Higher-Order Voronoi Diagrams of Polygonal Objects

Doctoral Dissertation submitted to the
Faculty of Informatics of the Università della Svizzera Italiana
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

presented by
Maksym Zavershynskyi

under the supervision of
Prof. Evanthia Papadopoulou

December 2014

Dissertation Committee

Prof. Kai Hormann	Università della Svizzera italiana, Lugano, Switzerland
Prof. Michael Bronstein	Università della Svizzera italiana, Lugano, Switzerland
Prof. Rolf Klein	University of Bonn, Bonn, Germany
Prof. Vera Sacristan	Universitat Politècnica de Catalunya, Barcelona, Spain

Dissertation accepted on December 2014

Research Advisor	PhD Program Director
Prof. Evanthia Papadopoulou	**Prof. Stefan Wolf**

I certify that except where due acknowledgement has been given, the work presented in this thesis is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; and the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program.

Maksym Zavershynskyi
Lugano, December 2014

To Alexander Leonidovich Kamin aka АЛ
and Alexander Alexandrovich Kamin aka A^2

Abstract

Higher-order Voronoi diagrams are fundamental geometric structures which encode the k-nearest neighbor information. Thus, they aid in computations that require proximity information beyond the nearest neighbor. They are related to various favorite structures in computational geometry and are a fascinating combinatorial problem to study.

While higher-order Voronoi diagrams of points have been studied a lot, they have not been considered for other types of sites. Points lack dimensionality which makes them unable to represent various real-life instances. Points are the simplest kind of geometric object and therefore higher-order Voronoi diagrams of points can be considered as the corner case of all higher-order Voronoi diagrams.

The goal of this dissertation is to move away from the corner and bring the higher-order Voronoi diagram to more general geometric instances. We focus on certain polygonal objects as they provide flexibility and are able to represent real-life instances. Before this dissertation, higher-order Voronoi diagrams of polygonal objects had been studied only for the nearest neighbor and farthest Voronoi diagrams. In this dissertation we investigate structural and combinatorial properties and discover that the dimensionality of geometric objects manifests itself in numerous ways which do not exist in the case of points. We prove that the structural complexity of the order-k Voronoi diagram of non-crossing line segments is $O(k(n-k))$, as in the case of points. We study disjoint line segments, intersecting line segments, line segments forming a planar straight-line graph and extend the results to the L_p metric, $1 \leq p \leq \infty$. We also establish the connection between two mathematical abstractions: abstract Voronoi diagrams and the Clarkson-Shor framework.

We design several construction algorithms that cover the case of non-point sites. While computational geometry provides several approaches to study the structural complexity that give tight realizable bounds, developing an effective construction algorithm is still a challenging problem even for points. Most of the construction algorithms are designed to work with points as they utilize their

simplicity and relations with data-structures that work specifically for points. We extend the iterative and the sweepline approaches that are quite efficient in constructing all order-i Voronoi diagrams, for $i \leq k$ and we also give three randomized construction algorithms for abstract higher-order Voronoi diagrams that deal specifically with the construction of the order-k Voronoi diagrams.

Acknowledgements

This research would not have been successful without the supervision of Evanthia Papadopoulou. I am thankful for her patience and kindness while guiding me with her experience through the professional science. I am thankful to her for inviting me in Lugano and giving me an opportunity to have an academic experience.

Special thanks to the members of my dissertation committee – Rolf Klein, Vera Sacristan, Kai Hormann, and Michael Bronstein – for dedicating their time to evaluate my work, and providing very valuable feedback. I am also grateful to Gill Barequet for reading through the journal paper on higher-order Voronoi diagrams of line segments and providing useful comments.

I owe my thanks to the people who gave me valuable comments on the subject of the dissertation: Chih-Hung Liu, Cecilia Bohler, Rafel Jaume, Matthias Henze, Birgit Strodthoff, Mario Kapl, Gabriela Majewska, and Gernot Walzl.

To the young members of our research group – Elena Khramtcova and Sandeep Kumar Dey: It was a pleasure to have our discussions on algorithms and data-structures. Our mutual support and encouragement was an important condition for finishing this dissertation. I am waiting to see you both on the other side of Ph.D.

I am also grateful to Konstantin Rubinov and Aibek Sarimbekov for the deep conversations that we had: You are the people who were shaping my personality during these four years. Thanks also to Iegor Nechyporenko, Ilya Markov, Kirill Lykov, Ilya Yanok and Artiom Kovnatsky for having a finger in a pie.

Beyond any measure is my eternal gratitude to my parents who are always by my side.

Maksym Zavershynskyi
December 2014

Contents

Figures

Chapter 1

Introduction

The Voronoi diagram is a powerful geometric structure. It represents the proximity information of objects and has many applications in areas where such information is important.

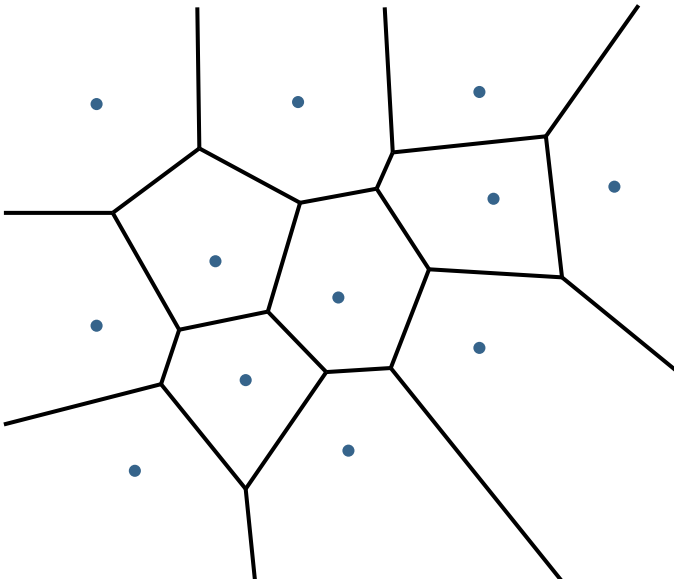

Figure 1.1. The nearest neighbor Voronoi diagram of points.

The classic Voronoi diagram is the nearest neighbor Voronoi diagram in the plane, see Figure 1.1. The nearest neighbor Voronoi diagram of a set of objects S in the plane, called sites, is the partitioning of the plane into regions, such that every point within a fixed region has the same nearest site. For instance, for

points in the Euclidean metric,[1] the Voronoi region of the site s can be defined as:

$$V(s,S) = \{x \in \mathbb{R}^2 \mid \forall t \in S \setminus \{s\} \;\; d(x,s) < d(x,t)\}. \tag{1.1}$$

The nearest neighbor Voronoi diagram is

$$\mathcal{V}(S) = \bigcup_{s \in S} \partial V(s,S), \tag{1.2}$$

where ∂ denotes the boundary.

We can define the nearest neighbor Voronoi diagram in different metrics and space dimensions. In this dissertation we consider only 2-dimensional Voronoi diagrams and mostly those in the Euclidean metric.

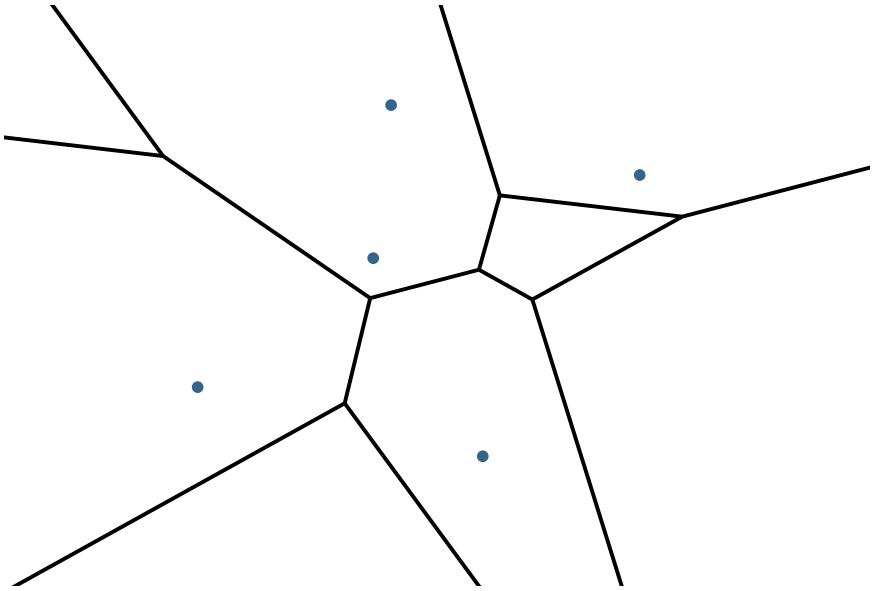

Figure 1.2. The order-2 Voronoi diagram of points.

Often, applications require more information, than just a nearest neighbor information. For instance k-nearest neighbor information can be represented by the order-k Voronoi diagram, which is an important generalization of the nearest neighbor Voronoi diagram. The order-k Voronoi diagram of a set of sites S in the plane is a partitioning of the plane into regions, such that each point within a fixed region has the same closest k sites (see Figure 1.2). The order-k Voronoi

[1] In the Euclidean metric distance between points $x = (x_1, x_2)$ and $y = (y_1, y_2)$ is $d(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$

diagram is also called the higher-order Voronoi diagram, if the order k is not specified. The order-k Voronoi region of the set of sites H can be defined as:

$$V_k(H,S) = \{x \in \mathbb{R}^2 \mid \forall s \in H \ \forall t \in S \setminus H \ \ d(x,s) < d(x,t)\}, \tag{1.3}$$

where $H \subset S$ and $|H| = k$. The order-k Voronoi diagram is

$$\mathcal{V}_k(S) = \bigcup_{H \subset S, |H|=k} \partial V_k(H,S). \tag{1.4}$$

For $k = 1$, the order-k Voronoi diagram is the nearest neighbor Voronoi diagram. For $k = n-1$, it is the farthest Voronoi diagram, $\mathcal{V}_f(S)$ – the partitioning of the plane into regions, such that every point within a fixed region has the same farthest site. The farthest Voronoi region of the site s can be defined as:

$$V_f(S) = V_{n-1}(S \setminus \{s\}, S).$$

The farthest Voronoi diagram has a tree structure and its structural complexity bound is linear on the number of sites. The tree structure of the farthest Voronoi diagram is the key in the structural complexity analysis of the higher-order Voronoi diagrams, see Section 3.2.

The higher-order Voronoi diagram of points was introduced by Shamos and Hoey [81]. For point sites the structural and combinatorial properties were studied by Lee [59], who first proved the $O(k(n-k))$ structural complexity bound and proposed the iterative construction algorithm. The results of Lee can be generalized to the L_p metric[2], see [59]. The alternative proof of the structural complexity was also done by Edelsbrunner [38]. In Chapter 2 we give a detailed overview of the construction algorithms for higher-order Voronoi diagrams of points.

A different way to define the Voronoi diagram is given by the abstract Voronoi diagram. The abstract Voronoi diagram is agnostic to the geometric qualities of Voronoi diagrams and is formulated in pure combinatorial and topological terminology. Instead of considering the sites as geometric objects in a fixed metric space, the abstract Voronoi diagram is defined in terms of bisecting curves satisfying some simple combinatorial properties. Once a concrete bisecting system is shown to satisfy these axioms, combinatorial properties and algorithms to construct the abstract Voronoi diagrams are directly applicable. The nearest neighbor abstract Voronoi diagram was introduced by Klein [54], who proved the

[2] In L_p metric distance between points $x = (x_1, x_2)$ and $y = (y_1, y_2)$ is $d(x,y) = \left(|x_1 - y_1|^p + |x_2 - y_2|^p\right)^{1/p}$.

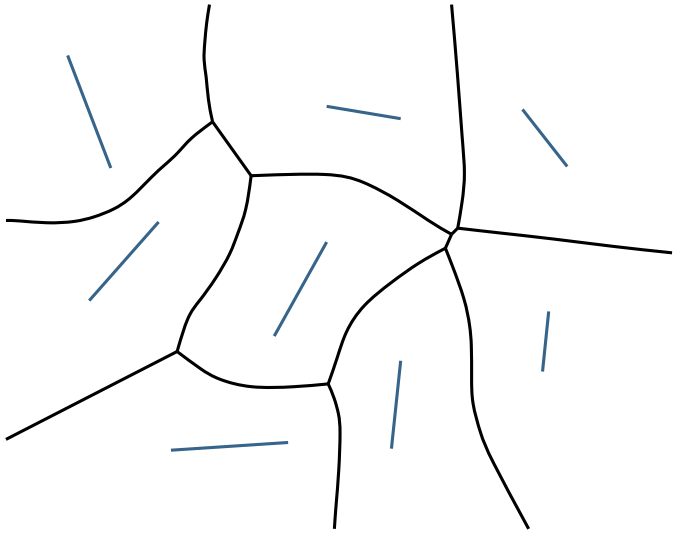

Figure 1.3. The nearest neighbor Voronoi diagram of line segments.

structural complexity bound and proposed construction algorithms. Later, the concept of abstract Voronoi diagrams was generalized to higher-order abstract Voronoi diagrams, see [17]. The number of faces in the higher-order abstract Voronoi diagram is proved to be less than $2k(n-k)$. Non-disjoint line segments, such as line segments forming a planar straight-line graph (PSLG for short) and intersecting line segments, do not fall under the umbrella of abstract Voronoi diagrams, as their bisectors do not comply with the axioms of the underlying system of bisectors. In Chapter 6 we propose three construction algorithms for higher-order abstract Voronoi diagrams.

Many concrete generalizations of Voronoi diagrams have been developed, including Voronoi diagrams in different metrics, object types and space dimensions (for a survey see [10]). The nearest neighbor Voronoi diagram was also studied for sites other than points: line segments, polygons, curves, etc. (see [7, 52, 91]). For instance in the case of line segments the nearest neighbor Voronoi regions are defined as in Eq.(1.1), where the distance between the point x and the line segment s is measured as the minimum distance $d(x,s) = \min_{y \in s} d(x,y)$ (see Figure 1.3). For many cases of disjoint sites the structural complexity is linear and efficient construction algorithms exist. The farthest Voronoi diagram was also studied for sites other than points, e.g. line segments and polygonal objects, see [9, 29]. In both cases the diagram has a tree structure and a linear structural complexity bound.

Surprisingly, the higher-order Voronoi diagram of points is the only concrete

higher-order Voronoi diagram that had been studied so far. However, a variant of the higher-order Voronoi diagram of line segments was introduced in [69].

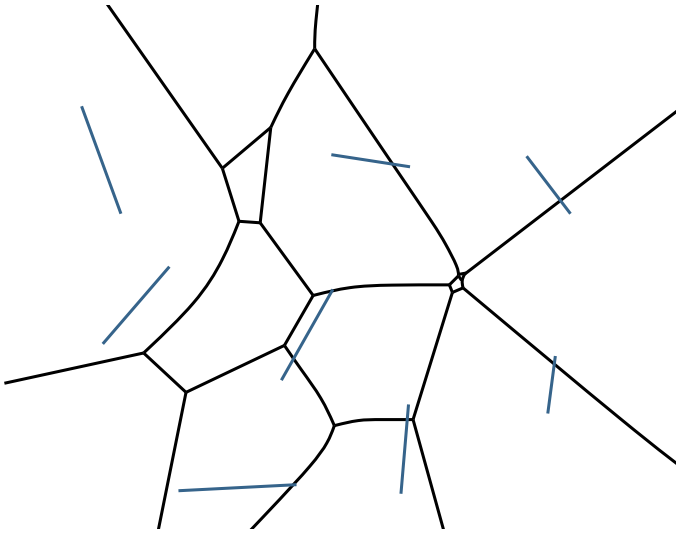

Figure 1.4. The order-2 Voronoi diagram of line segments.

In this dissertation we study higher-order Voronoi diagrams of polygonal objects in the plane, see Figure 1.4. The dissertation includes line segments, line segments forming a planar straight-line graph and some cases of convex polygons (as an application of the results of Chapter 6). We investigate structural properties, prove combinatorial bounds and develop construction algorithms. We investigate the phenomenon of disconnected regions and prove that a single order-k Voronoi region may disconnect into $\Omega(n)$ faces, for $k > 1$. Although an order-k Voronoi region may be disconnected "collectively" they remain connected, see Lemma 3.1.5 and its weaker but more general abstract analog Lemma 6.1.1. Despite the disconnectivities the overall structural complexity of the order-k Voronoi diagram of n disjoint line segments is $O(k(n-k))$.

The line segments that allow intersections introduce additional complications in the structure of the order-k Voronoi diagram. The number of intersections I can be quadratic, in the worst case. For the nearest neighbor Voronoi diagram of intersecting line segments, the structural complexity is $O(n+I)$, while the structural complexity of farthest Voronoi diagram is $O(n)$ and does not depend on the number of intersections. The intuition behind this phenomenon is that the intersections can contribute to the low-order Voronoi diagrams and their influence grows weaker with increasing k. Following the intuition we prove that the structural complexity is $O(k(n-k)+I)$ for $k < n/2$ and $O(k(n-k))$ for

$k \geq n/2$.

The derived structural complexity bounds are also applicable in L_p metric, for $1 \leq p \leq \infty$. In fact, L_1 and L_∞ metrics allow tighter bounds for Voronoi diagrams. The farthest Voronoi diagram of disjoint line segments in L_1 and L_∞ metrics has $O(1)$ structural complexity. We prove that the structural complexity of the order-k Voronoi diagram of disjoint line segments in L_1 and L_∞ metrics is $O(k(n-k))$ for $k < n/2$ and $O((n-k)^2)$ for $k \geq n/2$.

The line segments forming a planar straight-line graph can be used to represent solid structures; the dissertation dedicates a chapter to them. The endpoints and the open portions of the planar straight-line graph can represent different real-life instances and therefore we want to differentiate them on the definition-level. We extend the definition of the order-k Voronoi region in such a way that it respects the information represented by the endpoints of the planar straight-line graph. We study the structural properties of the order-k Voronoi diagram of line segments forming a planar straight-line graph and prove the $O(k(n-k))$ structural complexity bound.

On the algorithmic side of this dissertation we give several construction algorithms for higher-order Voronoi diagrams of polygonal objects.

We extend the iterative algorithm for the construction of higher-order Voronoi diagrams of line segments. In a separate chapter we discuss the sweepline technique for higher-order Voronoi diagrams of line segments. The iterative algorithm and the sweepline technique have $O(k^2 n \log n)$ time complexity; these algorithms can be used to construct all order-i Voronoi diagrams for $i \leq k$. In the case when the application requires all order-i Voronoi diagrams for $i \leq k$ they are the methods of choice. The main advantage of the sweepline algorithm is the low memory requirement. The sweepline algorithm does not need to keep the entire Voronoi diagram and therefore it allows on-the-fly computations.

As mentioned above we provide three construction algorithms for higher-order abstract Voronoi diagrams. These algorithms can be applied to a large variety of concrete sites including but not limited to: disjoint line segments and not-enclosing convex polygons. The algorithms have $O(k^2 n \log n)$, $O(n^2 2^{\alpha(n)} \log n)$ and $O(kn^{1+\varepsilon})$ expected construction time complexities. The first algorithm is the randomized version of the iterative algorithm, which is efficient for small k. The second algorithm is based on the traversal approach and is efficient for values of k close to n. The third algorithm has time complexity which is more suitable for the intermediate values of k that are neither extremely large nor extremely small.

1.1 Applications

The area of applications of the order-k Voronoi diagram should not be confused with the area of applications of the k-nearest neighbor data structures. Though the order-k Voronoi diagram could be used as the data-structure that answers the k-nearest neighbor queries, the purpose of the order-k Voronoi diagram is different. Moreover, the k-nearest neighbor queries are often used in high-dimensional space. The complexity of the order-k Voronoi diagram of points grows exponentially with increasing number of space dimensions [30]. There is however an extensive number of k-nearest neighbor data structures; for an overview of data structures see [77]. Mount and Arya developed a library for approximate nearest neighbor searching that can be also used for exact k-nearest neighbor queries [65].

The order-k Voronoi diagram aggregates the k-nearest information on the infinite number of points. The computation of the order-k Voronoi diagram can be used to subdivide the space into regions where each has the same k-nearest site. The order-k Voronoi diagram is the method of choice when we need a k-nearest neighbor information in the infinite number of points. For instance, we need to integrate a function on the plane the value of which depends on the k-nearest neighbor information. We can do it by subdividing the plane with the order-k Voronoi diagram and then integrating each of the order-k Voronoi regions separately. In the following section we describe how this approach can be used to analyze the vulnerability of the design in semiconductor manufacture.

1.1.1 Critical Area Computation

A variant of the order-k Voronoi diagram of line segments is used in the semiconductor manufacture, see [46, 68, 69, 71, 72]. During the manufacture of the integrated circuits the wafers are prone to defects. We want to evaluate the design of the wafers with respect to the vulnerability to the potential defects. This information can later be used to adjust the design of the wafers and improve the efficiency of the manufacture process.

The defects can be classified as [46]:

- *Random Defects* – defects which are random by nature and could occur because of particle contamination.

- *Systematic Defects* – defects which occur as a result of mechanical or chemical treatment, e.g. polishing.

Both types of defects can cause either partial or complete loss of functionality.

There are a number of models which allow us to predict the loss due to random defects. These models commonly measure the *critical area* that represents the sensitivity of a design to random defects. The methods to compute the *critical area* include: statistical methods (e.g. Monte Carlo simulation), deterministic iterative methods, the Voronoi deterministic method and others [46]. However, only the Voronoi method allows us the exact computation of the *critical area*.

The particle contamination causes two major types of defects:

- *A Short* – an unwanted connection between different conducting regions.
- *An Open* – a broken conducting region.

Critical area measures the sensitivity of the design and is defined as:

$$A_c = \int_0^\infty A(r)D(r)dr,$$

where $A(r)$ denotes the area in which the center of a defect of radius r must fall in order to cause circuit failure and $D(r)$ is the density function of the defect size. $D(r)$ can be estimated as:

$$D(r) = \begin{cases} cr^q/r_0^{q+1}, & 0 \le r \le r_0, \\ cr_0^{p-1}/r^p, & r_0 \le r \le \infty, \end{cases}$$

where typically $p = 3$, $q = 1$, $c = (q+1)(p-1)/(q+p)$, and r_0 is some minimum optically resolvable size.

Consider a defect centered at point t. The critical radius $r_c(t)$ of the defect that causes the short is determined by the second nearest polygon to t. Therefore $t \in A(r)$ iff $r_c(t) \le r$. The order-2 Voronoi diagram partitions the plane into regions where each point within a fixed region has the same pair of closest polygons. The refinement of the order-2 Voronoi diagram gives us the partitioning of the plane where within a fixed region every point t has the critical radiues $r_c(t)$ measured to the same polygon. The order-2 Voronoi diagram allows computation of the critical area by computing it separately for every region [46].

The open defects can be modeled as the problem dual to the shorts. Then the defect that causes an open can be seen as the defect that creates an unwanted connection between the edges of the conducting region.

In order to increase the design reliability, designers are introducing redundant loops that may vary in size, be local and span over a number of layers [69].

The loops are called to reduce the potential for open circuits, but make the critical area extraction more complicated. The defect may locally break a conducting region without breaking the circuit. The second nearest neighbor information is not sufficient to solve the dual problem. We need to construct the order-k Voronoi diagram with larger values of k, see [69]. The experimental results show that $k = 4$ is sufficient [69].

