms and Computation, pp. 306-316, 2013. *To appear in:* International Journal of Computational Geometry & Applications, special issue on ISAAC 2013.

[18] C. Bohler, R. Klein, and C.-H. Liu. Forest-Like Abstract Voronoi Diagrams in Linear Time. Proc. 30th European Workshop on Computational Geometry, Ein-Gedi, Israel, 2014.

[19] C. Bohler, R. Klein, and C.-H. Liu. Forest-Like Abstract Voronoi Diagrams in Linear Time. Proc. 26th Canadian Conference on Computational Geometry, Halifax, Canada, 2014.

[20] C. Bohler, C.H. Liu, E. Papadopoulou, and M. Zavershynskyi. A Randomized Divide and Conquer Algorithm for Higher-Order Abstract Voronoi Diagrams. Proc. of the 25th International Symposium on Algorithms and Computations, pp. 27–37, Jeonju, Korea, 2014.

[21] J. D. Boissonnat, O. Devillers, R. Schott, M. Teillaud, and M. Yvinec. Applications of Random Sampling to On-line Algorithms in Computational Geometry. Discrete & Computational Geometry 8, 1992, 51-71.

[22] J. D. Boissonnat, C. Wormser, and M. Yvinec. Curved Voronoi Diagrams. In: J. D. Boissonnat and M. Teillaud (Eds.), Effective Computational Geometry for Curves and Surfaces. Springer, Mathematics and Visualization, 2006.

[23] B. Chazelle. Triangulating a Simple Polygon in Linear Time. Discrete and Computational Geometry, 6, pp. 485–524, 1991.

[24] O. Cheong, H. Everett, M. Glisse, J. Gudmundsson, S. Hornus, S. Lazard, M. Lee, and H-S. Na. Farthest-Polygon Voronoi Diagrams. Computational Geometry: Theory and Applications, 44 (4), pp. 234–247, 2011.

[25] D. Cheriton, and R.E. Tarjan. Finding Minimum Spanning Trees. SIAM Journal on Computing 5:724-742, 1976.

[26] L.P. Chew, and R.L.S. Drysdale. Voronoi Diagrams Based on Convex Distance Functions. Proc. 1st Ann. ACM Symposium on Computational Geometry, pp. 235-244, 1985.

[27] F. Chin, J. Snoeyink, and C. A. Wang. Finding the medial axis of a simple polygon in linear time. Discrete Computational Geometry 21, pp. 405–420, 1999.

[28] F. Chin and C. A. Wang. Finding the constrained Delaunay triangulation and constrained Voronoi diagram of a simple polygon in linear time. SIAM Journal on Computing 28(2), pp. 471–486, 1998.

[29] K. Clarkson, K. Mehlhorn and R. Seidel. Four Results on Randomized Incremental Constructions. Computational Geometry: Theory and Applications 3, pp 185-212, 1993.

[30] K. Clarkson and P. Shor. Applications of Random Sampling in Computational Geometry, II. Discrete and Computational Geometry 4, pp. 387-421, 1989.

[31] A.G. Corbalan, M. Mazon and T. Recio. Geometry of Bisectors for Strictly Convex Distance Functions. International Journal of Computational Geometry and Applications 6 (1), pp 45-58, 1996.

[32] R. Descartes. Principia Philosophiae. Ludovicus Elzevirius, Amsterdam, 1644.

[33] M. Dillencourt. Finding Hamiltonian Cycles in Delaunay triangulations is NP-complete. Discrete Applied Mathematics 64, pp. 207-217, 1996.

[34] P.G.L. Dirichlet. Über die Reduction der positiven quadratischen Formen mit drei unbestimmten ganzen Zahlen. Journal für die reine und angewandte Mathematik 40, 209–227, 1850.

[35] H.N. Djidjev and A. Lingas. On Computing Voronoi Diagrams for Sorted Point Sets. International Journal of Computational Geometry and Applications, Volume 05, 327, 1995.

[36] D.P. Dobkin, and R.J. Lipton. Multidimensional Searching Problems. SIAM Journal on Computing 5, pp 181-186, 1976.

[37] H. Edelsbrunner. Algorithms in Combinatorial Geometry. Volume 10 of EATCS Monographs on Theoretical Computer Science. Springer-Verlag, Heidelberg, West Germany, 1987.

[38] H. Edelsbrunner, L.J. Guibas, and J. Stolfi. Optimal Point Location in a Monotone Subdivision. SIAM Journal on Computing 15, pp. 317-340, 1986.

[39] H. Edelsbrunner, and R. Seidel. Voronoi Diagrams and Arrangements. Discrete and Computational Geometry 1, pp. 25-44, 1986.

[40] S. Fortune. A Sweepline Algorithm for Voronoi Diagrams. Algorithmica, 2:153-174, 1987.

[41] S. Fortune. Voronoi Diagrams and Delaunay Triangulations. In: J.E. Goodman and J. O'Rourke (Eds.), Handbook of Discrete and Computational Geometry, Chapter 20, CRC Press LLC, pp. 377–388, 1997.

[42] J.E. Goodman and R. Pollack. On the Combinatorial Classification of Non-Degenerate Configurations in the Plane. Journal of Combinatorial Theory, Ser. A 29, pp. 220–235, 1980.

[43] L. J. Guibas and J. Stolfi. Primitives for the manipulation of general subdivisions and the computation of Voronoi Diagrams. ACM Trans. Graph., 4(2):74-123, Apr. 1985.