1.1.2 Other Applications

The Voronoi diagrams of polygonal objects have numerous applications. One of the most interesting applications of the Voronoi diagrams of polygonal objects is found in vectorization of raster images. When converting the raster image to a vector format, some of the contour lines may acquire unnecessary connections and others may break. The Voronoi diagram of the contour lines may reveal such mistakes of the vectorizations, see [64]. Additionaly, the Voronoi diagram of a polygon (a medial axis) is used in the shape analysis [14, 15, 16, 58], pattern recognition [21, 44], image processing [56, 88], and mesh generation [36, 47].

The applications of the higher-order Voronoi diagrams are mostly limited to the case of points (the exception is the critical area computation discussed in Section 1.1.1). This is probably due to the fact that higher-order Voronoi diagrams were studied only for points. In the case of points the applications should allow to approximate the real-life instances with a point, which can be done only for a narrow class of applications like: facility location problem [50], retail trade area analysis [19], and spatial interpolation [87]. We hope that the results of this dissertation will open the higher-order Voronoi diagrams to a broad range of applications.

Interesting results are achieved in the applications which substitute the nearest neighbor distance with the k-nearest distance. In some cases the k-nearest distance has shown to be less sensitive to the noises. For instance, the applications on the normal estimation from the point cloud are much more robust to noise if instead of the ordinary distance they use the k-nearest distance [32, 63].

For the full overview of applications see [67].

1.2 Dissertation Goals and Contributions

The goal of this dissertation is to study higher-order Voronoi diagrams of polygonal objects in the plane. We investigate structural and combinatorial properties of higher-order Voronoi diagrams of line segments, and we consider structural

complexity bounds for disjoint line segments, intersecting line segments and line segments forming a planar straight-line graph, including non-Euclidean metrics. We also develop several construction algorithms that can be used for higher-order Voronoi diagrams of polygonal objects; this includes three algorithms for higher-order abstract Voronoi diagrams. The rest of this section gives a detailed overview of the results described in the dissertation.

1.2.1 Structural Properties and Complexity Bounds

We study structural properties of the higher-order Voronoi diagrams of polygonal sites.

- We study the phenomenon of **disconnected regions** which manifests itself only for non-point sites. We show that a single order-k Voronoi region may disconnect into $\Omega(n)$ faces, for $k > 1$. We prove that the union of all faces induced by the same segment is a connected region with the "weakly star-shaped" structure.

- We investigate **the structural and combinatorial properties of the higher-order Voronoi diagram of disjoint line segments**. We prove the *visibility property* in a farthest Voronoi region and we count the number of unbounded faces, which we later use to prove the structural complexity bound:
$$O(k(n-k))$$

- We investigate **the structural and combinatorial properties for intersecting line segments**. Despite the fact that the number of intersections I can be quadratic in the worst case, the intersections may influence only the low-order Voronoi diagrams and the influence grows weaker with increasing order. Namely, the structural complexity is:
$$\begin{aligned} O(k(n-k)+I), &\quad k < n/2 \\ O(k(n-k)), &\quad k \geq n/2 \end{aligned}$$

- We extend **the results on the structural complexity to L_p metric**. We show that the same structural complexity bounds hold in any L_p metric, $1 \leq p \leq \infty$. For non-crossing line segments in L_1 and L_∞ we derive sharper bounds:
$$\begin{aligned} O(k(n-k)), &\quad k < n/2 \\ O\left((n-k)^2\right), &\quad k \geq n/2 \end{aligned}$$

- We investigate **the properties of the iterative construction**. We show that the portion of the order-(k+1) Voronoi diagram enclosed inside a single face of the order-k Voronoi diagram forms a forest. We show that appearances of the line segments along the boundary of the face of the higher-order Voronoi region form a Davenport-Schinzel sequence[3], for disjoint line segments it is order-2 and for intersecting line segments it is order-4.

- We **extend the definition of the Voronoi regions** to cover the case of line segments forming a planar straight-line graph. We investigate the properties of the Type-1 and Type-2 regions to show the consistency of the extended definition. We show that the extended definition has the relation with 3-dimensional arrangements of distance functions, similarly to the not-extended definition.

- Using the perturbation techniques we prove the structural complexity bounds of **the higher-order Voronoi diagram of line segments forming a planar straight-line graph**:

$$O(k(n-k))$$

- We **establish the connection between two mathematical abstractions: the Clarkson-Shor abstract framework and Klein's abstract Voronoi diagrams**. We define the abstract notion of a "conflict" on the abstract Voronoi diagrams in way that makes it possible to apply the results of the Clarkson-Shor framework for the abstract Voronoi diagrams.

- We further investigate **the properties of the higher-order abstract Voronoi diagrams**. We prove that the union of all faces induced by the same site is a simply connected set[4] and we show the properties of its boundary that allow us to traverse it.

[3] A finite sequence is said to be a Davenport-Schinzel sequence of order-s if: (1) No two consecutive values are equal; (2) If x and y occur in the sequence and $x \neq y$, then the sequence does not contain a subsequence $\ldots x, \ldots y, \ldots x, \ldots y, \ldots$ consisting of s+2 alternations between x and y.

[4] The set is simply connected if it is path-connected and every path between two points can be continuously transformed, staying within the set, into any other such path while preserving the endpoints.

1.2.2 Construction Algorithms

We give several construction algorithms that can be applied to the case of line segments, line segments forming a PSLG, abstract setting, certain specific cases of convex polygons, additively weighted points[2] and power diagrams[2].

- We extend an **iterative algorithm** to the case of line segments and line segments forming a PSLG. The algorithm is easy to implement and it has the following time complexity

$$O\left(k^2 n \log n\right),$$

  which is very efficient for small values of k. This algorithm is also very efficient in case all order-i Voronoi diagrams are needed, for $i \leq k$. The time complexity may also be further improved.

- We give a **sweepline algorithm** for the case of line segments and line segments forming a PSLG. The algorithm has the following time complexity:

$$O\left(k^2 n \log n\right)$$

  Similarly to the iterative algorithm, the sweepline algorithm is very efficient in the construction of all order-i Voronoi diagrams, for $i \leq k$. The sweepline algorithm does not need to maintain the entire order-k Voronoi diagram during the construction, which allows on-the-fly computations during the constructions.

- For higher-order abstract Voronoi diagrams we develop **a randomized iterative algorithm** which has the following expected time complexity:

$$O\left(k^2 n \log n\right)$$

  This algorithm is a natural generalization of the ordinary iterative algorithm. It can be applicable to many concrete Voronoi diagrams, such as: non-enclosed convex polygons, additively weighted points[2] and power diagrams[2].

- We also give a **random walk method** for the abstract setting that has the following expected time complexity:

$$O\left(n^2 2^{\alpha(n)} \log n\right),$$

[2]In power diagrams and additively weighted points the distance between the point p and the weighted point s is measured as $d_{\text{power}}(p,s) = d^2(p,s) - w(s)$ and $d_{\text{add}}(p,s) = d(p,s) - w(s)$, respectively, where $d(p,s)$ is the regular distance measure and $w(s)$ is the weight of the point s.

where $\alpha(n)$ is the inverse Ackermann function[5], which is less than 5 for any practical applications. This algorithm is an application of Har-Peled's walking technique [48] to the idea of Chazelle and Edelsbrunner [28].

- Using the randomized iterative algorithm and the random walk method we extend Clarkson's algorithm [30] and receive a **randomized divide and conquer algorithm** for the higher-order abstract Voronoi diagrams having the following time complexity:

$$O\left(kn^{1+\varepsilon}\right),$$

where $\varepsilon > 0$ is a constant. The application of the Clarkson-Shor framework to higher-order abstract Voronoi diagrams is not trivial. In particular, the application of the iterative approach of the Clarkson-Shor framework to higher-order abstract Voronoi diagrams is stated as an open problem [61].

1.3 Dissertation Outline

The remainder of the dissertation is organized as follows:

- Chapter 2 gives the literature overview which consists of the state of the art and important results used in this dissertation;
- Chapter 3 focuses on the structural properties and the complexity bounds of the higher-order Voronoi diagram of line segments. We also analyze the cases of intersecting line segments and line segments in L_1/L_∞ metrics;
- Chapter 4 analyzes the case of line segments forming a planar straight-line graph. We give structural properties, the complexity analysis and the iterative algorithm;
- Chapter 5 presents a sweepline algorithm for the higher-order Voronoi diagrams of disjoint line segments and line segments forming a planar straight-line graph;
- Chapter 6 gives three randomized algorithms for the higher-order abstract Voronoi diagrams: randomized divide and conquer algorithm, randomized iterative algorithm and random walk method. Any of these three algorithms can be applied to a concrete higher-order Voronoi diagram as long as the bisectors satisfy the standard assumptions;

[5]The inverse Ackermann function is an extremely slow-growing function, which is less than 5 for any practical input size.

- Chapter 7 concludes the dissertation and established the future directions.

1.4 Publications

The dissertation is based on several published and submitted works.

Chapters 3 and 4 are based on the results which first appeared as an abstract, then as a conference paper and are going to appear as a journal publication:

- Papadopoulou, E. and Zavershynskyi, M. [2014]. The higher-order Voronoi diagram of line segments. To appear in Algorithmica.
- Papadopoulou, E. and Zavershynskyi, M. [2012]. On higher-order Voronoi diagrams of line segments, in Chao, K.-M., Hsu, T.-s., T. and Lee, D.-T. (eds), ISAAC, Vol. 7676 of Lecture Notes in Computer Science, Springer., pp. 177–186.
  - First appeared as an extended abstract. Papadopoulou, E. and Zavershynskyi, M. [2012]. On higher-order Voronoi diagrams of line segments. The 28th European Workshop on Computational Geometry, Assisi, Italy, March 2012, pp. 233–236.

Chapter 5 is based on the paper which first appeared as an abstract then as a conference paper and is currently in the process of being submitted to a journal:

- Zavershynskyi, M. and Papadopoulou, E. [2014]. A sweepline algorithm for higher-order Voronoi diagrams of line segments. To be submitted to a journal.
- Zavershynskyi, M. and Papadopoulou, E. [2013]. A sweepline algorithm for higher-order Voronoi diagrams, ISVD, IEEE, pp. 16–22.
  - First appeared as an extended abstract. Papadopoulou, E. and Zavershynskyi, M. [2013]. A sweepline algorithm for higher-order Voronoi diagrams. The 29th European Workshop on Computational Geometry, Braunschweig, Germany, March 2013, pp. 233–236.

Chapter 6 is based on a conference paper and will be submitted to a journal:

- Bohler, C., Liu, C.-H., Papadopoulou, E. and Zavershynskyi, M. [2014]. A randomized divide and conquer algorithm for higher-order abstract Voronoi diagrams. To be submitted to a journal.

- Bohler, C., Liu, C.-H., Papadopoulou, E. and Zavershynskyi, M. [2014]. A randomized divide and conquer algorithm for higher-order abstract Voronoi diagrams. To appear in the conference proceedings, ISAAC 2014.

  – An abstract also appeared in the booklet of abstracts of EuroGIGA final conference in Berlin. Bohler, C., Liu, C.-H., Papadopoulou, E. and Zavershynskyi, M. [2014]. Randomized Algorithms for Higher-Order Voronoi Diagrams.

Related are also the following publications:

- Bohler, C., Cheilaris, P., Klein, R., Liu, C.-H., Papadopoulou, E. and Zavershynskyi, M. On the complexity of higher-order abstract Voronoi diagrams. Submitted to Computational Geometry Theory and Applications journal, first round of the revision.

- Bohler, C., Cheilaris, P., Klein, R., Liu, C.-H., Papadopoulou, E. and Zavershynskyi, M. [2013]. On the complexity of higher-order abstract Voronoi diagrams, in F. V. Fomin, R. Freivalds, M. Z. Kwiatkowska and D. Peleg (eds), ICALP (1), Vol. 7965 of Lecture Notes in Computer Science, Springer, pp. 208–219.

Chapter 2

Literature Review

In this chapter we give a short literature review on Voronoi diagrams and related topics. We start with the relation between Voronoi diagrams and arrangements which is an important instrument used throughout this dissertation. We then discuss construction algorithms for higher-order Voronoi diagrams of points and we discuss their potential to be extended to cover the cases of polygonal objects. Finally we discuss k-sets and k-levels which are companion topics to order-k Voronoi diagrams.

2.1 Voronoi Diagrams and Arrangements

Higher-order Voronoi diagrams have a close relation with arrangements. An arrangement is a fundamental structure in computational geometry. One can see many problems related to Voronoi diagrams through the prism of arrangements. By applying various mathematical transformations one can see the Voronoi diagram as the substructure of a particular arrangement. This method is widely used for higher-order Voronoi diagrams. In this section we give the definition of the *k-level* and the $\leq$ *k-level* of an arrangement in $\mathbb{R}^d$. We also present the most common transformations used for higher-order Voronoi diagrams which are used throughout the dissertation. For an overview of geometric transformations used in computational geometry see [20].

We follow the framework introduced by Edelsbrunner and Seidel [41] to define the k-level and the $\leq k$-level. The framework allows us to describe arrangements, including the arrangements with the non-transversal intersections used in Chapters 4 and 5.

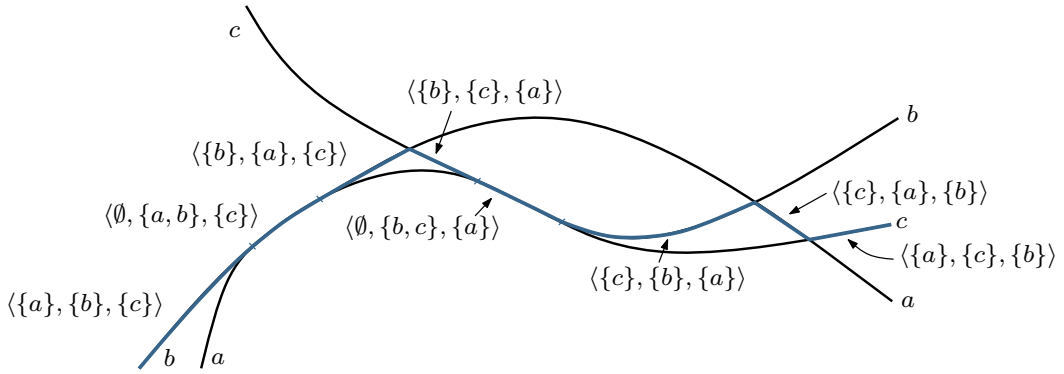

Figure 2.1. 1-level of functions in $\mathbb{R}^2$.

Note. The framework of Edelsbrunner and Seidel as well as other literature defines the k-level in such a way that the lower envelope corresponds to the 1-level [22, 24, 25, 26, 38, 76]. On the other hand, in the literature the lower envelope often corresponds to the 0-level [1, 2, 3, 4, 23, 27, 84, 85]. Moreover, in the Clarkson-Shor technique, the lower envelope can be naturally associated with the configurations with zero conflicts [30, 31, 82]. Since the Clarkson-Shor technique is often used in this dissertation, we define the k-level in a way that the lower envelope corresponds to the 0-level. Therefore, we slightly modify the definition of the k-level in the framework of Edelsbrunner and Seidel so that the lower envelope corresponds to the 0-level.

Let $f \in F$ be a real valued function $\mathbb{R}^{d-1} \to \mathbb{R}$. We define the *lower hemispace of* f, the *upper hemispace of* f, and the *surface of* f as:

$$\begin{aligned} f^- &= \{(x,r) \mid r < f(x)\}, \\ f^+ &= \{(x,r) \mid r > f(x)\}, \\ f^0 &= \{(x,r) \mid r = f(x)\}, \end{aligned}$$

respectively. For each point $y \in \mathbb{R}^d$ define a triple

$$\Pi(y) = \langle \Pi^-(y), \Pi^0(y), \Pi^+(y) \rangle,$$

where

$$\begin{aligned} \Pi^-(y) &= \{f \in F \mid y \in f^-\}, & \pi^-(y) &= |\Pi^-(y)|, \\ \Pi^0(y) &= \{f \in F \mid y \in f^0\}, & \pi^0(y) &= |\Pi^0(y)|, \\ \Pi^+(y) &= \{f \in F \mid y \in f^+\}, & \pi^+(y) &= |\Pi^+(y)|. \end{aligned}$$

For an integer k, $0 \le k < n$ we say that point $y \in \mathbb{R}^d$ belongs to the *k-level* if $\pi^-(y) \le k$ and $\pi^-(y) + \pi^0(y) > k$, see Figure 2.1. The complexity of the

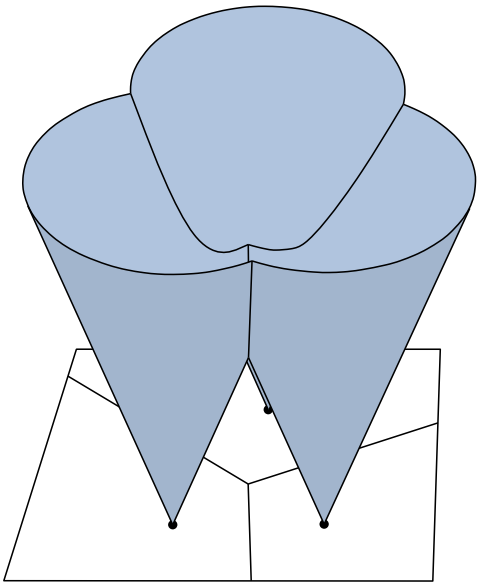

Figure 2.2. The 0-level of distance functions.

k-level is the number of its connected components that have the same triple Π. The $\leq k$-*level* is the union of all i-levels, for $0 \leq i \leq k$. The *lower envelope* is the bottommost level, i.e. 0-level. The *upper envelope* is the topmost level, i.e. $(n-1)$-level, where $|F| = n$.

We give two most commonly used transformations that allow us to map the order-k Voronoi diagram of point-sites with the $(k-1)$-level. These transformations can also be used in higher dimensions; in this case, order-k Voronoi diagram of point-sites in $\mathbb{R}^d$ dimensions is usually mapped to the $(k-1)$-level in $\mathbb{R}^{d+1}$ dimensions. However, in this dissertation we only consider arrangements in $\mathbb{R}^2$ and $\mathbb{R}^3$.

Higher-Order Voronoi Diagrams and Arrangements of Distance Functions One can easily see the relation between higher-order Voronoi diagrams and the arrangements of distance functions. Let S be a set of points located in the xy-plane of $\mathbb{R}^3$. For each point $s = (s_x, s_y) \in S$ consider the distance function $f_s : \mathbb{R}^2 \to \mathbb{R}$:

$$f_s(x) = d(s, x),$$

where $d(\cdot, \cdot)$ is the distance function between two points. Consider the edges and the vertices of the $(k-1)$-level of the distance functions f_s, $s \in S$. The projection of the edges and the vertices on the xy-plane is then the order-k Voronoi diagram of points S on that plane.

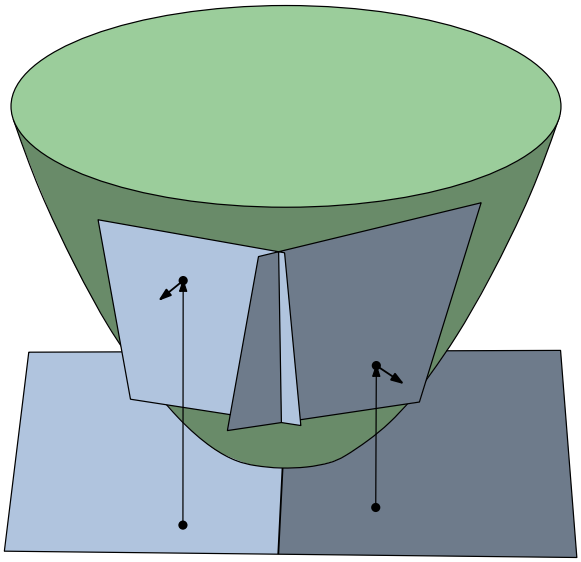

Figure 2.3. The lifting transformation of the points in the xy-plane.

This transformation is often used in the construction of the nearest neighbor Voronoi diagram of sites, see [3, 49, 80]. The transformation is not specific to points, however; it is usually applied only for the construction of nearest neighbor Voronoi diagrams. The reason is that the investigation of the $(k-1)$-level of general functions is not easy, to the best of our knowledge; see Section 2.4.

Lifting Transformation Let S be a set of points located in the xy-plane of $\mathbb{R}^3$. For each point $s \in S$ consider its vertical projection s' onto the paraboloid $z = x^2+y^2$. With every point s we associate a plane tangent to the paraboloid at the point s'. The plane is described with the following function:

$$f_s(x) = 2s \cdot x - s_x^2 - s_y^2,$$

where $\cdot$ denotes the scalar product between $s = (s_x, s_y, 0)$ and x, where x belongs to the xy-plane. Consider the $(n-k-1)$-level in the arrangement of planes that correspond to the points S in the xy-plane. Vertically project the edges of the $(n-k-1)$-level on the xy-plane. The projection will then coincide with the order-k Voronoi diagram of the points S in the xy-plane. The nearest neighbor Voronoi diagram of S corresponds to the upper envelope of the planes and the farthest Voronoi diagram of S corresponds to the lower envelope of the planes. The transformation can also be vertically flipped so that the $(k-1)$-level corresponds to the order-k Voronoi diagram, like in the case of distance functions.

For simplicity in this dissertation we assume the flipped transformation. This transformation is widely used in studying Voronoi diagrams.

The lifting transformation is often used for the construction of higher-order Voronoi diagrams of points; see Section 2.3. However, there is no similar transformation that can be applied to the construction of the higher-order Voronoi diagrams of non point-sites, to the best of our knowledge.

Point-Line Duality Transformation The following standard transformation T maps a point in $\mathbb{R}^d$ to a hyperplane in $\mathbb{R}^d$ and vice versa. This transformation is so common in the computational geometry that it sometimes fails to be explicitly mentioned when used. We call point p "dual" to hyperplane $T(p)$ and vice versa (also $T(T(p)) = p$).

We can use the duality transformation when we transform the lower envelope to the convex hull (or the opposite). Similarly, we use the duality transformation when we transform the $(k-1)$-level to the k-sets; see Section 2.4.

In the case when $d = 2$, it maps a point to a line in $\mathbb{R}^d$ and for this reason it is often called a point-line duality transformation. For a 2D-case we map a point $T(p) = (a, b)$ in the primal plane to a line $T(p) : y = ax - b$ in the dual plane. Points transform to lines and lines transform to points by the formulas:

$$\begin{aligned} (a, b) &\rightarrow y = ax - b \\ y = ax + b &\rightarrow (a, -b) \end{aligned}$$

In the 3D-case, points transform to planes and planes transform to points by the formulas:

$$\begin{aligned} (a, b, c) &\rightarrow z = ax + by - c \\ z = ax + by + c &\rightarrow (a, b, -c) \end{aligned}$$

In the general d-dimensional case, points transform to hyperplanes and hyperplanes transform to points by the formulas:

$$\begin{aligned} (p_1, \ldots, p_d) &\rightarrow x_d = \textstyle\sum_{i=1}^{d-1} p_i x_i - p_d \\ x_d = \textstyle\sum_{i=1}^{d-1} p_i x_i + p_d &\rightarrow (p_1, \ldots, -p_d) \end{aligned}$$

The main property of the transformation is that it preserves the relative position of the geometric objects. The point p is above the hyperplane h iff the point $T(h)$ is above the hyperplane $T(p)$.

This transformation is also rich on useful properties. For instance, consider the points $p_1, \ldots, p_n \in \mathbb{R}^2$ in convex position, where each point belongs to the upper part of the convex hull. Apply the duality transformation and receive the lines $T(p_1), \ldots, T(p_n)$, where each line participates in the lower envelope. In

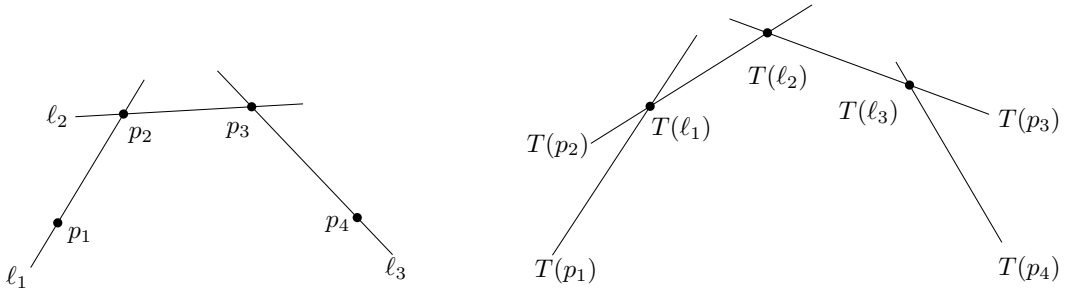

Figure 2.4. Duality transformation performed on four points in convex position.

this dissertation, we say that lines/planes/hyperplanes are in convex position if they correspond to the points in the convex position, under the duality transformation, see Figure 2.4.

We use this transformation throughout the thesis.

2.2 Nearest Neighbor and Farthest Voronoi Diagrams

In this section we give a brief overview of nearest neighbor Voronoi diagrams and farthest Voronoi diagrams. For $k = 1$ the order-k Voronoi diagram is the nearest neighbor Voronoi diagram. For $k = n-1$ the order-k Voronoi diagram is the farthest Voronoi diagram. We use these results as the corner cases in our investigation of higher-order Voronoi diagrams of polygonal objects for general k. In particular, we use the structural complexity bounds for $k = 1$ and $k = n-1$ to derive the structural complexity bound for general k in the case of line segments, see Chapter 3. The construction algorithms for higher-order Voronoi diagrams discussed throughout the dissertation use the construction algorithms for $k = 1$ and $k = n-1$ for subroutines.

The nearest neighbor Voronoi diagram of points is the oldest and the most well-studied Voronoi diagram. The structural complexity is $O(n)$, where n is the number of points. In particular, the number of faces F_1 is equal to n. The lifting transformation (see Section 2.1) allows to study the nearest neighbor Voronoi diagram of points as the projection of planes in $\mathbb{R}^3$. It is also dual to the Delaunay triangulation. The unbounded faces of the nearest neighbor Voronoi diagram correspond to sites that appear on the convex hull of points. Therefore the number of unbounded faces U_1 is equal to the size of the convex hull. There are construction algorithms available with the best deterministic time complexity $O(n \log n)$ and $O(n)$ space complexity [43].

The farthest Voronoi diagram of points is a counterpart of the nearest neigh-

bor Voronoi diagram of points. The structural complexity is also $O(n)$. However, the number of faces F_{n-1} is less or equal to n. In fact all faces are unbounded and therefore the number of unbounded faces U_{n-1} is equal to the total number of faces F_{n-1}. Moreover, the structure of the farthest Voronoi diagram of points is a tree. The unbounded faces of the farthest Voronoi diagram correspond to sites that appear on the convex hull of points. This immediately implies $U_1 = U_{n-1}$ for the same set of sites. This is a very useful property for the structural complexity analysis. For instance, one can derive $V_1 + V_{n-1} = 2(n-1)$, where V_1 is the total number of nearest neighbor Voronoi vertices and V_{n-1} is the number of farthest Voronoi vertices. This equality does not depend on the position of the sites. It is used in the structural complexity analysis of the order-k Voronoi diagram [31]. For general k this property is known as *symmetry property*, see Chapter 3. There are also construction algorithms available for the farthest Voronoi diagram of points, with $O(n \log n)$ time complexity and $O(n)$ space complexity [10, Section 6.5.1].

The nearest neighbor Voronoi diagram of line segments is also well-studied. For disjoint line segments and line segments forming a planar straight-line graph, the structural complexity is $O(n)$, see [52]. For intersecting line segments the structural complexity is $O(n + I)$, where I is the number of intersections, which can be quadratic. For line segments forming a planar straight-line graph there are construction algorithms available with $O(n \log n)$ time complexity and $O(n)$ space complexity [52, 58, 91].

The farthest Voronoi diagram of line segments was studied only recently, showing properties surprisingly different from the case of points [9]. For instance, in the case of line segments the farthest Voronoi region may disconnect into $\Omega(n)$ faces. However, the overall structural complexity of the farthest Voronoi diagram is $O(n)$, same as for points. The structural complexity also remains linear, $O(n)$ even in the case of intersecting line segments. The structure of the farthest Voronoi diagram is a tree, the same as for points. There is a construction algorithm available with $O(n \log n)$ time complexity and $O(n)$ space complexity [9]. However, the number of unbounded faces $U_{n-1} = F_{n-1}$ of the farthest Voronoi diagram is not equal to the number of unbounded faces of the nearest neighbor Voronoi diagram U_1. This narrows down the number of techniques that can be used to study the order-k Voronoi diagram of line segments, see Chapter 3.

The nearest neighbor Voronoi diagram was also studied for curved objects [7, 91]. The structural complexity is the same as in the case of line segments. Construction algorithms with $O(n \log n)$ time complexity and $O(n)$ space complexity are also available.

The farthest Voronoi diagram of simple polygons was also studied only recently [29]. Simple polygons introduce additional complications in comparison to points and line segments. In particular, the bisectors can be closed curves, which happens even in the case of disjoint polygons that are not enclosed. Since line segments are a special case of simple polygons, the farthest Voronoi diagram of simple polygons can also have disconnected regions. The structural complexity is $O(n)$. A construction algorithm with $O(n \log^3 n)$ time complexity is available.

2.3 Construction Algorithms

In this Section we give an overview of construction algorithms for higher-order Voronoi diagrams of points. Some of these algorithms use the lifting transformation. The lifting transformation allows us to reduce the construction of the order-k Voronoi diagram of points in $\mathbb{R}^2$ to the construction of the $(k-1)$-level of planes in $\mathbb{R}^3$. Planes have the following important properties that are used by some of the algorithms:

- The plane is uniquely defined by the three points. For instance, this property is used in Clarkson's algorithm [30];
- We can triangulate an arrangement of planes in $\mathbb{R}^3$. For instance, this operation is used by the algorithm of Agarwal et al. [2];
- We can consider the intersection of the rest of the 3D-arrangement with one of the planes, which allows us to operate in 2D. This operation is used by Chazelle and Edelsbrunner [28].

Moreover, the planes obtained by the lifting transformation are tangent to the paraboloid and they are said to be in convex position (the dual points are in convex position). The planes in convex position are naturally easier to investigate than the planes in general position. Consequently, the lifting transformation is often the method of choice.

Lee's Iterative Algorithm The iterative algorithm by Lee [59] was the first construction algorithm for higher-order Voronoi diagrams of points. Starting with the nearest neighbor Voronoi diagram of points, the algorithm iteratively constructs the order-i Voronoi diagram using the order-$(i-1)$ Voronoi diagram, until $i = k$ is reached. The iterative algorithm has $O(k^2 n \log n)$ time complexity,

which is greater than the complexity of the order-k Voronoi diagram, $O(k(n-k))$. However, the algorithm constructs not only the order-k Voronoi diagram but all order-i Voronoi diagrams for $i \leq k$. For some applications, the construction of all order-i Voronoi diagrams is needed, $i \leq k$. In this case the iterative algorithm is very close to efficient. The iterative algorithm is also very efficient for small values of k. The simplicity of the iterative algorithm makes it easy to implement. The fact that the algorithm does not use any point-specific transformations allows it to be extended to some non-point sites, see Sections 4.4 and 6.3.

The Improvement of the Iterative Algorithm of Aggarwal et al. The nature of the iterative algorithm implies the construction of all order-i Voronoi diagrams for $i \leq k$. Their total complexity is $O(k^2 n)$. Therefore we may try to improve the running time of the algorithm by reducing the logarithmic factor. The construction of the order-i Voronoi diagram is done by subdividing the face F of the order-$(i-1)$ Voronoi diagram with the nearest neighbor Voronoi diagram $\mathcal{V}(S_F)$, where S_F is the set of the sites defined by the neighboring faces. Lee's version of the iterative algorithm constructs $\mathcal{V}(S_F)$ using the ordinary construction algorithm for the nearest neighbor Voronoi diagrams in $O(|S_F| \log |S_F|)$ time. We can improve the subdivision step using the properties of $\mathcal{V}(S_F)$. Namely, $\mathcal{V}(S_F)$ has a tree structure inside the face F and the ordering of the sites S_F along the boundary of F is known. This allows the deterministic linear time construction of $\mathcal{V}(S_F)$; see the algorithm by Aggarwal et al. [5]. The total construction time is therefore $O(k^2 n + n \log n)$, since we still need to construct the order-1 Voronoi diagram using the $O(n \log n)$ algorithm. The original description of the algorithm by Aggarwal et al. is specific to points, since it uses the fact that the points are in convex position. However, this property can be extended to the "tree-like" structures that include non-point sites, see [51].

The Sweepline Algorithm of Rosenberger Rosenberger proposed a sweepline algorithm for higher-order Voronoi diagrams of weighted points [76]. His algorithm refers to the surfaces represented by the distance functions; however this transformation is not essential for the algorithm. The algorithm extends the idea of Fortune's algorithm [43] to higher-order Voronoi diagrams. The algorithm sweeps the plane with the horizontal line while maintaining the $\leq k$-level of parabolas. Similarly to the iterative algorithm, it constructs not only the order-k Voronoi diagram, but all order-i Voronoi diagrams for $i \leq k$. The time complexity is $O(k^2 n \log n)$.

The Zone Algorithm of Chazelle and Edelsbrunner Chazelle and Edelsbrunner proposed the first algorithm which constructs the order-k Voronoi diagram directly, without constructing the diagrams of lower orders [28]. We refer to this as "the zone algorithm" because it reduces the problem to computing the zone of a certain arrangement. The algorithm is an improvement of the algorithm proposed by Edelsbrunner [37]. The algorithm performs the lifting transformation and constructs the $(k-1)$-level of planes in $\mathbb{R}^3$. However, the transformation is not essential for the algorithm and in Section 6.4 we show how to use the same idea without the transformation. The algorithm takes one of the planes and considers the intersection of the $(k-1)$-level with it. It then constructs the intersection and the process is repeated n times for each of the planes. Depending on the data structures used in the construction process, the algorithm can have $O(n^2 \log n + k(n-k)\log^2 n)$ or $O(n^2 + k(n-k)\log^2 n)$ time complexity. The intersection of the plane and the rest of the arrangement can also be considered without the geometric transformations. In this case it becomes the boundary of the k-neighborhood described in the Sections 3.1 and 6.4.

The Semi-dynamic Randomized Algorithm of Boissonat et al. The algorithm of Boissonnat et al. [18] allows us to add points after the order-k Voronoi diagram is constructed. This is the first algorithm that allows us to add the points without reconstructing the entire order-k Voronoi diagram. It introduces the k-Delaunay tree data structure, which can be used to deduce all order-i Voronoi diagrams for $i \leq k$. With the insertion of the new point, the algorithm updates the k-Delaunay tree. The k-Delaunay tree contains all successive versions of all order-i Voronoi diagrams for $i \leq k$ and it allows fast point location. The total expected construction time is $O(n \log n + k^3 n)$.

The Algorithm of Aurenhammer Aurenhammer introduced a beautiful transformation for higher-order Voronoi diagrams of points S in $\mathbb{R}^d$ [8]. The transformation maps a set of points in $\mathbb{R}^d$ to a single point in $\mathbb{R}^{d+1}$. Each subset $H \subset S$ of size k is associated with the point $\xi(H) = (\xi_1, \ldots, \xi_{d+1})$, where

$$(\xi_1, \ldots, \xi_d) = \sum_{p \in H} p,$$
$$\xi_{d+1} = \sum_{p \in H} p^2.$$

The lower part of the convex hull of the set of points $\Xi = \{\xi(H) \mid H \subset S, |H| = k\}$ in $\mathbb{R}^{d+1}$ corresponds to the order-k Voronoi diagram of points S in $\mathbb{R}^d$. Since the

number of points in Ξ is $\binom{n}{k}$ and the complexity of the order-k Voronoi diagram is $O(k(n-k))$, it is clear that the majority of the points in Ξ do not contribute to the lower part of the convex hull. The transformation is obviously specific to points. The algorithm constructs the order-k Voronoi diagram of n points in $O(k^2 n \log n)$ time.

The On-Line Randomized Incremental Algorithm of Aurenhammer and Schwarzkopf Based on the previous algorithm Aurenhammer and Schwarzkopf developed an on-line randomized incremental algorithm [11]. An on-line randomized incremental algorithm constructs the order-k Voronoi diagram adding the points one by one, while maintaining the intermediate order-k Voronoi diagram. Thus it can be used in a dynamic setting. The time complexity of the algorithm is $O(k^2 n \log n + nk \log^3 n)$.

In addition to the construction algorithm, Aurenhammer and Schwarzkopf investigated why it is so difficult to devise an efficient randomized incremental algorithm for higher-order Voronoi diagrams of points and derived the following result:

If the n sites are added at random while their order-k Voronoi diagram is maintained, the expected number of Voronoi vertices that appear at some intermediate stage during the algorithm is $\Theta(nk^2)$. [11]

Suppose we have an algorithm that constructs the order-k Voronoi diagram by adding points one by one and updating the entire order-k Voronoi diagram. The above result implies that during this process we encounter $\Theta(nk^2)$ expected number of vertices. Since all of these vertices have to be constructed, the running time of the algorithm can not be better than $\Theta(nk^2)$.

Therefore we need to avoid maintaining the entire intermediate order-k Voronoi diagram in incremental algorithms. For instance, the algorithm of Agarwal et al. (see below) maintains only the necessary parts of the intermediate order-k Voronoi diagrams.

The Algorithm of Mulmuley Mulmuley proposed an output sensitive randomized algorithm to construct the order-k Voronoi diagram of points [66]. The algorithm also constructs all order-i Voronoi diagrams for $i \leq k$, and it is output-

sensitive with respect to their total size, which is $O(k^2n)$. The expected running time of the algorithm is $O(k^2n + n\log n)$. The algorithm constructs the order-k Voronoi diagram by applying the lifting transformation and then constructing the $\leq k$-level of planes in $\mathbb{R}^3$. It can also be used for higher-dimensions where output sensitivity is important.

The Randomized Divide and Conquer Algorithm of Clarkson Clarkson developed an abstract framework for the random sampling technique [30]. The framework provided a number of new combinatorial results as well as the construction algorithm for higher-order Voronoi diagrams of points. Later this framework was extended and generalized by Clarkson and Shor, and since then it is called the Clarkson-Shor framework [31]. Clarkson's algorithm lifts the points on the paraboloid and finds all possible k-sets in $\mathbb{R}^3$ (see the definition of k-set in the Section 2.4). It is a randomized divide and conquer algorithm which uses the algorithms of Lee [59], Chazelle and Edelsbrunner [28] to solve the problem for small subinstances. The time complexity analysis uses the combinatorial results of the Clarkson-Shor framework. The algorithm has $O(kn^{1+\varepsilon})$ expected time complexity, where $\varepsilon > 0$ is a constant.

The main idea of Clarkson's algorithm is to cut the set of points into smaller subsets that allow independent construction. This is achieved through defining the notion of a conflict that allows us to "bracket" the space. The subsets may however overlap. Moreover, the "conquer" step of the divide and conquer approach requires verification of the results of the computation for smaller subsets. However, the complicated expected time complexity analysis proves that the expected running time is good enough, $O(kn^{1+\varepsilon})$. The analysis uses the following trick: it performs both the lower and the upper bound complexity analysis at the same time. Thus the conflicts are said to "bracket" the space. It allows to bound the size of the subsets and the depth of the recursion at the same time.

Despite the fact that the algorithm exploits the lifting transformation and the properties of the planes in $\mathbb{R}^3$, we have developed an extension of the algorithm to the case of higher-order abstract Voronoi diagrams, see Chapter 6.

The Algorithm of Agarwal et al. Agarwal et al. proposed the first construction algorithm with $\widetilde{O}(kn)$ time complexity [2], where $\widetilde{O}$-notation hides a polylogarithmic factor in this dissertation. The algorithm does the lifting transformation and then constructs the k-level of planes in $\mathbb{R}^3$. It inserts the planes one by one; however, it does not maintain the complete k-level in between. Instead it maintains those parts of the intermediate k-levels that can potentially contribute to

the final k-level. The algorithm also performs a triangulation of the arrangement of the planes to achieve the representation of the arrangement through the simplices. In the process the simplices are evaluated with respect to their possible contribution to the final k-level, and those simplices that can not contribute are discarded. The expected running time of the algorithm is $O(n\log^3 n + nk\log n)$.

The Algorithm of Chan Chan presents the following result [23]:

Given an algorithm that constructs a k-level of n planes in $\mathbb{R}^3$ in $O(f(n))$ time (where $f(n)$ is a regular function) we can construct the k-level in $O(n\log n + (n/k)f(k))$ expected time.

We can use the lifting transformation to reduce the problem of constructing the order-k Voronoi diagram of points to constructing the k-level of planes in $\mathbb{R}^3$. Chan plugs the algorithm of Agarwal et al. [2] in the above formula and derives the algorithm with $O(n\log n + nk\log k)$ expected running time complexity.

Chan's result is extremely important since it allows us to speed up almost any construction algorithm for higher-order Voronoi diagrams of points.

The Randomized Divide and Conquer Incremental Algorithm of Ramos Ramos proposed a randomized construction algorithm that combines the divide and conquer approach with incremental construction [74]. For small subinstances of the problem it uses the bruteforce method. It also combines the results of Agarwal et al. [2] to subdivide the arrangement. This implies $O(n\log^3 n + nk2^{c\log^* n})$ time complexity[1]. Plugging this result into Chan's result, Ramos receives an algorithm with $O(n\log n + nk2^{c\log^* k})$ expected running time complexity, where c is a constant.

The Zone Algorithm of Chan In the remarks on the k-level algorithm on the plane [22] Chan discusses the possibility of applying the new algorithms and data structures to the old approach of Chazelle and Edelsbrunner [28]. Using the algorithm for constructing the k-level of lines in $\mathbb{R}^2$ [22], Chan's algorithm can construct the order-k Voronoi diagram of points in $O(n^2\log n + m\log^{1+\varepsilon} n)$ deterministic time, where m is the structural complexity of the diagram. One can combine this approach with the results of the other paper of Chan [23] to achieve the best deterministic time $O(nk\log^{1+\varepsilon} k\cdot(\log n/\log k)^{O(1)})$.

The following tables summarize the running time of the algorithms.

[1] The iterated logarithm, $\log^*$ is the number of times the logarithm function must be iteratively applied before the result is ≤ 1, for most practical applications it is less than 6.

Time complexity	Author(s)
$O(k^2 n \log n)$	Lee [59]
$O(k^2 n + n \log n)$	Aggarwal et al. [5]
$O(k^2 n \log n)$	Rosenberger [76]
$O(k(n-k)\sqrt{n} \log n)$	Edelsbrunner [37]
$O(n^2 \log n + k(n-k)\log^2 n)$	Chazelle and Edelsbrunner [28]
$O(n^2 + k(n-k)\log^2 n)$	Chazelle and Edelsbrunner [28]
$O(n \log n + k^3 n)$ Randomized Semi-dynamic	Boissonnat et al. [18]

Table 2.1. The algorithms that do not use the transformations or the usage is not essential.

Time complexity	Author(s)
$O(k^2 n \log n)$	Aurenhammer [8]
$O(k^2 n \log n + nk \log^3 n)$ On-line Randomized Incremental	Aurenhammer and Schwarzkopf [11]
$O(k^2 n + n \log n)$ Randomized	Mulmuley [66]
$O(kn^{1+\varepsilon})$ Randomized	Clarkson [30]
$O(n \log^3 n + nk \log n)$ Randomized	Agarwal et al. [2]
$O(n \log n + nk \log n)$ Randomized	Chan [23]
$O(n \log n + nk 2^{c \log^* k})$ Randomized	Ramos [74]
$O(nk \log^{1+\varepsilon} k \cdot (\log n / \log k)^{O(1)})$	Chan [22]

Table 2.2. The algorithms that use the transformations

Whether the usage of point-specific transformations is essential or not in a particular algorithm is an arguable question. However, one can be sure that there is no simple way to generalize the algorithms in Table 2.2 to general sites other than points. This is due to the fact that there is no transformation for sites like line segments that maps them to a simple geometric object like a plane, to the best of our knowledge.

2.4 k-Sets and k-Levels

In this section we present important results on k-sets and k-levels. The combinatorial and algorithmic results on k-sets and k-levels are used in most of the construction algorithms discussed in Section 2.3. We use the results on the k-levels of curves in every chapter of this dissertation. Moreover, the combinatorial

problems of the k-sets/k-levels bear close resemblance to the problem of finding the optimal construction algorithm for order-k Voronoi diagrams, as we will see later in this section.

Consider a set of points S in $\mathbb{R}^2$. We call H a k-set, $|H| = k$ if there is a line that separates H and $S \setminus H$. The duality transformation implies the immediate relation between the k-sets of points in $\mathbb{R}^2$ and the k-level of lines in $\mathbb{R}^2$. Namely, the number of k-sets of a set of points is the total complexity of the k-level and $(n-k)$-level of the dual lines (see the duality transformation in Section 2.1).

The k-sets can be similarly defined for d-dimensions. In $\mathbb{R}^d$ we call H a k-set, $|H| = k$ if there is a hyperplane that separates H and $S \setminus H$. The k-set then corresponds to a k-level of hyperplanes in $\mathbb{R}^d$.

The structural complexity of the k-level of curves/surfaces in $\mathbb{R}^d$ is an open question for $d \geq 2$. Moreover, even the structural complexity of the k-level of lines (or equivalently, k-sets) in $\mathbb{R}^2$ is a long-standing open problem. Erdős, Lovász, Simmons and Straus conjectured that the number of k-sets of n points in $\mathbb{R}^2$ is bounded by $O(n(k+1)^{\epsilon})$, for every fixed $\varepsilon > 0$ [42].

k-sets and k-levels are important tools for higher-order Voronoi diagrams. The usage of k-sets and k-levels in construction algorithms is described in detail in Section 2.3. However, while using the lifting transformation to reduce the problem of constructing the higher-order Voronoi diagram of points to constructing the k-level of planes it is absolutely necessary to take into account the fact the the produced planes are tangent to the paraboloid. Otherwise constructing the k-level of general planes is a hard problem.