[44] S. Har-Peled and B. Raichel. On the Complexity of Randomly Weighted Multiplicative Voronoi Diagrams. Proc. 29th Annual Symposium on Computational Geometry (SoCG '13), 2013.

[45] M. I. Karavelas and M. Yvinec. The Voronoi Diagram of Planar Convex Objects. 11th European Symposium on Algorithms (ESA'03). LNCS 2832, pp. 337–348, 2003.

[46] D. G. Kirkpatrick. Efficient Computation of Continuous Skeletons. Proc. 20th Ann. IEEE Symposium on Foundations of Computer Science, 1979, 18-27.

[47] R. Klein. Algorithmische Geometrie. Springer Berlin Heidelberg New York, 2. Auflage, 2005.

[48] R. Klein. Concrete and Abstract Voronoi Diagrams. Lecture Notes in Computer Science 400, Springer-Verlag, 1989.

[49] R. Klein, E. Langetepe, and Z. Nilforoushan. Abstract Voronoi Diagrams Revisited. Computational Geometry: Theory and Applications 42(9), pp. 885-902, 2009.

[50] R. Klein and A. Lingas. Hamiltonian abstract Voronoi diagrams in linear time. 1994 International Symposium on Algorithms and Computation (ISAAC'94), pp. 11–19, 1994.

[51] R. Klein and A. Lingas. Manhattonian proximity in a simple polygon. International Journal of Computational Geometry and Applications 5, pp. 53–74, 1995.

[52] R. Klein and A. Lingas. A linear-time randomized algorithm for the bounded Voronoi diagram of a simple polygon. International Journal of Computational Geometry and Applications 6(3), pp. 263–278, 1996.

[53] R. Klein, K. Mehlhorn, and S. Meiser. Randomized Incremental Construction of Abstract Voronoi Diagrams. Computational Geometry: Theory and Applications 3, pp. 157–184, 1993.

[54] F. Labelle and J.R. Shewchuk. Anisotropic Voronoi Diagrams and Guaranteed-Quality Mesh Generation. Proc. of the nineteenth annual symposium on Computational geometry, SoCG pp. 191-200, 2013.

[55] C.L. Lawson. Software for C^1 surface interpolation. In: J.R. Rice (ed.) Mathematical Software III, Academic Press, New York, pp. 161-194, 1977.

[56] D.T. Lee. Medial Axis Transformation of a Planar Shape. IEEE Transaction on Pattern Analysis and Machine Intelligence 4, 1982, 363-369.

[57] D.T. Lee. On k-Nearest Neighbor Voronoi Diagrams in the Plane. IEEE Trans. Computers 31(6), pp. 478-487, 1982.

[58] D. T. Lee and A. Lin, Generalized Delaunay triangulations for planar graphs, Discrete Computational Geometry 1, pp. 201–217, 1986.

[59] C.-H. Liu, E. Papadopoulou, and D.-T. Lee. An output-sensitive approach for the L_1/L_∞ k-Nearest-Neighbor Voronoi diagram. 19th European Symposium on Algorithms, ESA '11, LNCS 6942, pp. 70–81, 2011.

[60] K. Mehlhorn, St. Meiser, and C. Ó'Dúnlaing. On the Construction of Abstract Voronoi Diagrams. Discrete and Computational Geometry 6, pp. 211–224, 1991.

[61] K. Mehlhorn, S. Meiser, and R. Rasch. Furthest Site Abstract Voronoi Diagrams. International Journal of Computational Geometry and Applications 11(6), pp. 583–616, 2001.

[62] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu. Spatial Tessellations: Concepts and Applications of Voronoi Diagrams. Wiley Series in Probability and Statistics, 2000.

[63] E. Papadopoulou and M. Zavershynskyi. The Higher Order Voronoi Diagram of Line Segments. Algorithmica, DOI 10.1007/s00453-014-9950-0, 2014.

[64] F.P. Preparata, and M. I. Shamos. Computational Geometry: An Introduction. Springer-Verlag, New York, NY, 1985.

[65] W. Rinow. Topologie. VEB Deutscher Verlag der Wissenschaften, Leipzig, 1975.

[66] R. Seidel. A Simple and Fast Algorithm for Computing Trapezoidal Decompositions and for Triangulating Polygons. Computational Geometry: Theory and Applications 1, pp. 51–64, 1991.

[67] M. I. Shamos and D. Hoey. Closest-point problems. Proc. 16th Ann. IEEE Symposium on Foundations of Computer Science, 151-162, 1975.

[68] M. Sharir and P. Agarwal. Davenport-Schinzel Sequences and Their Geometric Applications. Cambridge University Press, 1995.

[69] K. Siddiqi and S.M. Pizer. Medial Representations. Mathematics, Algorithms, and Applications. Springer Series on Computational Imaging and Vision 37, 2008.

[70] A.H. Thiessen. Precipitation average for large area. Monthly Weather Review 39, 1082–1084, 1911.

[71] G.F. Voronoi. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Deuxième Mémoire: Recherches sur les parallélloèdres primitifs. Journal für die reine und angewandte Mathematik 134, 198–287, 1908.

[72] G.F. Voronoi. Deuxième Mémoire: Recherches sur les parallélloèdres primitifs. Journal für die reine und angewandte Mathematik 136, 67–181, 1909.

[73] C.K. Yap. An $O(n \log n)$ Algorithm for the Voronoi Diagram of a Set of Simple Curve Segments. Discrete & Computational Geometry 2, 1987, 365-393.