Counting the number of k-sets in $\mathbb{R}^2$ is a long-standing open problem. However, counting the number of $\leq k$-sets is relatively easy. We call H a $\leq k$-set if it is a i-set, where $i \leq k$. Goodman and Pollack showed that for $k < n/2$ the number of $\leq k$-sets is less than $2nk - 2k^2 - k$, see [45]. Alon and Györi improved later the upper bound to kn, see [6]. The bounds are mostly derived by studying the cyclic sequences which are more general than the k-sets. The results are very important for the structural complexity analysis of the higher-order Voronoi diagrams of points. Both Lee's [59] and Edelsbrunner's [38] derivations of the $O(k(n-k))$ structural complexity bound for higher-order Voronoi diagrams of points use the results on $\leq k$-sets.

The looser bounds are also available for structural complexity of the $\leq k$-levels in the arrangements of general Jordan curves. The Clarkson-Shor framework provides an important result that implies the following result:

$$g_{\leq k}(n) = O\left((k+1)^d g_0\left(\left\lfloor \frac{n}{k+1} \right\rfloor\right)\right),$$

where $g_{\leq k}(n)$ is the structural complexity of the $\leq k$-level in the arrangement of n Jordan curves, $g_0(n)$ is the structural complexity of the lower envelope in the arrangement of n Jordan curves and $d = 2$. The bound can also be applied in general d-dimensions. If a pair of curves intersects $O(1)$ number of times then the Davenport-Schinzel Sequences can be used to bound $g_0(n)$, see [84].

Unfortunately, there is a lack of tight bounds for k-sets and k-levels. It is surprising how the problem of bounding the structural complexity of the k-level (as opposed to $\leq k$-level) bears close resemblance with the problem of constructing only the order-k Voronoi diagram (as opposed to constructing all order-i Voronoi diagrams for $i \leq k$). In the following tables we summarize the most recent results on the structural complexity of the k-level.

Class of objects	Best upper bound
Lines in $\mathbb{R}^2$	$O(nk^{1/3})$ [35]
Planes in $\mathbb{R}^3$	$O(nk^{3/2})$ [86]
Hyperplanes in $\mathbb{R}^4$	$O(n^2k^{2-1/18})$ [1, 27, 83]
Hyperplanes in $\mathbb{R}^d$	$O(n^{\lfloor d/2 \rfloor \lceil d/2 \rceil - \alpha_d})$, [1, 12, 27]
2-intersecting curves in $\mathbb{R}^2$	$O(n^{3/2}\log n)$ [26]
3-intersecting curves in $\mathbb{R}^2$	$O(n^{2-1/(3+\sqrt{7})}) = O(n^{1.823})$ [26]
s-intersecting curves in $\mathbb{R}^2$	$O(n^{2-\frac{1}{2s-(s-1)\alpha_s}})$ [26]
Pseudo-planes in $\mathbb{R}^3$	$O(nk^{1.9966})$ [1, 27]

Table 2.3. The most recent results on the upper bounds of the complexity of the k-level.

Class of objects	Best lower bound
Lines in $\mathbb{R}^2$	$n2^{\Omega(\sqrt{\log k})}$ [89]
Planes in $\mathbb{R}^3$	$nk2^{\Omega(\sqrt{\log k})}$ [89]

Table 2.4. The most recent results on the lower bounds of the complexity of the k-level.

In Table 2.3, α_s is a function of s, see [26], α_d is equal to $1/(2d)^d$. Pseudo-planes are surfaces where each triple intersects at most once. The upper bound of Dey on the lines in $\mathbb{R}^2$ is hard to improve, since Dey's proof works for a more general problem than k-sets, for which there is a matching lower bound [25, 35].

For the historical overview of the k-level in $\mathbb{R}^2$, see [22, 90]. For complexity bounds of the k-level in higher dimensions, see [27].

Chapter 3

Higher-Order Voronoi Diagrams of Line Segments

Among all possible types of polygonal objects, line segments are the simplest, but also the most interesting. Surprisingly, the order-k Voronoi diagram was investigated only for the case of points. In this chapter we are going to fill this gap.

We define the order-k Voronoi diagram of line segments S as a partitioning of the plane into regions, such that each point within a fixed region has the same closest k line segments, similarly to the definition of the order-k Voronoi diagram of points. For instance, the order-k Voronoi region of H can be defined as:

$$V_k(H,S) = \{x \mid \forall s \in H, \forall t \in S \setminus H \; d(x,s) < d(x,t)\}, \tag{3.1}$$

where $H \subset S$ and $|H| = k$.

The partitioning of the plane into order-k Voronoi regions gives the order-k Voronoi diagram of S:

$$\mathcal{V}_k(S) = \bigcup_{H \subset S, |H|=k} \partial V_k(H,S). \tag{3.2}$$

The distance between a point p and a line segment s is measured as the minimum distance, $d(p,s) = \min_{q \in s} d(p,q)$.

Unlike points, line segments have a dimension which allows them to stretch across the plane and exert their influence in multiple places. The dimensionality of the line segments manifests itself in disconnected regions which represents the phenomenon of the same k line segments inducing faces in multiple areas on the plane. In fact, as we will see in Section 3.1, a single order-k Voronoi

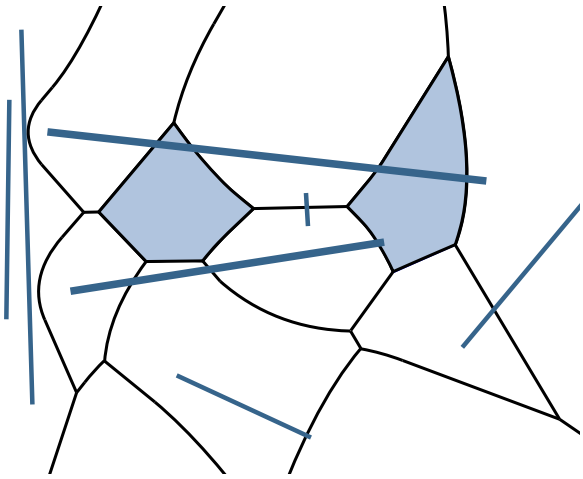

Figure 3.1. The order-2 Voronoi diagram of line segments. The shaded faces correspond to the order-2 Voronoi region induced by the pair of bold line segments.

region may disconnect into $\Omega(n)$ faces, for $k > 1$. Figure 3.1 illustrates an order-2 Voronoi diagram of line segments. Surprisingly, the union of all faces induced by the same segment is a connected set, nevertheless.

The order-k Voronoi diagram of line segments has even more properties different from its counterpart for points.

- In the case of points the faces are convex polygons. In the case of line segments they are not convex and the edges are composed of line segments and parabolic arcs.

- In the case of points the unbounded faces of the order-k Voronoi diagram correspond to the k-sets, see Section 2.4. In the case of line segments we have to use the relation with k-level, which is a more general structure than k-sets, see Section 3.2.

- For points, the relation with k-sets implies the *symmetry property*: the number of unbounded faces of the order-k Voronoi diagram is equal to the number of faces of the order-$(n-k)$ Voronoi diagram. For line segments this property does not hold.

However, despite all these differences, the structural complexity for disjoint line segments is the same as for points, $O(k(n-k))$. This property gives us a reason to think that some of the construction algorithms for points can be extended to handle the case of disjoint line segments, while preserving their

time and space complexity. This however is not trivial and is discussed in detail in every chapter of this dissertation.

Intersecting line segments allow additional complication which is not possible in the case of points. Since each pair of line segments may produce an intersection, the total number of intersections is quadratic. This is expected to bring input in the overall structural complexity of the higher-order Voronoi diagram. For instance, for the nearest neighbor Voronoi diagram the structural complexity is $O(n+I)$, where I is the number of intersections. However, despite the influence of the intersections on the low-order Voronoi diagrams, the influence grows weaker as the order increases. In fact, it converges to linear when approaching the farthest Voronoi diagram. For general k the structural complexity is:

$$\begin{array}{rl} O(k(n-k)+I), & k < n/2 \\ O(k(n-k)), & k \geq n/2 \end{array}$$

The structural properties remain the same even for the general L_p metric, for $1 \leq p \leq \infty$. The cases of L_1 and L_∞ are the corner cases that have additional interesting properties. For instance, in L_1/L_∞ the farthest Voronoi diagram of disjoint line segments has $O(1)$ structural complexity. As an implication, the structural complexity of the order-k Voronoi diagram of disjoint line segments allows tighter bounds:

$$\begin{array}{rl} O(k(n-k)), & k < n/2 \\ O\left((n-k)^2\right), & k \geq n/2 \end{array}$$

The standard method to construct the order-k Voronoi diagram is the iterative method, see Section 3.5. It is simple and efficient if we need to construct all order-i Voronoi diagrams for $i \leq k$. It is also useful when we need to construct only the order-k Voronoi diagram for small values of k. The standard iterative algorithm has $O(k^2 n \log n)$ deterministic time complexity and $O(kn)$ space complexity.

3.1 Properties of Voronoi Regions

Let $S = \{s_1, s_2, \ldots, s_n\}$ be a set of n line segments in $\mathbb{R}^2$. Line segments are assumed disjoint in this section and Section 3.2, but in subsequent sections they may touch at endpoints or intersect. Unless stated otherwise, we make the *general position assumption* (applicable to disjoint line segments) that no more than three sites touch the same circle and no more than two endpoints lie on the same line.

The bisector of two segments s_i and s_j is the locus of points equidistant from both segments, i.e., $b(s_i,s_j) = \{x \mid d(x,s_i) = d(x,s_j)\}$. For two disjoint line segments in the Euclidean plane, $b(s_i,s_j)$ is a curve, which consists of a constant number of line segments, rays, and parabolic arcs. Unlike the bisectors of points, the bisectors of line segments can intersect a multiple number of times or not intersect at all.

The important part of the structural complexity analysis of the higher-order Voronoi diagram is the analysis of the unbounded faces. We use the structural properties of unbounded faces to show the lower bound for the number of disconnected faces of a single Voronoi region. The combinatorial properties of the unbounded faces are very important for the analysis of the structural complexity of the entire diagram, see Section 3.2.

The following lemma gives the main property of the unbounded faces. It is a simple generalization of [9] for $1 \leq k \leq n-1$.

Lemma 3.1.1. *Consider a face F of region $V_k(H,S)$. F is unbounded (in the direction r) iff there exists an open halfplane (normal to r) that intersects all segments in H but no segment in $S \setminus H$.*

Proof. ($\Rightarrow$) Let F be an unbounded face of region $V_k(H,S)$. Let $x \in V_k(H,S)$, and let r be a ray emanating from x to an unbounded direction of the face. Since $x \in V_k(H,S)$, x is the center of the open disk that intersects all segments in H and does not intersect segments in $S \setminus H$. While we move x along r towards infinity, the disk expands until it becomes an open halfplane that intersects all segments in H but no segment in $S \setminus H$. Thus, such a halfplane exists.

($\Leftarrow$) Let h be an open halfplane that intersects all segments in H but no segment in $S \setminus H$. Let h' be the open halfplane h translated parallel to itself until one of the segments $s \in H$ stops intersecting h. At this moment, s touches the boundary of h' at some point x. Consider the ray r in h emanating from x orthogonal to the boundary of h. Let D be a disk centered at an arbitrary point y on r, which intersects all segments in H. Then, $D \subset h$, which means that D does not intersect any segment in $S \setminus H$. Therefore, $y \in V_k(H,S)$. Since the point $y \in r$ was taken arbitrarily, the ray r is entirely enclosed in $V_k(H,S)$, i.e., the Voronoi region is unbounded in this direction. □

Definition 1. *A* supporting halfplane *of segments $s_1,s_2 \in S$ and $H \subseteq S$, where $s_1,s_2 \notin H$, is an open halfplane h whose boundary passes through endpoints of s_1,s_2 (at least one endpoint of each segment), with the property that h intersects all segments in H but no segment in $S \setminus H$.*

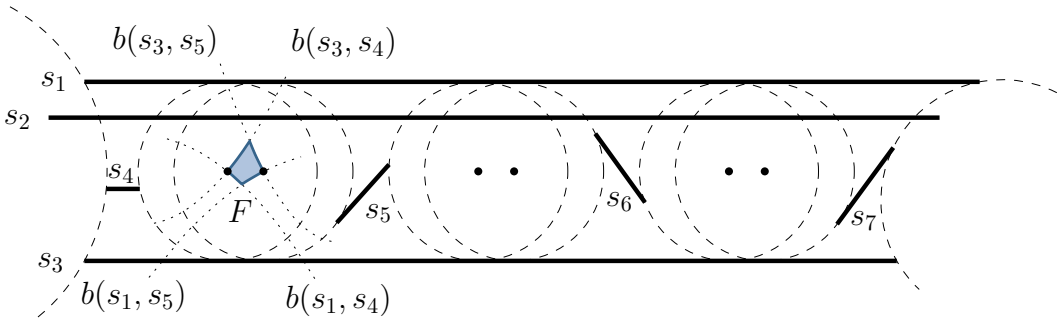

Figure 3.2. The obstacles in between the long segments H induce $n-k+1$ disconnectivities in the region of $V_3(H,S)$, $H = \{s_1, s_2, s_3\}$, $k = 3$. The face $F \subset V_3(H,S)$ is enclosed in between bisectors $b(s_3,s_5)$, $b(s_3,s_4)$, $b(s_1,s_5)$ and $b(s_1,s_4)$.

Corollary 3.1.2. *(of Lemma 3.1.1) There is an unbounded Voronoi edge separating regions $V_k(H \cup \{s_1\}, S)$ and $V_k(H \cup \{s_2\}, S)$ if and only if there is a halfplane supporting s_1, s_2, and H.*

For line segments, a single order-k Voronoi region may be disconnected and it may consist of multiple disjoint faces, unlike its counterpart for points. For example in Figure 3.1, the order-2 Voronoi region of the pair of line segments shown in bold, consists of two faces, which are shown shaded. This phenomenon was first pointed out by Aurenhammer et al [9] for the farthest line segment Voronoi diagram, where a single Voronoi region was shown possible to disconnect into $\Theta(n)$ faces in the worst case.

Lemma 3.1.3. *For $k > 1$, an order-k region of $\mathcal{V}_k(S)$ can have $\Omega(n)$ disconnected faces in the worst case.*

Proof. We first describe an example where an order-k Voronoi region is disconnected into $n-k-1$ bounded and two unbounded faces. Consider a set H of k almost parallel long segments. These segments induce a region $V_k(H,S)$. Consider the minimum disk that intersects all segments in H, and moves along their length. We place the remaining $n-k$ segments of $S \setminus H$ in such a way that they create obstacles for the disk. While the disk moves along the tree of $\mathcal{V}_f(H)$, it intersects the segments of $S \setminus H$ one by one and creates $\Omega(n-k)$ disconnectivities (see Figure 3.2).

In particular, $V_k(H,S)$ has $n-k-1$ bounded and two unbounded faces.

We now follow [9] and describe an example in which an order-k Voronoi region is disconnected into k unbounded faces. Consider $n-k$ segments in

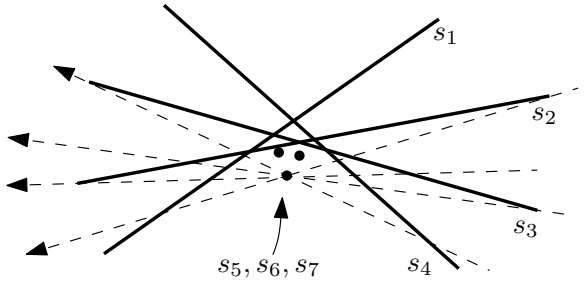

Figure 3.3. During the rotation of the directed line, the positions in which the open halfplane to the left of it intersects all non-degenerate segments, alternate with the positions in which it does not.

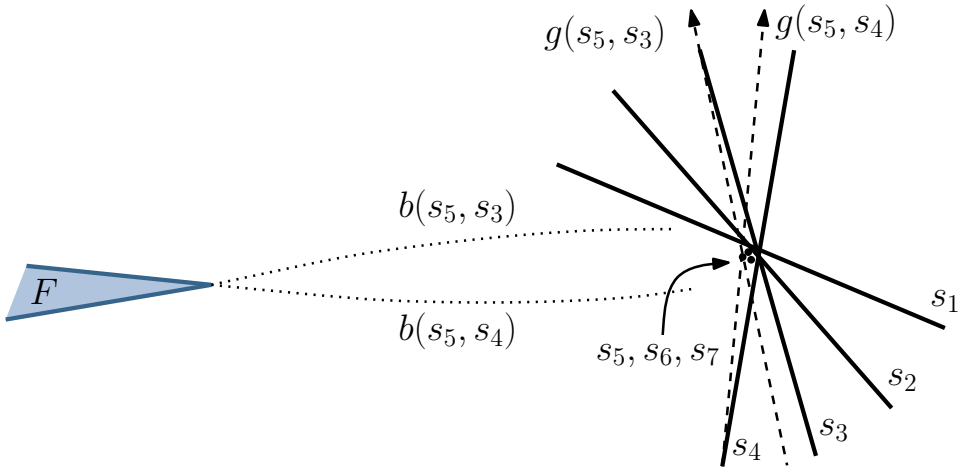

Figure 3.4. The alternations produce bisectors that bound distinct unbounded faces of the region $V_4(H, S)$.

$S \setminus H$, degenerated into points and placed close to each other. The remaining k non-degenerate segments in H are organized in a cyclic fashion around them (see Figure 3.3).

Consider a directed line g through one of the degenerate segments s. Rotate g around s and consider the open halfplane to the left of g. During the rotation, the positions of g, in which the halfplane intersects all k segments, alternate with the positions in which it does not (see Figure 3.3). The positions at which the halfplane touches endpoints of non degenerate segments correspond to unbounded Voronoi edges, such as $g(s_5, s_3)$ and $g(s_5, s_4)$ in Figure 3.4, that define an unbounded Voronoi face of $V_k(H, S)$. Each pair of consecutive unbounded Voronoi edges bounds a distinct unbounded face. Each unbounded edge corresponds to a halfplane that touches an endpoint of a line segment in H. Thus, the

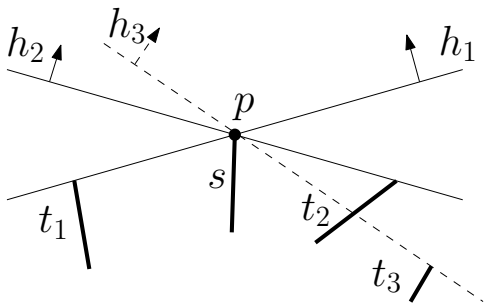

Figure 3.5. Every endpoint of a segment $s \in H$ can induce at most two supporting halfplanes.

number of unbounded faces of $V_k(H,S)$ is $|H| = k$.

For small k, $1 < k < n/2$, the number of faces in the first example $(n-k+1)$ is $\Omega(n)$, while for large k, $n/2 \leq k \leq n-1$, the number of faces in the second example k is also $\Omega(n)$. □

It may seem as if disconnected regions are present because of the crossings between segments; however, this is not the case. In the example of Figure 3.4, we can untangle the segments to form a non-crossing configuration, while the same phenomena remain. Consider a segment $s \in H$ whose endpoints define two supporting halfplanes. We can move the endpoints of s along the boundaries of the halfplanes away from the rest of the line segments in H, and untangle all line segments in H, while maintaining the same halfplanes that define the corresponding unbounded Voronoi edges. For $k = n-1$, this was illustrated in [9].

Lemma 3.1.4. *An order-k region $V_k(H,S)$ has $O(k)$ unbounded disconnected faces.*

Proof. We show that an endpoint p of a segment $s \in H$ may induce at most two unbounded Voronoi edges bordering $V_k(H,S)$ (see Figure 3.5).

Consider two such unbounded Voronoi edges. By Corollary 3.1.2, there are open halfplanes h_1, h_2, such that the boundary of h_1 and h_2 pass through point p and the endpoints of the line segments t_1 and t_2, respectively. The open halfplanes h_1 and h_2 intersect all line segments in H and do not intersect line segments in $S \setminus H$. Thus, any other supporting halfplane h_3, with boundary passing through point p and an endpoint of some line segment $s_3 \in S \setminus H$, must intersect either t_1 or t_2. Since $|H| = k$ and a segment has two endpoints, the claim follows. □

Although an order-k Voronoi region may be disconnected, the union of all

faces induced by a segment s is a connected region which encloses s. In particular, let

$$VN_k(s,S) = \bigcup_{H \subset S, s \in H} \overline{V_k(H,S)},$$

where $\overline{X}$ denotes the topological closure of the set X.

A set X is said to be *weakly star-shaped* with respect to a line segment s if for every point $x \in X$ there exists a point $y \in s$, such that the line segment xy is entirely enclosed in X.

Lemma 3.1.5. *Consider the order-k Voronoi diagram $\mathcal{V}_k(S)$. The union of all faces in $\mathcal{V}_k(S)$ affiliated with a segment s, $VN_k(s,S)$, is weakly star-shaped with respect to s ($s \in VN_k(s,S)$).*

Proof. Let x be an arbitrary point in $VN_k(s,S)$. Denote by $D_k(x)$ the minimum disk, centered at x, that intersects at least k line segments, and by $D_s(x)$ the minimum disk, centered at x, that touches the line segment s. Since $x \in VN_k(s,S)$, x must be in one of the regions $V_k(H,S)$, where $s \in H$. Therefore, $D_s(x) \subseteq D_k(x)$.

Let y be a point on the line segment s that is closest to x. Consider an arbitrary point a on the line segment xy. Then, $D_s(a) \subseteq D_s(x) \subseteq D_k(x)$. This implies that the line segment s is the ith-closest line segment from point a, where $i \leq k$. Therefore, $a \in VN_k(s,S)$. Since a is taken arbitrarily, the entire line segment xy is enclosed in $VN_k(s,S)$. □

3.2 Structural Properties and Complexity

In this section we show structural properties of the order-k Voronoi diagram of n disjoint line segments and prove that its combinatorial complexity is $O(k(n-k))$, despite the presence of disconnected regions.

We first prove Theorem 3.2.7, which is a generalization to line segments of the formula in [59, Theorem 2], which counts the total number of faces of $\mathcal{V}_k(S)$ as a function of n, k, and the number of unbounded faces. To this aim, we exploit the fact that the farthest line-segment Voronoi diagram is a tree structure [9]. Then in Lemma 3.2.8, we analyze the number of unbounded faces in the order-k Voronoi diagram in a dual setting (see e.g. [20]) by using the results on arrangements of wedges [9, 40] and ($\leq k$)-level in arrangements of Jordan curves [84]. We derive the result by combining Theorem 3.2.7 and Lemma 3.2.8.

The definition of an order-k Voronoi region implies that two adjacent order-k Voronoi faces must differ in exactly two sites. Therefore, any point on a Voronoi edge separating two faces, must be the center of a disk that intersects $k+1$ and

touches two line segments. Under the general position assumption, an order-k Voronoi vertex v is incident to three Voronoi edges and to three faces. There are two cases [59]:

1. The incident order-k Voronoi regions are $V_k(H \cup \{a\}, S)$, $V_k(H \cup \{b\}, S)$, $V_k(H \cup \{c\}, S)$;

2. The incident order-k Voronoi regions are $V_k(H \cup \{a, b\}, S)$, $V_k(H \cup \{b, c\}, S)$, $V_k(H \cup \{c, a\}, S)$.

In the first case, $|H| = k - 1$ and v is called a *new* order-k Voronoi vertex. In the second case, $|H| = k - 2$ and v is called an *old* order-k Voronoi vertex. In both cases, v is the center of the disk whose interior intersects all the line segments in H, and whose boundary touches the line segments a, b and c. Thus, Voronoi vertices in $\mathcal{V}_k(S)$ are classified into *new* and *old*. A new Voronoi vertex in $\mathcal{V}_k(S)$ is an old Voronoi vertex in $\mathcal{V}_{k+1}(S)$, and it appears for the first time in the order-k diagram. Under the general position assumption, an *old* Voronoi vertex in $\mathcal{V}_k(S)$ is a *new* Voronoi vertex in $\mathcal{V}_{k-1}(S)$.

Lemma 3.2.1. *Consider a face F of the region $V_{k+1}(H, S)$ ($|H| = k + 1$). The portion of $\mathcal{V}_k(S)$ enclosed in F is exactly the portion of the farthest Voronoi diagram $\mathcal{V}_f(H)$ enclosed in F.*

Proof. Let x be a point in F. Suppose x belongs to the region $V_k(H_j, S)$ of $\mathcal{V}_k(S)$. Then H_j is the set consisting of the k line segments closest to x. Let $\{s_j\} = H \setminus H_j$; then s_j is the k+1-closest line segment to x. Therefore, s_j is the line segment farthest from x, among all segments in H. Therefore, $x \in V_f(s_j, H)$.

Suppose x belongs to the edge separating regions $V_k(H_j, S)$ and $V_k(H_r, S)$ of $\mathcal{V}_k(S)$. Then we can show in a similar way that x belongs to the edge separating farthest regions $V_f(s_j, H)$ and $V_f(s_r, H)$, where $\{s_j\} = H \setminus H_j$ and $\{s_r\} = H \setminus H_r$. □

Consider a region $V_f(s, H)$ of the farthest line-segment Voronoi diagram, $\mathcal{V}_f(H)$. This region has the following *visibility property*, see Figure 3.6.

Lemma 3.2.2 (Visibility property in a farthest Voronoi region)**.** *Let x be a point in a farthest Voronoi region $V_f(s, H)$ of $\mathcal{V}_f(H)$. Let $r(s, x)$ be the ray realizing the distance $d(s, x)$, emanating from point $p \in s$ such that $d(p, x) = d(s, x)$, and extending to infinity. The ray $r(s, x)$ must intersect the boundary of $V_f(s, H)$ at a point a_x, and the unbounded portion of $r(s, x)$ beyond a_x must lie entirely in $V_f(s, H)$.*

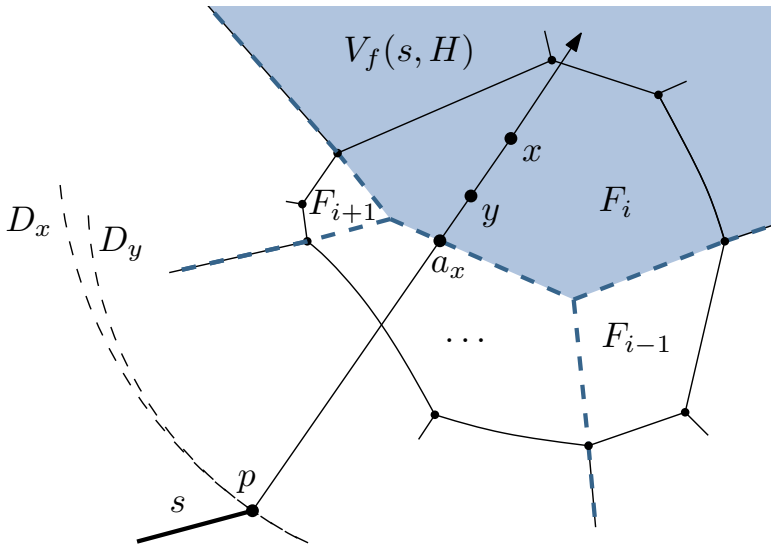

Figure 3.6. The part of the ray $r(s,x)$ beyond a_x entirely belongs to $V_f(s,H)$.

Proof. Consider a point y along $r(s,x)$, which is a slight translation of the point x towards p. Let D_x (resp., D_y) be the minimum disk centered at x (resp., y), that intersects all segments in H. Then, $D_y \subset D_x$. The disk D_y intersects all segments in H and touches s at point p, which implies that $y \in V_f(s,H)$. If we continue to move y towards p, the disk D_y will eventually touch some segment in $H \setminus \{s\}$, at position $y = a_x$. Therefore, the point a_x belongs to an edge of the farthest Voronoi diagram $\mathcal{V}_f(H)$. Now, if we move y, starting from x and away from p, then the disk D_y will continue to contain D_x and touch s. Therefore, the part of the ray $r(s,x)$ beyond a_x must entirely belong to $V_f(s,H)$. □

Using this property, we derive the following lemma.

Lemma 3.2.3. *Let F be a face of a region $V_{k+1}(H,S)$ in $\mathcal{V}_{k+1}(S)$. The graph structure of $\mathcal{V}_k(S)$ enclosed in F is a tree that consists of at least one edge. Each leaf of the tree is incident to an old Voronoi vertex on the boundary of F (see Figure 3.6).*

Proof. Consider a point x in F (see Figure 3.6) and let s be the segment in H farthest away from x. Consider the ray $r(s,x)$, and the point a_x as defined in Lemma 3.2.2. Lemma 3.2.2 implies that a_x is a point in the interior of F, therefore, F must contain a portion of the tree of $\mathcal{V}_f(H)$, and, thus Lemma 3.2.1 implies that F must contain at least one edge of $\mathcal{V}_k(S)$.

Now we prove that the portion of $\mathcal{V}_k(S)$ enclosed in F is connected. Lemma 3.2.1 implies that this portion is equal to the portion of $\mathcal{V}_f(H)$ enclosed in F. Assume, to the contrary, that the portion of $\mathcal{V}_f(H)$ enclosed in F is disconnected. Then,

there is a subface F_i of F that separates two disconnected subtrees of $\mathcal{V}_f(H)$, say, T_1 and T_2. Let $F_i \subseteq V_f(s,H)$, v be a point on the boundary of $V_f(s,H)$ between T_1 and T_2, and $r(s,v)$ be the ray that realizes the distance from s to v extending to infinity. The *visibility property* of $V_f(s,H)$ in Lemma 3.2.2 implies that the portion of $r(s,v)$ beyond v belongs entirely to $V_f(s,H)$. Since T_1 and T_2 bound F_i, the ray $r(s,v)$ must intersect F_i beyond the point v. Consider the minimum disk centered at v, that intersects all segments in H. The disk must also intersect some segments in $S \setminus H$ because v does not belong to F. If we move the center of the disk along $r(s,v)$ away from s, the new minimum disk will contain the previous disk, and, therefore, it will also intersect the same segments in $S \setminus H$. Thus, no portion of $r(s,v)$ can be in F, which is a contradiction. □

Corollary 3.2.4. *Consider a face F of the Voronoi region $V_{k+1}(H,S)$. Let m be the number of Voronoi vertices in the portion of $\mathcal{V}_k(S)$ enclosed in the interior of F. Then, F encloses $2m+1$ Voronoi edges of $\mathcal{V}_k(S)$.*

Let F_k, E_k, V_k, and U_k denote the number of faces, edges, vertices and unbounded faces in $\mathcal{V}_k(S)$ respectively. By the Euler's formula we derive the following lemma.

Lemma 3.2.5.

$$E_k = 3(F_k - 1) - U_k \tag{3.3}$$

$$V_k = 2(F_k - 1) - U_k \tag{3.4}$$

Proof. Consider $\mathcal{V}_k(S)$ and connect every unbounded edge with an artificial point at infinity. Then Euler's formula implies that $F_k - E_k + V_k = 1$.

Consider the dual graph of $\mathcal{V}_k(S)$. Connect every vertex of the dual graph, representing an unbounded face of $\mathcal{V}_k(S)$, with an artificial point at infinity. Then, under the general-position assumption, every face in the dual graph must have exactly three edges, and every edge is adjacent to exactly two faces. Therefore, $3(V_k + U_k) = 2(E_k + U_k)$. The combination of these equations proves the lemma. □

Lemma 3.2.6. *The total number of unbounded faces in the order-k Voronoi diagram of all orders is*

$$\sum_{i=1}^{n-1} U_i = n(n-1).$$

Proof. Consider an arbitrary pair of segments s_1 and s_2. There are exactly two open halfplanes r_1 and r_2 that touch s_1 and s_2. Corollary 3.1.2 implies that these

open halfplanes define unbounded Voronoi edges for some order-(k_1+1) and order-(k_2+1) Voronoi diagrams, where k_1 and k_2 are the numbers of segments that r_1 and r_2 intersect, respectively. In addition, any unbounded Voronoi edge is induced by such a halfplane. Thus, $\sum_{i=1}^{n-1} U_i = 2\binom{n}{2} = n(n-1)$. □

Theorem 3.2.7. *The number of faces in the order-k Voronoi diagram of n disjoint line segments is*

$$F_k = 2kn - k^2 - n + 1 - \sum_{i=1}^{k-1} U_i \tag{3.5}$$

$$\textit{or, equivalently,} \quad F_k = 1 - (n-k)^2 + \sum_{i=k}^{n-1} U_i \tag{3.6}$$

Proof. Let V_k, V_k' and V_k'' be the number of Voronoi vertices, *new* Voronoi vertices, and *old* Voronoi vertices in $\mathcal{V}_k(S)$, respectively. (Notation follows [59].) Then, $V_k = V_k' + V_k'' = V_k' + V_{k-1}'$.

Following [59], we obtain a recursive formula for the number of faces F_k of the order-k Voronoi diagram. Assuming that segments do not intersect, $F_1 = n$, since each segment induces exactly one face in $\mathcal{V}_1(S)$. In $\mathcal{V}_2(S)$, each face encloses exactly one edge of $\mathcal{V}_1(S)$, thus, $F_2 = E_1$. Then by Lemma 3.2.5 we derive $F_2 = 3(F_1 - 1) - U_1$, thus, $F_2 = 3(n-1) - U_1$.

We now prove that $F_{k+2} = E_{k+1} - 2V_k'$ (*Claim 1*). Note that $V_1' = V_1$ and $V_1 = 2(n-1) - U_1$ (using Eq. (3.4) of Lemma 3.2.5). The definition of *old* Voronoi vertices implies that *old* Voronoi vertices of $\mathcal{V}_{k+1}(S)$ lie in the interior of the faces of $\mathcal{V}_{k+2}(S)$. Consider a face F_i of $\mathcal{V}_{k+2}(S)$. Let m_i be the number of *old* Voronoi vertices of $\mathcal{V}_{k+1}(S)$ enclosed in the interior of F_i. Then, F_i encloses $e_i = 2m_i+1$ Voronoi edges of $\mathcal{V}_{k+1}(S)$ (see Corollary 3.2.4). Summing up the numbers of all faces in $\mathcal{V}_{k+2}(S)$, we obtain that $\sum_{i=1}^{F_{k+2}} e_i = 2\sum_{i=1}^{F_{k+2}} m_i + F_{k+2}$. However, $\sum_{i=1}^{F_{k+2}} m_i = V_{k+1}'' = V_k'$ and $\sum_{i=1}^{F_{k+2}} e_i = E_{k+1}$. Therefore, $F_{k+2} = E_{k+1} - 2V_k'$, and, Claim 1 follows.

We now use Claim 1 to obtain a recursive formula for F_k. Summing up F_{k+2} and F_{k+3}, we obtain $F_{k+3} = E_{k+2} + E_{k+1} - F_{k+2} - 2V_{k+1}' - 2V_k' = E_{k+2} + E_{k+1} - F_{k+2} - 2V_{k+1}$. We then substitute Eqs. (3.3) and (3.4) in the last formula and obtain

$$F_{k+3} = 2F_{k+2} - F_{k+1} - 2 - U_{k+2} + U_{k+1}. \tag{3.7}$$

where, $F_1 = n$ and $F_2 = 3(n-1) - U_1$. Because $F_2 = E_1$, Eq. (3.7) can also be derived for F_3, i.e. the formula applies to $k \geq 0$.

By induction, using Eq. (3.7) and the above base cases, we derive Eq. (3.5). Lemma 3.2.6 implies that $\sum_{i=1}^{k-1} U_i + \sum_{i=k}^{n-1} U_i = \sum_{i=1}^{n-1} U_i = n(n-1)$. Combining this result with Eq. (3.5), we derive Eq. (3.6). □

Lemma 3.2.8. *Given a set S of n line segments,*

$$\sum_{i=k}^{n-1} U_i = O(n(n-k)).$$

Proof. We use the well-known point-line duality transformation T that maps a point $p = (a, b)$ in the primal plane to a line $T(p) : y = ax - b$ in the dual plane, and vice versa (see [9]). We call the set of points above both lines $T(p)$ and $T(q)$ the *wedge* of $s = (p, q)$. Consider a line ℓ and a segment $s = (p, q)$. The segment s is above the line ℓ if and only if the point $T(\ell)$ is strictly above lines $T(p)$ and $T(q)$ [9].

Consider the arrangement W of the wedges w_i, $i = 1, \ldots, n$, corresponding to the segments in $S = \{s_1, \ldots, s_n\}$. For our purposes in this section, the complexity of the r-level and the $(\leq r)$-level is the number of their vertices, excluding the wedge apices. We denote the maximum complexity of the r-level and the $(\leq r)$-level of n wedges by $g_r(n)$ and $g_{\leq r}(n)$, respectively. We first prove the following claim.

Claim 3.2.9. *The number of unbounded Voronoi edges of $V_k(S)$, unbounded in direction $\phi \in [\pi, 2\pi]$, is exactly the number of vertices shared by the $(n-k-1)$-level and the $(n-k)$-level of W. Thus, $U_k = O(g_{n-k-1}(n))$.*

*P*roof of Claim. Let s_i, s_j be two line segments that define an unbounded bisector in a direction $\phi \in [\pi, 2\pi]$. Then, there is a line ℓ passing through their endpoints, such that the open halfplane ℓ^- below ℓ intersects $k-1$ line segments and does not intersect s_i nor s_j. Then, ℓ passes strictly below $n-(k-1)-2 = n-k-1$ line segments. Thus, ℓ corresponds to a point p in the arrangement of wedges shared by the $(n-k-1)$-level and $(n-k)$-level (see Figure 3.7). By the above claim

$$\sum_{i=k}^{n-1} U_i = O(g_{\leq n-k-1}(n)). \tag{3.8}$$

Since the arrangement of wedges is a special case of arrangements of Jordan curves, we use a formula from [84] to bound the complexity of the $(\leq r)$-level in such an arrangement:

$$g_{\leq r}(n) = O\left((r+1)^2 g_0\left(\left\lfloor \frac{n}{r+1} \right\rfloor\right)\right) \tag{3.9}$$

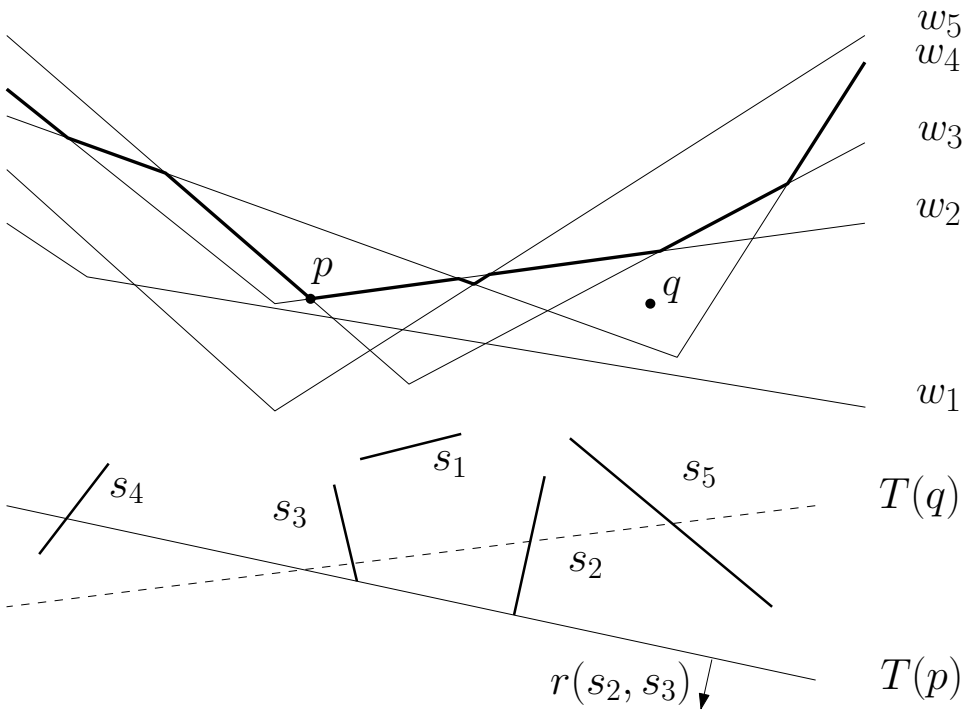

Figure 3.7. (a) In the dual plane, the point p belongs to the 2-level and the 3-level of the arrangement W; (b) In the primal plane, the halfplane $r(s_2, s_3)$ below $T(p)$ defines the unbounded Voronoi edge that separates $V_2(\{s_2, s_4\}, S)$ and $V_2(\{s_3, s_4\}, S)$.

The complexity of the lower envelope of such wedges $g_0(n)$ is $O(n)$ [9, 40]. (In [84] one can find the weaker bound $g_0(n) = O(n \log n)$). Therefore, $g_{\leq r}(n) = O(n(r+1))$. By substituting this into Eq. (3.8) we obtain that $\sum_{i=k}^{n-1} U_i = O(n(n-k))$. □

By combining Lemma 3.2.8 and Theorem 3.2.7, we obtain the following result.

Theorem 3.2.10. *The combinatorial complexity of the order-k Voronoi diagram of n disjoint line segments is*

$$F_k = O(k(n-k)).$$

Proof. For $1 \leq k < n/2$, Eq. (3.5) implies that $F_k = O(k(n-k))$.

For $n/2 \leq k \leq n-1$, Lemma 3.2.8 implies that $\sum_{i=k}^{n-1} U_i = O(n(n-k)) = O(k(n-k))$. The dual formula (3.6) implies that $F_k = 1 - (n-k)^2 + \sum_{i=k}^{n-1} U_i \leq \sum_{i=k}^{n-1} U_i$, which is $O(k(n-k))$. □

3.3 Intersecting Line Segments

In this section we extend our complexity results of Section 3.2 to intersecting line segments with a total of I intersection points, $I = O(n^2)$. We show that segment-intersections influence the Voronoi diagram for small k and the influence grows weaker as k increases. For $k \geq n/2$, intersections no longer affect the asymptotic complexity of the order-k Voronoi diagram.

In the following, we extend Lemma 3.2.6, Theorem 3.2.7, and Theorem 3.2.10 to intersecting line segments as Lemma 3.3.1, Theorem 3.3.2, and Theorem 3.3.3, respectively. To simplify the analysis, we assume that no two segments share a common endpoint and that no more than two segments intersect at the same point. Recall that the numbers of faces, edges, vertices, and unbounded faces of $\mathcal{V}_k(S)$ are denoted as F_k, E_k, V_k, and U_k, respectively.

Lemma 3.3.1. *The total number of unbounded faces in the order-k Voronoi diagram for all orders is*

$$\sum_{i=1}^{n-1} U_i = n(n-1) + 2I.$$

Proof. Consider a pair of line segments. If the pair does not intersect, then it defines exactly two open halfplanes, such that each halfplane induces exactly one unbounded Voronoi edge in $\mathcal{V}_k(S)$ for some order k (see Lemma 3.2.6). If the pair intersects, then it induces exactly four such unbounded Voronoi edges. Thus, each pair of intersecting segments induces exactly two additional unbounded Voronoi edges, in addition to those counted in Lemma 3.2.6. Therefore, the total number of unbounded faces in all orders is $\sum_{i=1}^{n-1} U_i = 2\binom{n}{2} + 2I = n(n-1) + 2I$. □

Theorem 3.3.2. *The number of faces in the order-k Voronoi diagram of a set S of n line segments with I intersections is:*

$$F_k = 2kn - k^2 - n + 1 - \sum_{i=1}^{k-1} U_i + 2I \tag{3.10}$$

$$\text{or equivalently } F_k = 1 - (n-k)^2 + \sum_{i=k}^{n-1} U_i \tag{3.11}$$

Proof. Consider the partitioning of segments into pieces as obtained by their intersection points. Every component of a segment induces exactly one face in $\mathcal{V}_1(S)$, thus, $\mathcal{V}_1(S)$ has two types of vertices: (1) I *intersection* points, which are incident to exactly four Voronoi edges each; and (2) $V_1 - I$ *regular* Voronoi

vertices, which are incident to three Voronoi edges each (under the general position assumption). Regular Voronoi vertices are the *new vertices* of $\mathcal{V}_1(S)$, thus, $V_1' = V_1 - I$.

Consider the dual graph of $\mathcal{V}_1(S)$, augmented with a vertex at infinity to connect the dual of unbounded faces. Using standard arguments, $2E_1 = 4I + 3(V_1 - I) + U_1$ (see also the proof of Theorem 3.2.10). Note that the dual graph consists of faces of four edges each that correspond to *intersections*, and faces of three edges each that correspond to *regular* Voronoi vertices of $\mathcal{V}_1(S)$. Euler's formula and the latter equation imply $E_1 = 3(F_1 - 1) - U_1 - I$. Thus, $E_1 = 3n - 3 - U_1 + 5I$. By Euler's formula, $V_1 = 1 + E_1 - F_1 = 1 + E_1 - n - 2I$, thus, $V_1 = 2n - 2 - U_1 + 3I$.

Consider now $\mathcal{V}_2(S)$, which has two types of faces: faces that contain exactly one edge of $\mathcal{V}_1(S)$ and faces that contain an *intersection* point of $\mathcal{V}_1(S)$. As a result, the total number of faces in $\mathcal{V}_2(S)$ is $F_2 = (E_1 - 4I) + I = E_1 - 3I$. Therefore, $F_2 = 3(F_1 - 1) - U_1 - 4I = 3(n-1) - U_1 + 2I$. Since all Voronoi vertices of $\mathcal{V}_2(S)$ have degree three, Lemma 3.2.5 implies that $E_2 = 3F_2 - 3 - U_2$. Plugging in the formula for F_2, we obtain $E_2 = 9n - 12 - 3U_1 - U_2 + 6I$.

For an order i-diagram, $i \geq 3$, every vertex of the diagram and every vertex of the farthest subdivision is incident to exactly 3 edges, and thus, Claim 1 in the proof of Theorem 3.2.7 and its proof remain identical. Thus, the recursive formula of Eq. (3.7) remains valid for any $k \geq 1$.

Using Claim 1 of Theorem 3.2.7, $F_3 = E_2 - 2V_1' = E_2 - 2(V_1 - I)$. Plugging in the formulas obtained for E_2 and V_1, we obtain $F_3 = 5n - 8 - U_1 - U_2 + 2I$.

Since the recursive formula in Eq. (3.7) remains valid for any $k \geq 1$, we can use induction, with bases cases the above formulas for F_2 and F_3, and derive Eq. (3.10). Note that the main difference with the derivation of Theorem 3.2.7 are the base cases F_1, F_2, and F_3, where F_3 is no longer obtained by Eq. (3.7). Then Eq. (3.11) can be derived from Eq. (3.10) using Lemma 3.3.1. □

Theorem 3.3.3. *The combinatorial complexity of the order-k Voronoi diagram of n properly intersecting line segments with I intersections is*

$$O(k(n-k)+I), \quad \text{for } 1 \leq k < n/2;$$
$$O(k(n-k)), \quad \text{for } n/2 \leq k \leq n-1.$$

Proof. For $1 \leq k < n/2$, Eq. (3.10) of Theorem 3.3.2 directly implies $F_k = O(k(n-k)+I)$. The proof of Lemma 3.2.8 remains valid for any set of arbitrary line segments, including intersecting ones. Thus, for $n/2 \leq k \leq n-1$, Eq. (3.11) of Theorem 3.3.2 and Lemma 3.2.8 imply $F_k = O(k(n-k))$. □

3.4 Extending to the L_p Metric

The results of Sections 3.2 and 3.3, extend naturally to the general L_p, $1 \leq p \leq \infty$, metric.

Consider the disk $n(x,r)$ centered at point x of radius r in the L_p metric, $n(x,r) = \{y \mid d(x,y) \leq r\}$, where $d(x,y) = \left(|x_1 - y_1|^p + |x_2 - y_2|^p\right)^{1/p}$ and $x = (x_1, x_2)$, $y = (y_1, y_2)$. As the radius r tends to the infinity the disk tends to become a halfplane [57], for $1 < p < \infty$. This observation implies that any observations on supporting halfplanes and the unbounded order-k Voronoi regions remain identical for any p, $1 < p < \infty$, see Lemma 3.1.1, Def. 1 and Corollary 3.1.2. Similarly, Lemmas 3.2.6 and 3.3.1 also remain the indentical.

The results of Lemmas 3.2.1-3.2.3 in Section 3.2 are proved using the techniques of expanding the moving disks. The following lemma shows the key observation used in the proofs of these lemmas.

Lemma 3.4.1. *Let x, y, z be the three points placed from left to right along the line. Consider the disk $n(y, d(y,x))$ centered at the point y and the boundary passing through the point x. Consider the disk $n(z, d(z,x))$ centered at the point z and the boundary passing through the point x. Then the former disk is enclosed in the latter, i.e. $n(y, d(y,x)) \subset n(z, d(z,x))$.*

Proof. The proof is straightforward if one uses the property that the disk $n(x,r)$ is convex in L_p metric, $1 \leq p \leq \infty$. □

Lemma 3.4.1 explicitly uses the fact that for $1 \leq p \leq \infty$ the metric is convex and it does not hold for $p < 1$. Thus the results of Lemmas 3.2.1-3.2.3 hold in general L_p metric, for $1 \leq p \leq \infty$. Therefore, the formulas of Theorem 3.2.7 and the $O(k(n-k))$ complexity bound of Theorem 3.2.10 remain the same in L_p for $1 < p < \infty$.

Similarly for Lemma 3.3.1, and Theorems 3.3.2, 3.3.3, in case of intersecting line segments. Thus, all structural properties of the order-k Voronoi diagram in the Euclidean metric remain the same in L_p, for $1 < p < \infty$.

In the remaining of this section, we extend our results to the L_∞ metric (equiv. L_1). In the L_∞ metric, as the radius r tends to the infinity, the disk $n(x,r)$ tends to become a *quadrant*. A *quadrant* is the common intersection of two halfplanes, which are defined by axis parallel perpendicular lines. Therefore in L_∞ metric, the equivalent of a supporting halfplane (see Def. 1) is a *supporting quadrant*. Thus, Corollary 3.1.2 is adapted as follows: There is un unbounded Voronoi edge separating the L_∞ unbounded regions $V_k(H \cup \{s_1\}, S)$ and $V_k(H \cup \{s_2\}, S)$ if and only if there is an open quadrant that touches s_1 and s_2, intersects

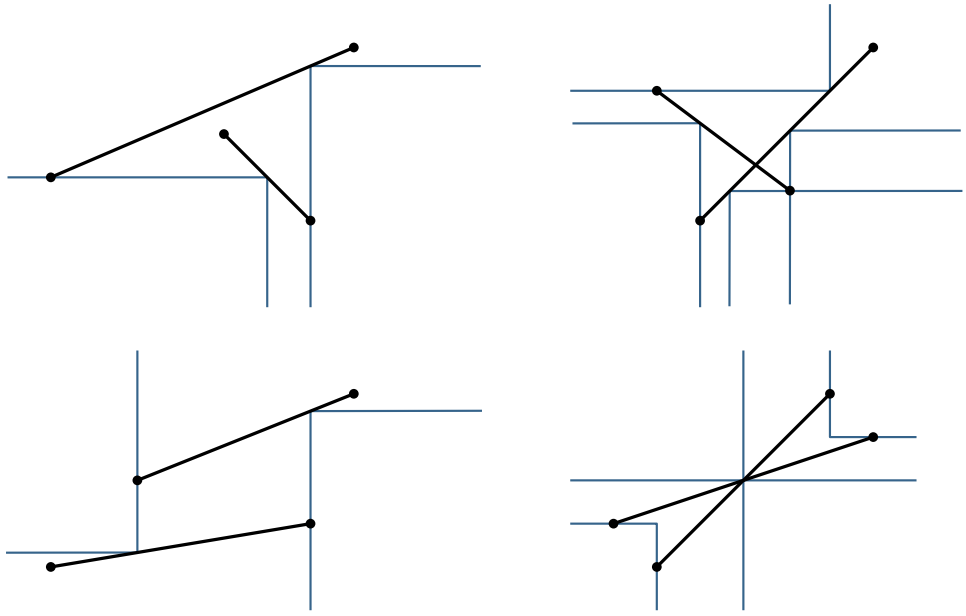

Figure 3.8. Examples of supporting quadrants of pairs of line segments in the L_∞ metric.

all line segments in H, but no line segment in $S \setminus H$. Such a quadrant is called a *supporting quadrant* (see e.g., Figure 3.8).

In L_∞, a pair of disjoint line segments admits two supporting quadrants and a pair of intersecting line segments admits four supporting quadrants. Thus, Lemmas 3.2.6 and 3.3.1 remain valid. We now extend Lemma 3.2.8 to the L_∞ metric.

Lemma 3.4.2. *In L_∞ (resp. L_1), for a given set of n line segments,*

$$\sum_{i=k}^{n-1} U_i = O(n(n-k)).$$

If segments are disjoint then

$$\sum_{i=k}^{n-1} U_i = O\left((n-k)^2\right).$$

Proof. The duality transformation in the proof of Lemma 3.2.8 is not extendible to the L_∞ metric. Instead, we use the abstract framework presented in [30, 31, 82].

Let a supporting quadrant be called a *configuration*. A configuration is defined by two line segments s_1 and s_2 if there is a quadrant whose boundary touches s_1, s_2 and its interior does not intersect s_1, s_2. A configuration is said to be *in conflict* with line segment s' if its supporting quadrant does not intersect s'. The *weight* of a configuration is the number of its conflicts. The maximum number of configurations of weight i in a set of n line segments is denoted as $N_i(n)$, and the maximum number of configurations of weight at most i is denoted as $N_{\leq i}(n)$. The configurations with weight i correspond to unbounded Voronoi edges in the order-$(n-i-1)$ Voronoi diagram, thus $U_{n-i-1} \leq N_i(n)$. The configurations with weight 0 correspond to unbounded edges in the farthest Voronoi diagram. The Clarkson-Shor abstract framework implies $N_{\leq i}(n) = O\left(i^2 N_0(n/i)\right)$. Substituting $i = n-k-1$, we derive

$$\sum_{i=k}^{n-1} U_i \leq N_{\leq n-k-1}(n) = O\left((n-k-1)^2 N_0\left(\frac{n}{n-k-1}\right)\right) \tag{3.12}$$

In L_∞, $N_0(n)$ is $O(n)$ for arbitrary line segments, and $O(1)$ for non-crossing line segments [34, 70]. Substituting these values in Eq. (3.12), we derive $\sum_{i=k}^{n-1} U_i = O(n(n-k))$ for arbitrary line segments, and $\sum_{i=k}^{n-1} U_i = O\left((n-k)^2\right)$ for non-crossing line segments. □

Using Lemma 3.4.2 in place of Lemma 3.2.8, we can extend the proofs of Theorems 3.2.10 and 3.3.3 to the L_∞ metric in a straightforward way. For non-crossing line segments, Lemma 3.4.2 directly implies a tighter bound. The same tighter bound was shown for points in [60] by a different derivation based on a *Hanan grid*, which is not applicable to line segments. We summarize in the following theorem.

Theorem 3.4.3. *The structural complexity of order-k Voronoi diagram of n arbitrary line segments, with I intersections, in the L_p metric, $1 \leq p \leq \infty$, is:*

$$\begin{aligned} O(k(n-k)+I), &\quad \text{for } 1 \leq k < n/2; \\ O(k(n-k)), &\quad \text{for } n/2 \leq k \leq n-1; \\ O\left((n-k)^2\right), &\quad \text{for } n/2 \leq k \leq n-1, \text{ non-crossing segments and } p = 1, \infty. \end{aligned}$$

3.5 Iterative Construction

To compute the diagram, we can use the standard iterative approach to construct higher-order Voronoi diagrams (see e.g., [59]). The iterative construction

is basic, and although not very efficient for arbitrary k, it can be valuable to applications, where lower order diagrams are required in any case, see Section 1.1.1.

The iterative construction can be described as follows:

- Construct $\mathcal{V}_1(S)$ using any available algorithm with $O(n \log n)$ time complexity.
- For $i = 1, \ldots, k-1$ do:
  - For every face F of every region $V_i(H,S)$ of $\mathcal{V}_i(S)$ compute the part of $\mathcal{V}_1(S \setminus H)$ enclosed within F.
  - Remove/Disregard the edges of $\mathcal{V}_i(S)$.

Given a face F of region $V_i(H,S)$, let S_F denote the collection of segments in $S \setminus H$ that define edges along the boundary of F, ∂F. Let $\mathcal{V}_1(F)$ denote the portion of $\mathcal{V}_1(S_F)$ in the interior of F. By the definition of an order-$(i+1)$ region, $\mathcal{V}_1(F)$ corresponds exactly to $\mathcal{V}_1(S \setminus H)$ within F. The main operation of the iterative construction is to compute $\mathcal{V}_1(F)$. Figure 3.9 illustrates an unbounded face F and its internal subdivision by $\mathcal{V}_1(F)$. If F is unbounded, $\mathcal{V}_1(F)$ is augmented with an artificial point at infinity, which is assumed to be incident to all unbounded Voronoi edges, see Figure 3.12.

Because order-k Voronoi regions may be disconnected, a segment $s \in S_F$ may appear multiple times along ∂F. However, the appearances of the line segments along ∂F form an order-2 Davenport-Schinzel sequence (DSS) and therefore the complexity of ∂F is linear on $|S_F|$, see the following lemma.

Lemma 3.5.1. *The appearances of disjoint line segments in S_F along ∂F form an order-2 DSS. The appearances of intersecting line segments in S_F along ∂F form an order-4 DSS.*

Proof. Consider a pair of disjoint line segments s_a and s_b. Denote as a and b their appearances along ∂F. Then we want to prove that there is no a, b, a, b subsequence along ∂F, see Figure 3.9. Suppose there is a subsequence a, b, a then we prove that b cannot occur another time after a. Consider line segment s_a and consider a pair of points x_1 and x_2 on ∂F that belong to the left and right appearance of s_a in a, b, a sequence. Let $y_1, y_2 \in s_a$ be a pair of points closest to points x_1 and x_2, respectively. Consider a pair of line segments x_1y_1 and x_2y_2. Points x_1, x_2 correspond to the appearances of s_a and line segments x_1y_1, x_2y_2 realize the shortest distance to s_a. Thus line segments x_1y_1, x_2y_2 do not intersect the line segment s_b. The line segment s_b does not also intersect the line segment

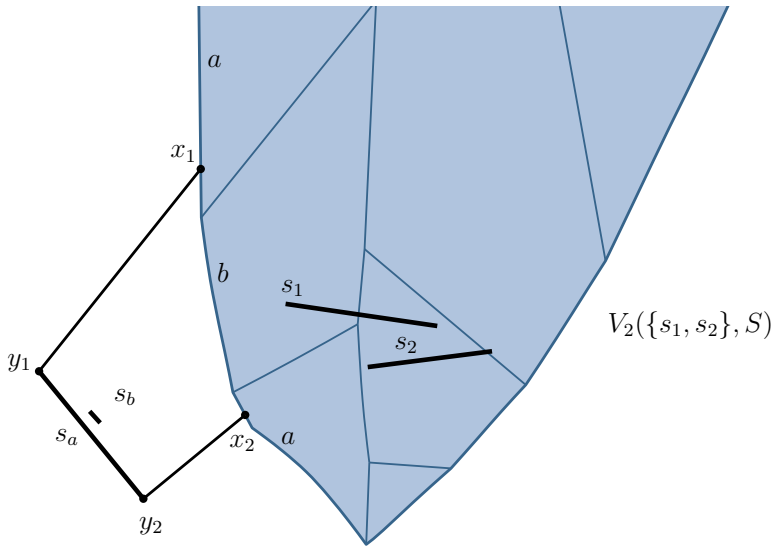

Figure 3.9. A face F of an order-2 Voronoi region $V_2(\{s_1,s_2\},S)$ and the partitioning $\mathcal{V}_1(F)$ of the face F, where $S = \{s_1,s_2,s_a,\ldots,s_d\}$.

s_a and the boundary ∂F. Moreover, there is an appearance of the line segment s_b along the boundary of ∂F between the points x_1 and x_2. Therefore, the line segment s_b is enclosed within the region bounded by the segments s_a, x_1y_1, x_2y_2 and the boundary ∂F.

For the sake of contradiction suppose the line segment s_b has an appearance on ∂F to the right of the point x_2. Let x_3 be a point that belongs to the appearance. Consider a point $y_3 \in s_b$ closest to x_3. Since line segment s_b is inside the region and the point x_3 is outside the region, the line segment x_3y_3 must intersect either s_a or x_2y_2. Suppose that x_3y_3 intersects x_2y_2 (the case when it intersects s_a is similar).

Consider a disk $D(x,s)$ of minimum radius centered at a point x that intersects a line segment s. For any $z \in x_3y_3$ $D(z,s_b) \subseteq D(x_3,s_b)$. Since x_3 is closer to s_b than to s_a the disk $D(x_3,s_b)$ does not intersect s_a; therefore $D(z,s_b)$ does not intersect s_a, $z \in x_3y_3$. Similarly, for any $z' \in x_2y_2$ $D(z',s_a) \subset D(x_2,s_a)$, and $D(z',s_a)$ does not intersect s_b, $z' \in x_2y_2$. Let z'' be an intersection point of x_2y_2 and x_3y_3. Then there a disk $D(z'',s_a)$ which intersects a line segment s_a but does not intersect s_b, and a disk $D(z'',s_b)$ which intersects a line segment s_b but does not intersect s_a. First means that s_a is closer to z'' than s_b and second means that s_b is closer to z'' than s_a, a contradiction. Therefore the appearances of the line segments in S_F along ∂F form an order-2 DSS 3.9.

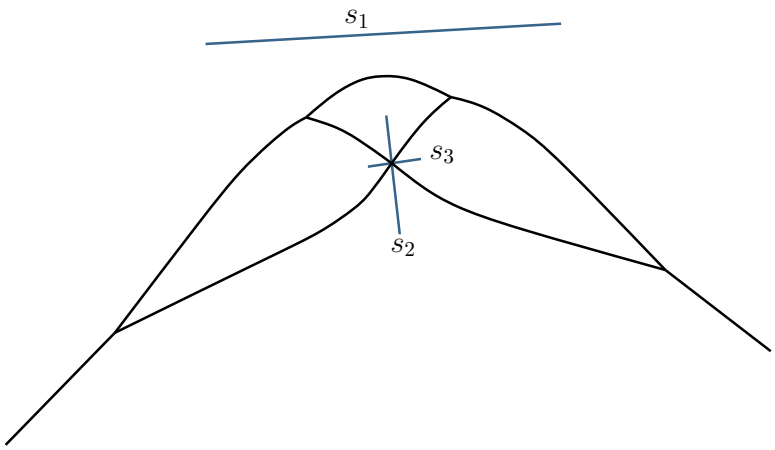

Figure 3.10. The order-1 Voronoi diagram of intersecting line segments s_1, s_2 and s_3. The boundary of the face of the order-1 Voronoi region $V_1(\{s_1\}, \{s_1, s_2, s_3\})$ has appearances of line segments s_2, s_3 which form an order-4 Davenport-Schinzel sequence.

We want to prove that the appearances of intersecting line segments along ∂F form an order-4 DSS. Consider a pair of line segments s_2 and s_3 that intersect at point a. The bisector of s_2 and s_3 has four unbounded branches. We want to prove that only two following cases are possible: (1) ∂F intersects each of the branches at most once; (2) ∂F intersects one of the branches twice and does not intersect other branches. The first case immediately implies that the appearances of s_2 and s_3 form an order-4 DSS, because of the following observation: Since each branch intersects ∂F at most once, there are at most five alternations of the appearances of line segments s_2 and s_3, which implies that the sequence is of order-4, see Figure 3.10. The second case also implies that the apperances of s_2 and s_3 form an order-4 DSS, because there can be at most three alternations of the appearances of line segments s_2 and s_3.

Now let us prove that only these two cases can happen (the proof follows Figure 3.11). Suppose the first case does not happen, which means that ∂F intersects some branch at least once. Then we want to prove that ∂F does not intersect other branches. Consider two intersection points of the branch with ∂F, and consider two disks centered at the points and are touching line segments s_2 and s_3. Let D_1 be the greater disk and D_2 be the smaller disk. D_1 touches a line segment s_1 which induces the face F. D_2 intersects or touches the line segment s_1, because D_2 is centered on the boundary of the face F induced by the line segment s_1. Disk D_1 touches the line segments s_2 and s_3 at two points

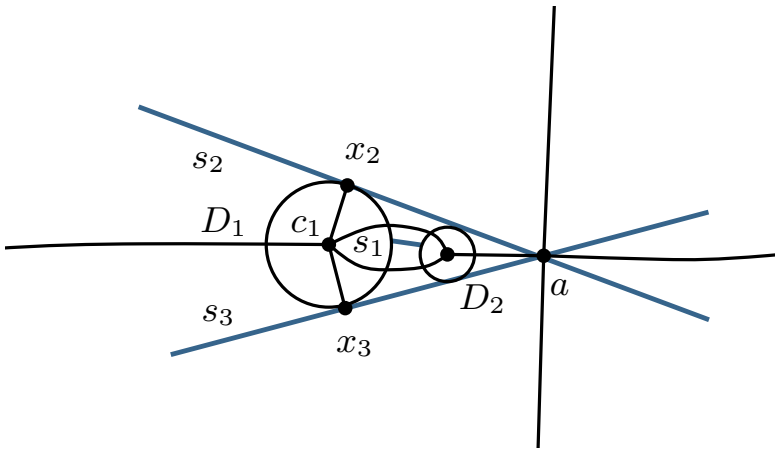

Figure 3.11. The proof of Lemma 3.5.1.

x_2 and x_3. Points x_2 and x_3 split the boundary of D_1 into two arcs. Since D_2 intersects or touches s_1, s_1 touches the arc which is closer to D_2. Let c_1 be the center of the disk D_1. We want to prove that the face F is enclosed in polygon $c_1x_2ax_3$. There is obviously a part of F inside the polygon $c_1x_2ax_3$ and F does not intersect s_2 and s_3. Consider a disk of minimum radius that intersects at least k line segments that is centered at a point on line segments c_1x_2 or c_1x_3. If we move the center of the disk slightly outside the polygon it stops intersecting s_1. Therefore there are no points of the face F outside $c_1x_2ax_3$. Therefore the face F is enclosed is the polygon $c_1x_2a_x3$. Therefore the boundary ∂F does not intersect any other branches of the bisector between line segments s_2 and s_3. Therefore only the two cases discussed above can happen, which implies that the appearances of s_2 and s_3 form an order-4 DSS. □

A single line segment may appear $\Theta(|S_F|)$ times as illustrated in Figure 3.12. Nevertheless, $\mathcal{V}_1(F)$ always remains a tree structure as shown in the following lemma. In fact, using the visibility property of Lemma 3.5.2, it is not hard to see that the sequence of segment appearances along ∂F form a DSS of order-2, if segments do not intersect.

Lemma 3.5.2. *The graph structure of $\mathcal{V}_1(F)$ is a tree*[1]*. Any face P of $\mathcal{V}_1(F)$ has the following visibility property: For every point x in P, there exists a point a_x on ∂F such that the open segment xa_x lies entirely in P, where a_x is the first intersection*

[1]In case of an unbounded face F, we assume an artificial vertex at infinity incident to all unbounded edges

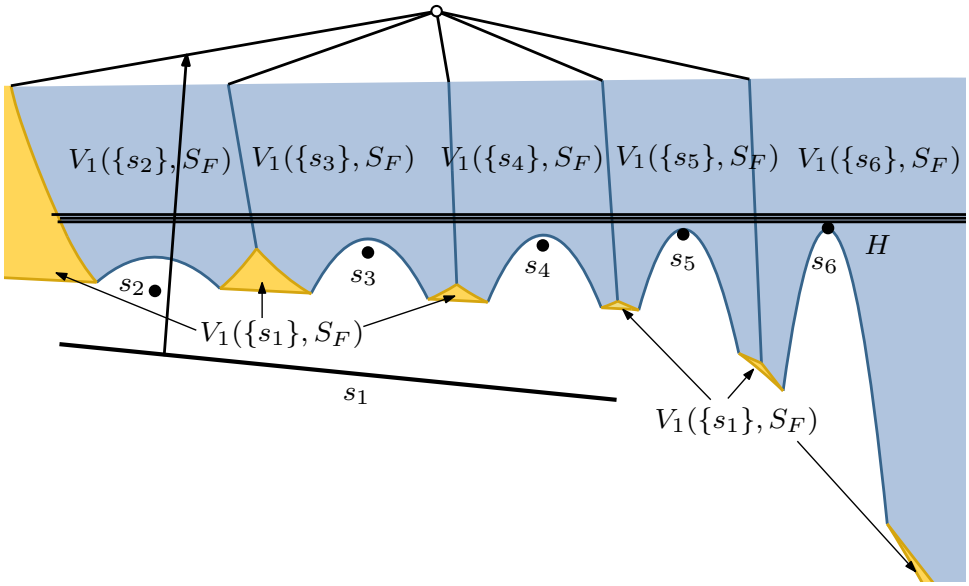

Figure 3.12. A face of an order-i Voronoi region induced by set H of line segments, for $i = 3$. The segment s_1 contributes linear number of subfaces.

of ∂F and the ray $r(s, x)$ emanating from s, which realizes $d(s, x)$, where s is the line segment that induces the face P in $\mathcal{V}_1(F)$ (see Figure 3.12).

Proof. Let $D_{i+1}(x)$ be the order-$(i+1)$ disk centered at point x in P. $D_{i+1}(x)$ touches segment s and intersects all segments in H. Let y be an arbitrary point along segment xa_x. Since $y \in F$, disk $D_{i+1}(y)$ must intersect all line segments in H. Furthermore, since y is closer to s than x and $D_{i+1}(x)$ touches s, $D_{i+1}(y)$ must also touch s. Thus, $y \in P$. Since y is taken arbitrarily, the segment xa_x must lie entirely in P.

Since every face of $\mathcal{V}_1(F)$ must touch ∂F, the graph structure T of $\mathcal{V}_1(F)$ must be a tree or a forest. To prove that T is a tree it is enough to show that every occurrence of a segment $s \in S_F$ along ∂F corresponds to a distinct face of $\mathcal{V}_1(F)$. To this aim, consider a point y on ∂F between two consecutive occurrences of segment s on ∂F. Ray $r(s, y)$ cannot intersect any face P of s because for any point x along the portion of $r(s, y)$ in P segment xa_x is not entirely contained in P. Thus, if x was in a face of s the above visibility property would not hold for x, see Figure 3.13. Thus, the two distinct occurrences of s along ∂F must correspond to distinct faces of s at opposite sides of $r(s, y)$. Therefore, T must be a tree. □

$\mathcal{V}_1(F)$ can be computed in $O(|S_F| \log |S_F| + |\partial F|)$ time by computing $\mathcal{V}_1(S_F)$

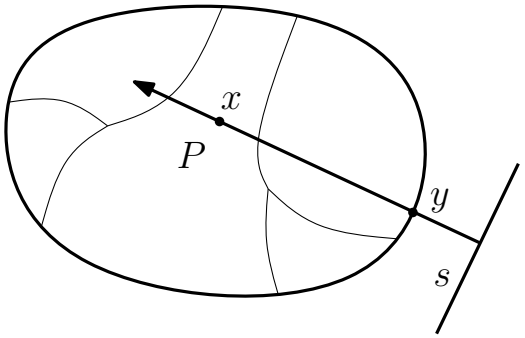

Figure 3.13. Proof of Lemma 3.5.2. There is a point $x \in r(s, y)$ that belongs to P.

independently and truncating it within the interior of F. This results in the standard $O(k^2 n \log n)$-time iterative construction (assuming non-crossing segments). The space complexity corresponds to the size of the largest diagram among all order-i Voronoi diagrams for $i \leq k$. Thus the space complexity is $O(kn)$. It was recently shown that $\mathcal{V}_1(F)$ can be computed directly in linear time, linear in the complexity of ∂F [51], thus, $\mathcal{V}_k(S)$ can be computed in $O(k^2 n + n \log n)$ time. This complexity bound has been known for points [5], however, the repetition of site appearances along ∂F, as shown in [51], makes the adaptation of this result to line segments far from trivial.

3.6 Summary

In this chapter we have investigated the higher-order Voronoi diagram of line segments. The case of line segments had not been studied before; however, there is an obvious need for such investigation, see Section 1.1.1.

We have investigated the phenomenon of disconnected Voronoi regions and proved that a single order-k Voronoi region may disconnect into $\Omega(n)$ faces, for $k > 1$. We have also proven that the union of all faces induced by the same line segment is a connected set with the "weakly star-shaped" property.

We have investigated the structural properties of the higher-order Voronoi diagram of line segments. We have proven the visibility property in the farthest Voronoi region. Using the point-line duality transformation and the results on the complexity of $\leq k$-level (see Section 2.4), we have bounded the number of unbounded faces. Combining these two results we have derived the structural complexity bound for disjoint line segments, $O(k(n-k))$.

We have investigated the case of intersecting line segments and proved the

following structural complexity bounds: $O(k(n-k)+I)$ for $k < n/2$ and $O(k(n-k))$ for $k \geq n/2$. The result is very interesting since it shows that the intersections influence only the low-order Voronoi diagrams and the influence grows weaker with increasing order.

We have extended the results to general L_p metric, for $1 \leq p \leq \infty$. For L_1 and L_∞ we have derived a tighter structural complexity bound: $O((n-k)^2)$, for $k \geq n/2$.

We have also investigated the extension of the standard iterative construction technique to the case of line segments. The iterative technique gives a very simple way to construct the order-k Voronoi diagram of line segments. The running time of the algorithm is $O(k^2 n \log n)$ and the space complexity is $O(kn)$. The algorithm is very useful in the case when all order-i Voronoi diagrams for $i \leq k$ are required. We have also discussed possible improvements of the algorithm.

Chapter 4

Higher-Order Voronoi Diagrams of a Planar Straight-Line Graph

Line segments provide much more flexibility than points. However, in order to make them useful in real-life applications they should be allowed to form complicated solid structures. One of the most fundamental structures that can be formed with line segments is *a planar straight-line graph* (or PSLG for short). A PSLG is a graph in which the edges do not intersect and each edge is a line segment. This is important for applications involving polygonal objects in the plane, for an example see Section 1.1.1. Also, the PSLG can be used to approximate curved structures to a certain extent. In this chapter we investigate the higher-order Voronoi diagram of line segments forming PSLG.

Unlike disjoint or intersecting line segments, line segments forming a PSLG introduce inconsistency in the definition of the order-k Voronoi region. Consider a pair of line segments that share a common endpoint, see Figure 4.1(a). There is a region of the plane which is equidistant from both line segments and thus it cannot "favour" any of them, see also Figure 4.2. In terms of the bisectors it means that a bisector is not a curve, but a two-dimensional object. Since the edges of the higher-order Voronoi diagrams are composed of bisectors, we do not want the bisectors to be two-dimensional, because this will alter the nature of an edge. Furthermore, bisectors in PSLG may intersect non-transversely, see Figure 4.1(b).

There is a large variety of perturbation methods dealing with the degeneracies, but not all of them are applicable in this case. For $k = 1$, a standard convention to cope with the high-degree vertices of a PSLG, is to consider elementary sites as distinct entities, see e.g., [52]. A segment consists of three elementary sites: two endpoints and an open line segment. For $k > 1$, this stan-

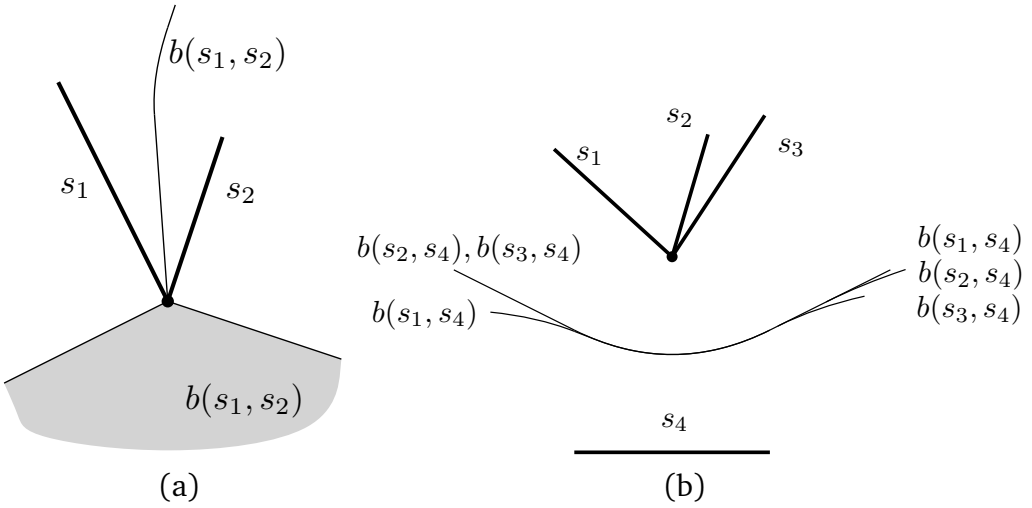

Figure 4.1. (a) A bisector containing a 2-dimensional portion; (b) Bisectors intersecting non-transversely.

dard convention is not sufficient because the issue of equidistant regions from multiple elementary sites remains, and it is independent of k. Moreover, the standard convention, alters the definition of the problem under consideration. For example, for $k = n - 1$, the farthest Voronoi diagram of the elementary sites is the farthest-point Voronoi diagram of the segment endpoints, and not the farthest line-segment Voronoi diagram as defined in [9]. Similarly, this issue is not addressed by other standard techniques, which deal with two-dimensional bisectors, such as assigning a priority to sites while offering an entire equidistant region to the segment of higher priority (e.g. [54]), or using an angular bisector to split equidistant regions [9]. Perturbation techniques (see e.g., [79]) to transform the PSLG into a set of disjoint line segments, may create artificial faces and tedious decompositions that are unrelated to the problem under consideration, see e.g., Figure 4.3. Moreover, these techniques eliminate the entire presence of endpoints with a high degree of incidence. In applications such endpoints represent some instances and thus we want to preserve all information related to them.

Instead of altering the structure of a planar straight-line graph, we extend the definition of the Voronoi regions, see Section 4.1. The extended definition is compatible with the ordinary definition meaning that in the case of disjoint line segments, they are equivalent. This achieves simplicity in the resulting order-k decomposition, avoiding the tedious regions that would be created if we perturbed the PSLG into a set of disjoint line segments. It also reveals the exact

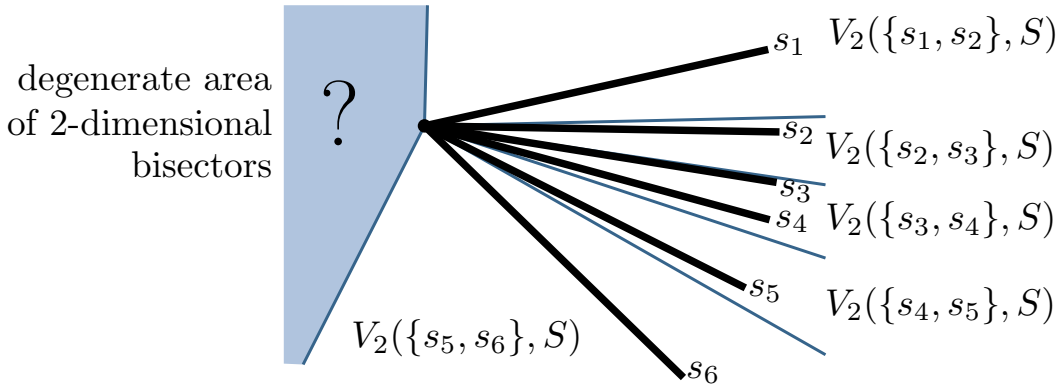

Figure 4.2. The degenerate area created by the endpoint of PSLG in case we use the ordinary definition of the order-k Voronoi diagram, where $k = 2$ and $S = \{s_1, \ldots, s_6\}$.

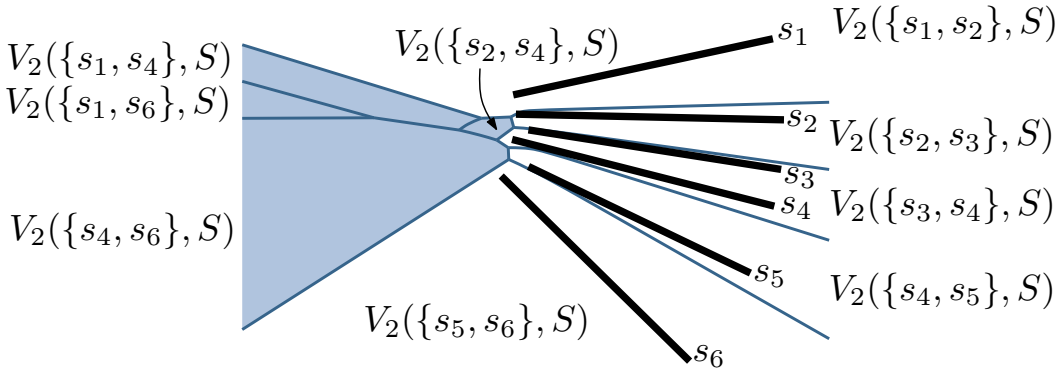

Figure 4.3. The order-k Voronoi diagram of untangled line segments, where $k = 2$ and $S = \{s_1, \ldots, s_6\}$. Artificial faces are shown shaded.

elementary site, which actually defines the order-k distance for every point in the plane, similarly to the standard convention for $k = 1$ of considering distinct elementary sites.

In Section 4.2 we show that the extended definition preserves the relation between Voronoi diagrams and the arrangements of distance functions (see Section 2.1). In Section 4.3 we prove that the asymptotic structural complexity does not increase in the case of a planar straigh-line graph. In Section 4.4 we extend the iterative algorithm and study its properties in the case of line segments.

4.1 Augmenting the Definition of a Voronoi Region

In this chapter we make a *weak general position assumption* that no more than three *elementary sites* are tangent or touch the same circle. In case the line through a segment s is tangent to a circle C at one of the segment endpoints, only the endpoint is considered to touch the disk C.

Definition 2. *Let $D_k(x)$ be the disk of minimum radius, centered at point x, which intersects (or touches) at least k line segments. $D_k(x)$ is called an* order-k disk. *If $D_k(x)$ touches exactly one elementary site p then it is called a proper order-k disk and it is denoted as $D_k^p(x)$. The set of line segments in S that have a non-empty intersection with an order-k disk $D_k(x)$ is denoted as $S_k(x)$.*

For every point x in the plane, $D_k(x)$, and thus, $S_k(x)$, are unique. If $D_k(x)$ is proper then x must be a point in the interior of a Voronoi region. Otherwise, x must be a point along the bisector of two elementary sites.

For segments forming a PSLG, we extend the notion of a subset of S of cardinality k to an *order-k* subset, which may have cardinality greater than k.

Definition 3. *A set $H \subseteq S$ is called an order-k subset if*

1. *$|H| = k$ (Type-1); or*
2. *$|H| > k$ (Type-2), and there exists a proper order-k disk $D_k^p(x)$, such that $S_k(x) = H$ and p is an endpoint common to at least two segments in H. Point p is called a* representative *of H. An order-k subset of representative p is denoted as H_p. The set of segments incident to p is denoted as $I(p)$.*

Remark. A set of segments H may have two (or more) representatives p, q, resulting in two distinct order-k subsets H_p and H_q, where each has a distinct region in $\mathcal{V}_k(S)$.

An order-k Voronoi region can now be defined in terms of order-k subsets of S instead of cardinality-k subsets. For a Type-1 subset H, its order-k Voronoi region $V_k(H,S)$ is defined in the ordinary way Eq. (3.1) and it is referred to as Type-1. This is equivalent to $V_k(H,S) = \{x \mid S_k(x) = H\}$. For a Type-2 order-$k$ subset H with representative p, its order-k Voronoi region $V_k(H_p,S)$ is referred to as Type-2 and it is defined as follows.

$$V_k(H_p,S) = \{x \mid S_k(x) = H_p \wedge D_k(x) = D_k^p(x)\}^o, \tag{4.1}$$

where X^o denotes the interior of a set X.

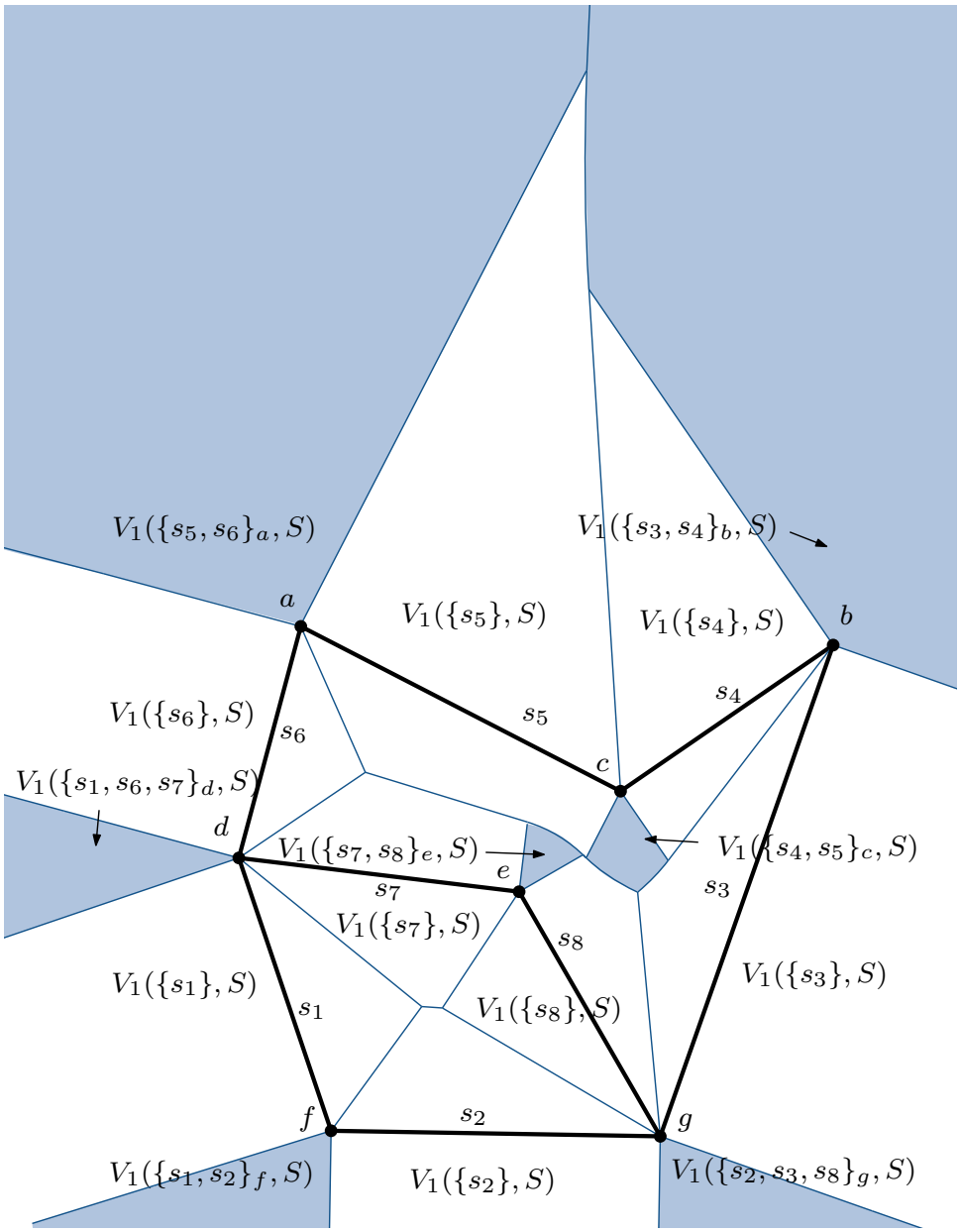

Figure 4.4. The order-1 Voronoi diagram of PSLG, where $S = \{s_1, \ldots, s_8\}$ are the line segments and $a, \ldots, g$ are the endpoints. The Type-2 Voronoi regions are shown shaded.

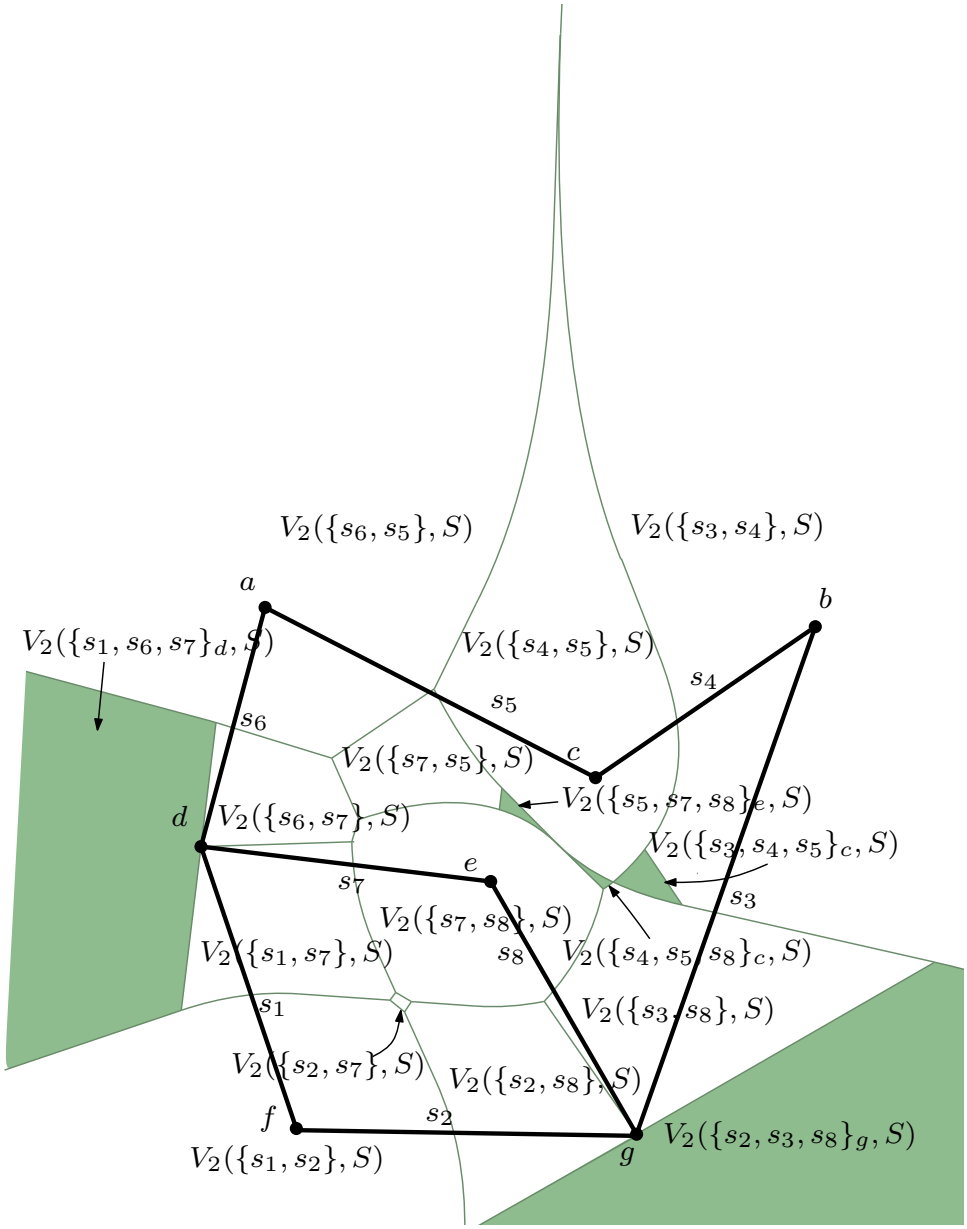

Figure 4.5. The order-2 Voronoi diagram of PSLG, where $S = \{s_1, \ldots, s_8\}$ are the line segments and $a, \ldots, g$ are the endpoints. The Type-2 Voronoi regions are shown shaded.

The order-k Voronoi diagram remains the partitioning of the plane into order-k Voronoi regions and their boundaries, which reveal the graph structure of the diagram consisting of Voronoi edges and vertices. Figures 4.4 and 4.5 illustrate the 1st and 2nd order Voronoi diagrams of a PSLG. Type-2 Voronoi regions are illustrated shaded. The following lemma gives the main property of a Type-2 Voronoi region.

Lemma 4.1.1. *Let $V_k(H_p, S)$ be a Type-2 order-k Voronoi region. For any point x in $V_k(H_p, S)$, and for any segments $s \in H_p$ and $t \in S \setminus H_p$, $d(x,s) \leq d(x,p) < d(x,t)$. Furthermore, $S_k(x) = S_{k+1}(x)$. $V_k(H_p, S)$ contains no graph elements of $\mathcal{V}_{k-1}(S)$ nor of $\mathcal{V}_{k+1}(S)$.*

Proof. The first claim directly derives by the definitions of $V_k(H_p, S)$ and $D_k^p(x)$ for any point x in $V_k(H_p, S)$. Since this is a Type-2 region, $|S_k(x)| > k$, thus, $S_k(x)$ must equal $S_{k+1}(x)$. The last claim derives by the fact that $D_k(x)$ is a proper order-k disk, i.e., $D_k(x) = D_k^p(x)$, for any point x in $V_k(H_p, S)$, by its definition, while a graph element of $\mathcal{V}_{k-1}(S)$ and $\mathcal{V}_{k+1}(S)$ must always correspond to a non-proper order-k disk. □

By Lemma 4.1.1, each Type-2 Voronoi region is assigned to exactly one endpoint of the PSLG and $d(x, H_p) = d(x, p)$ for every point x in $V_k(H_p, S)$. Consequently there is no farthest subdivision inside a Type-2 Voronoi region. For the same reason, a Type-2 order-k Voronoi region can only enlarge in the order-$(k+1)$ diagram, spreading its influence into neighboring Type-1 regions. At order $k = |H|$, $V_k(H_p, S)$ becomes Type-1. Figures 4.6 and 4.7 illustrate the evolution of a Type-2 region as the order of the diagram increases. Non-shaded and shaded regions are Type-1 and Type-2 respectively. Details of the figures are discussed below.

We now consider the properties of Voronoi edges and vertices in the presence of Type-2 regions. A Voronoi edge bounding a Type-2 region is an ordinary bisector between the region representative and one element of the neighboring k-subset. A Voronoi vertex v is the intersection point of three ordinary bisectors, under the weak general position assumption. Thus it can have a degree between 3 and 6. The following lemma summarizes.

Lemma 4.1.2. *Consider a Type-2 Voronoi region $V_k(H_p, S)$, a neighboring Voronoi face $V_k(J, S)$, and their incident boundary e.*

- *If $V_k(J, S)$ is Type-1, then $J \cup I(p) = \{y\} \cup H_p$, and e belongs to bisector $b(p, y)$.*

- *If $V_k(J,S)$ is Type-2, with representative $q \neq p$ ($J = J_q$), then e belongs to $b(p,q)$ and $H_p \setminus I(p) = J_q \setminus I(q)$.*
- *If $V_k(J,S)$ is Type-2, with the same representative p ($J = J_p$), then e belongs to $b(p,y)$, where $\{y\} = H_p \triangle J_p$.*

Proof. Let e be incident to a Type-1 region $V_k(J,S)$. Consider an order-k disk centered at the point x on the edge e. If we move slightly the disk inside $V_k(H_p,S)$ it will touch the endpoint p and it will intersect line segments in H_p. If we move slightly the disk inside $V_k(J,S)$ it will touch some line segment $y \in J$ and it will intersect line segments in J. Thus when the disk is centered on the edge it touches the line segment y and the endpoint p, i.e. $x \in b(p,y)$.

Let e be incident to a Type-2 region $V_k(J_q,S)$. Suppose $p \neq q$, then similarly to the previous case, the order-k disk that is centered on the edge e, should touch two endpoints p and q, i.e. $x \in b(p,q)$. Suppose $p = q$, then consider the order-k disk that is centered on the edge e. If we center the disk at the point $x' \in V_k(J_p,S)$ it will touch the endpoint p and will intersect the line segments in J. Similarly, if center the disk at the point $x'' \in V_k(H_p,S)$ it will touch the endpoint p and will intersect the line segments in H. Since $V_k(H_p,S)$ and $V_k(J_p,S)$ are not equal the interiors of the order-k disks $D_k(x')$ and $D_k(x'')$ intersect different sets of line segments. Therefore as we continuously move the order-k from x' to x'' some line segment y escapes the interior of the disk. Therefore when the disk is centered at the point x it touches the line segment y and the endpoint p, i.e. $x \in b(p,y)$. $y \in H_p \triangle J_p$, where $\triangle$ denotes the symmetric difference. □

In the ordinary case of Type-1 Voronoi regions, the degree of a Voronoi vertex is always 3, or 6 for an old vertex if we consider the augmented diagram that includes the farthest subdivision within each region (i.e., $\mathcal{V}_k(S)$ and $\mathcal{V}_{k-1}(S)$ superimposed). However, Voronoi vertices incident to Type-2 Voronoi regions can have degree any number between 3 and 6. Examples are illustrated in Figures 4.6 and 4.7.

Lemma 4.1.3. *The order-k Voronoi diagram of a PSLG is a tessellation, i.e. each point on the plane belongs either to the interior of a single order-k Voronoi region or to the boundary of an order-k Voronoi region.*

Proof. Let x be an arbitrary point on the plane. Consider the order-k disk centered at x. If $S_k(x)$ as exactly k elements then x belongs to the interior of the Type-1 Voronoi region $V_k(S_k(x),S)$.

Otherwise $S_k(x)$ has more than k elements. Since order-k disk is the disk of minimum radius which intersects at least k line segments it means that we

cannot reduce the radius of the disk while intersecting at least k line segments. This can happen in two cases: (a) either the radius is zero; or (b) any disk of smaller radius will intersect less than k line segments. If the disk has zero radius and it intersects at least k line segments then the center of the disk is an endpoint of the PSLG that is incident to at least k line segments. In this case if we move x slightly in any direction it will belong either to the interior of the Voronoi region or to the edge of the Voronoi region as described by the cases below. Therefore in the case (a) x belongs to the boundary of a Voronoi region.

Consider the case (b) in this case the order-k disk is of minimal radius because any disk of smaller radius will intersect less than k line segments. This implies that the disk touches some line segments. Consider the case when the disk touches exactly one elementary site. This should be an endpoint, because otherwise we can reduce the disk and still intersect k line segments. Let this be the endpoint p. Consider the bisectors between the endpoint p and the open portions of the line segments in $I(p)$. If the center x of the disk does not belong to any of these bisectors than x belongs to the interior of the Type-2 Voronoi region $V_k(H_p, S)$, $H_p = S_k(x)$, because we can slightly move the center in any direction and the order-k disk will intersect the same line segments. Suppose x belongs to one of the bisectors $b(p,s)$, then the boundary of the disk touches an endpoint e and the open portion of the segment s is tangent to the disk. Denote as $S_k^o(x) \subset S_k(x)$ the set of line segments that intersect the interior of the disk. If $S_k^o(x)$ has exactly $k-1$ line segments then we can move the center of the disk slightly so the interior intersects the open portion. Then we can shrink the disk so it intersects exactly k line segments: $S_k^o(x)$ and s. In this case x belongs to the edge between the Type-2 region $V_k(H_p, S)$, $H_p = S_k(x)$ and Type-1 region $V_k(S_k^o(x) \cup \{s\}, S)$. Note that this is the reason why we have to explicitly exclude the boundary in Def. 4.1. Otherwise the Type-2 region would include the part of the bisector $b(p,s)$ and the region would not be an open set. If $S_k^o(x)$ has less than $k-1$ line segments then even if we move the center of the disk slightly around and the open portion of s may come inside the interior of the disk, it will not be enough for this disk to shrink and exclude the endpoint p. So in this case the neighborhood of the point x belongs to the interior of $V_k(H_p, S)$, $H_p = S_k(x)$.

Consider the case when the disk touches more than one elementary site. Then clearly if we move the disk slightly around it will touch one elementary site and become one of the cases described above.

Therefore in all the cases the point x belongs either to the interior of the Type-1 or Type-2 Voronoi region or the boundary. □

Figure 4.6(a) depicts a vertex v incident to a Type-2 region $V_1(H_p, S)$ and two

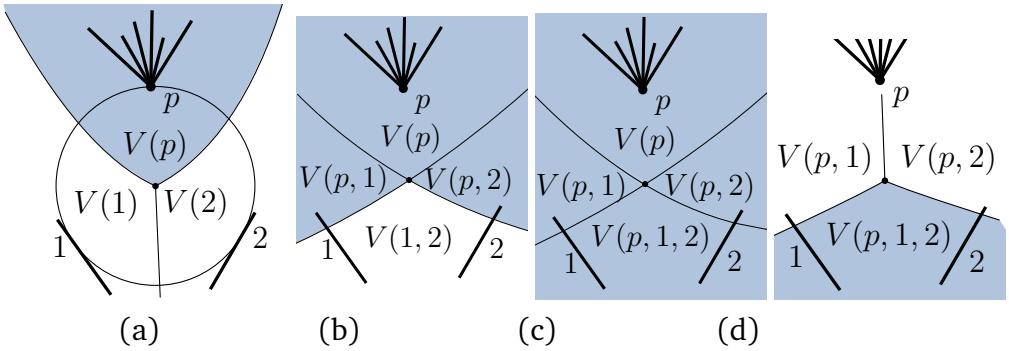

Figure 4.6. A Type-2 Voronoi region of representative p, denoted as $V(p)$, and an incident Voronoi vertex for various orders k. (a) $k = 1$; (b) $k = 2$; (c) $2 < k \leq |I(p)|$ (for $k = |I(p)|$, $V(p)$ is Type-1); (d) $k = |I(p)| + 1$. ($V(p, e_1, \ldots, e_m)$ stands for $V_k(p, H, S)$, where $\{e_1, \ldots, e_m\} = H \setminus I(p)$.)

Type-1 regions. Figure 4.6(b) shows how p spreads into its neighboring regions and transforms them into Type-2 in $\mathcal{V}_2(S)$. Figure 4.6(c) shows the diagram for several orders k, $3 \leq k \leq |I(p)|$. At $k = |I(p)|$, $V_k(H_p, S)$ becomes Type-1. Figure 4.6(d) illustrates the diagram for $k = |I(p)| + 1$. Under the weak general position assumption, a vertex v incident to a Type-2 region can have degree between 3 and 6, as a result of being the intersection point of three bisectors intersecting at v. Figure 4.7 illustrates an example of a vertex initially incident to three Type-2 Voronoi regions with representatives p, r, and q, respectively as shown in Figure 4.7(a). As the order increases, the Voronoi region of q (q has the smallest degree) becomes Type-1; in the next order it is split between two Type-2 regions of representatives r and p respectively, as shown in Figure 4.7(b). In Figure 4.7(c), after the region of r becomes Type-1 for $k = |I(r)|$, it is split by the representatives of the neighboring Type-2 regions at order $k = |I(r)| + 1$. This creates a Voronoi vertex of degree 5 incident to portions of three bisectors. Later, the Voronoi region $V(p)$ will be split by its two neighbors and the incident Voronoi vertex will obtain degree 6.

Type-2 Voronoi regions illustrate the peculiarities listed above, however, they pose no difficulty in the construction of the diagram. In fact, their presence simplifies the diagram as compared with one that could be obtained by perturbing the PSLG into a set of disjoint segments. The complexity of the diagram remains $O(k(n-k))$ as shown in the following subsection.

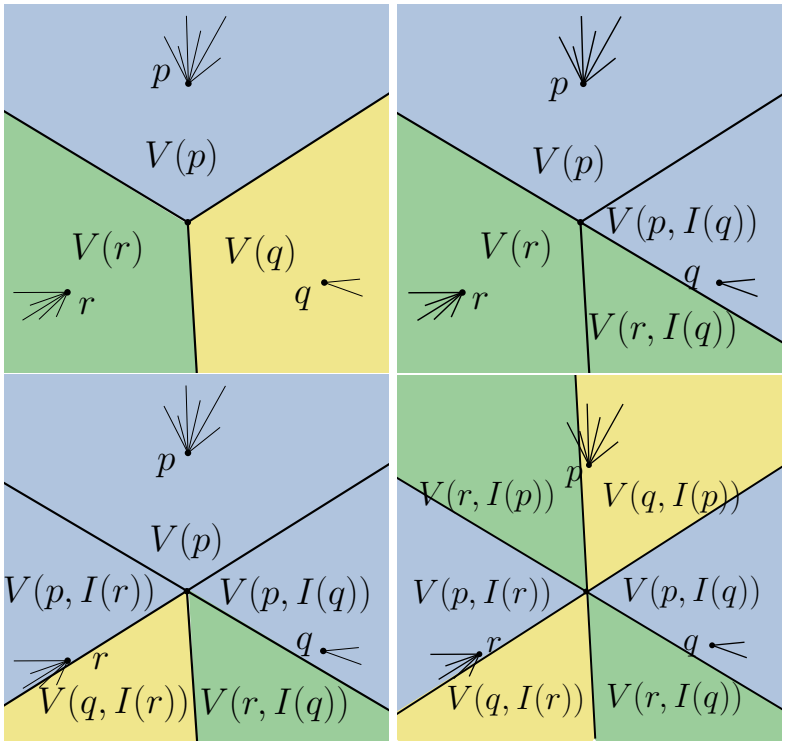

Figure 4.7. Top left: A Voronoi vertex in $\mathcal{V}_1(S)$ incident to three Type-2 regions; Top right: In $\mathcal{V}_{j+1}(S)$, $j = |I(q)|$, region $V(q)$ is split by the representatives of the neighboring Type-2 regions; Bottom left: In $\mathcal{V}_{k+1}(S)$, $k = |I(r)|$, region $V(r)$ is split by the representatives of the neighboring Type-2 regions; Bottom right: In $\mathcal{V}_{s+1}(S)$, $s = |I(p)|$, region $V(p)$ is split by the representatives of the neighboring Type-2 regions.

4.2 Relation with Arrangements

In case of line segments forming a PSLG the definition of the Type-2 order-k Voronoi region is given through the notion of the order-k disk. This is different from the way the order-k Voronoi region is defined for point sites, see Eq. (1.3), disjoint line segments, see Eq. (3.1) or Type-1 Voronoi regions in case of a PSLG. In these cases the definition uses the notion of distance functions, which naturally implies relation between Voronoi diagrams and the arrangements of distance functions, see Section 2.1. Since relation with arrangements of distance functions is a favorite property of Voronoi diagrams, we want to preserve it for the case of line segments forming a PSLG.

In this section we show that the definition of the Type-2 order-k Voronoi region preserves the relation with arrangements of distance functions. Following Section 2.1, let $f_s(x) = d(s,x)$ be the distance function between points on a plane and the line segment s. We consider the arrangement of the surfaces in $\mathbb{R}^3$ that correspond to the distance functions. Consider the $(k-1)$-level and take a point x on the xy-plane. Let x^{k-1} be the point x projected vertically on the $(k-1)$-level. The definition of the i-level implies that $\pi^-(x^{k-1}) \leq k-1$ and $\pi^-(x^{k-1}) + \pi^0(x^{k-1}) > k-1$. Since the surfaces correspond to the distance functions the distance between the point x and the projection x^{k-1}, $d(x,x^{k-1})$ is equal to to the distance between point x and the line segments that correspond to the surfaces in $\Pi^0(x^{k-1})$, i.e. $d(x,s) = d(x,x^{k-1})$ for every $f_s \in \Pi^0(x^{k-1})$. Similarly, the surfaces strictly below point x^{k-1}, $\Pi^-(x^{k-1})$ correspond to the sites that are closer to x than point x^{k-1}, i.e. $d(x,s) < d(x,x^{k-1})$ for every $f_s \in \Pi^-(x^{k-1})$. Therefore the disk on the xy-plane with the center at point x and the radius equal to $d(x,x^{k-1})$ will intersect $> k-1$ line segments and its interior will intersect $\leq k-1$ line segments, i.e. this will be an order-k disk. Therefore by slightly abusing the notation and considering the point x on the xy-plane in $\mathbb{R}^3$ is as if it is a point in $\mathbb{R}^2$ we derive the following equation:

$$S_k(x) = \left\{ s \in S \mid f_s \in \Pi^- \left(x^{k-1}\right) \cup \Pi^0 \left(x^{k-1}\right) \right\}$$

Consider now two points x^{k-1} and y^{k-1} on the $(k-1)$-level. We say that they are equivalent if $\Pi^-(x^{k-1}) \cup \Pi^0(x^{k-1}) = \Pi^-(y^{k-1}) \cup \Pi^0(y^{k-1})$ or, equivalently $S_k(x) = S_k(y)$. The equivalence relation induces the subdivision of the $(k-1)$-level into sets with equal equivalence classes. If we project this subdivision onto the xy-plane we receive the subdivision of the xy-plane which will be a coarsing of the order-k Voronoi diagram $\mathcal{V}_k(S)$. More precisely, this coarsing is agnostic with respect to the elementary site that defines the radius of the order-k disk. This may result in the situation when two neighboring Type-2 order-k Voronoi

regions that have the same set S_k but have different representative are merged in the coarsing. Since the representative of the Type-2 Voronoi region may correspond to some real-life instance that carries important information (see [69]) we may want to distinguish these Type-2 Voronoi regions. Therefore the definition of the Type-2 Voronoi region includes the notion of the representative.

Therefore there is a relation between the order-k Voronoi diagram of a PSLG and the $(k-1)$-level of distance functions, the projection of the latter is the coarsing of the former.

4.3 Structural Properties and Complexity

Let $S(\varepsilon)$ be a set of disjoint line segments as obtained from S by a small perturbation $\varepsilon > 0$ of the incident segment endpoints, see Figure 4.8(a). In particular, for every endpoint p, with $|I(p)| > 1$, and for every line segment $s \in I(p)$, move the endpoint of s incident to p along the line through s by a small amount $\delta_s < \varepsilon$, remaining within $n(p,\varepsilon)$, $n(p,\varepsilon) = \{x \mid d(x,p) < \varepsilon\}$. By using variable amounts for δ_s, and given the weak general-position assumption, the general-position assumption can be maintained. Despite many artificial faces (see Figure 4.3), The structural complexity of $\mathcal{V}_k(S(\varepsilon))$ is $O(k(n-k))$, however, many artificial faces are created (see Figure 4.3).

In the following we show that the number of faces in $\mathcal{V}_k(S)$ cannot exceed those of $\mathcal{V}_k(S(\varepsilon))$ for certain ε, and thus, the complexity of $\mathcal{V}_k(S)$ is also $O(k(n-k))$. To this aim, we use the refined versions of $\mathcal{V}_k(S)$ and $\mathcal{V}_k(S(\varepsilon))$, where all regions are subdivided into the finest sub-faces by superimposing their respective order-$(k-1)$ diagrams. The faces of $\mathcal{V}_k(S(\varepsilon))$ are further subdivided by their elementary sites, such that for every point x in a fine face, $D_k(x) = D_k^p(x)$ for exactly one elementary site p.

Lemma 4.3.1. *There is an injection from the (fine) faces of $\mathcal{V}_k(S)$ to the (fine) faces of $\mathcal{V}_k(S(\varepsilon))$, for some $\varepsilon > 0$.*

Proof. For a given $\varepsilon > 0$, we define the mapping in the following way. For every (fine) face F_j of $\mathcal{V}_k(S)$, consider an arbitrary point a_j in the interior of F_j, and map a_j to the face F'_j in $\mathcal{V}_k(S(\varepsilon))$ where it belongs. This creates a function from the faces of $\mathcal{V}_k(S)$ to the faces of $\mathcal{V}_k(S(\varepsilon))$. Our goal is to find a value of ε for which the mapping is guaranteed to be injective. The difficulty is due to the presence of disconnected regions. Since the perturbation may change the adjacency relations in the diagram, different faces in $\mathcal{V}_k(S)$ that belong to the same Voronoi region may merge into the same face in $\mathcal{V}_k(S(\varepsilon))$.

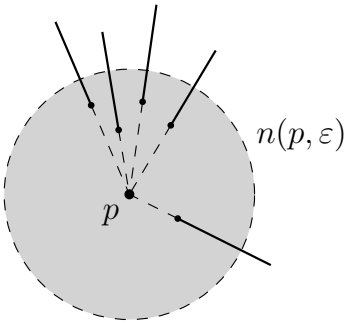

Figure 4.8. Untangling abutting line segments at endpoint p.

Consider $\mathcal{V}_k(S)$. To avoid the merging, we surround each face F_j of $\mathcal{V}_k(S)$ with a closed curve γ_j that passes around F_j without touching it and intersects all the faces adjacent to F_j. For each face F_j we consider the set $\mathcal{D}_j$ of the order-k disks which consists of: (1) the order-k disk with center at a_j; (2) all order-k disks with centers on γ_j. For each order-k disk D in $\mathcal{D}_j$ consider two sets of line segments: line segments that intersect the interior of the disk, but do not touch the disk, and line segments that neither touch nor intersect the disk. Since neither of these sets touch the boundary of D, there is a non-zero value $\delta(D)$ by which we can shrink or expand the disk, until one of the line segments leaves one of the two sets. If we choose $\varepsilon > 0$ such that each disk D shrinks or expands by less than $\delta(D)$, for every D in D_j and every face F_j of $\mathcal{V}_k(S)$, then the mapping becomes injective. The choice of γ_j ensures that the face F'_j of $\mathcal{V}_k(S(\varepsilon))$ assigned to a_j is completely surrounded by faces of Voronoi regions that belong to order-k subsets that are different from the one of F_j. Since we also consider the order-k disk centered at a_j, the face F'_j can not merge with any of the neighbors intersected by the curve γ_j. Thus, no two faces of $\mathcal{V}_k(S)$ can map to the same face of $\mathcal{V}_k(S(\varepsilon))$ for the this choice of ε. □

By Lemma 4.3.1, we conclude.

Theorem 4.3.2. *The structural complexity of the order-k Voronoi diagram of n line segments forming a planar straight-line graph is* $O(k(n-k))$.

Proof. Lemma 4.3.1 implies that there is an injection from fine faces of $\mathcal{V}_k(S)$ to fine faces of $\mathcal{V}_k(S(\varepsilon))$, for some $\varepsilon > 0$. The refined diagram $\mathcal{V}_k(S)$ is obtained by superimposing $\mathcal{V}_k(S)$ and $\mathcal{V}_{k-1}(S)$. The refined diagram $\mathcal{V}_k(S(\varepsilon))$ is obtained by superimposing $\mathcal{V}_k(S(\varepsilon))$ and $\mathcal{V}_{k-1}(S(\varepsilon))$ and further subdividing by elementary sites. Since $S(\varepsilon)$ is the set of disjoint line segments, the structural complexity of the refined $\mathcal{V}_k(S(\varepsilon))$ is $O(k(n-k)) + O((k-1)(n-k+1)) = O(k(n-k))$. The

additional subdivision by elementary sites does not increase the structural complexity since every line segment has three elementary sites.

Therefore the structrual complexity of $\mathcal{V}_k(S)$ is bounded by $O(k(n-k))$. □

4.4 Extending the Iterative Construction

The iterative construction is the universal approach to construct the order-k Voronoi diagram. It can be also applied to the case of line segments forming a PSLG. In Section 3.5 we have discussed the iterative construction and its extension to the case of line segments. In this section we will show how to extend the iterative construction to the case of line segments forming a PSLG.

In the case of a PSLG the Type-2 faces may remain for several orders until they become Type-1 as we have seen in Section 4.1. As a result we should skip the Type-2 faces during the subdivision step. Moreover, some of the Voronoi edges that bound Type-2 faces may also remain for several orders. The iterative algorithm can be described as follows:

- Construct $\mathcal{V}_1(S)$ using any available algorithm with $O(n \log n)$ time complexity.
- For $i = 1, \ldots, k-1$ do:
  - For every Type-1 face F of every region $V_i(H,S)$ of $\mathcal{V}_i(S)$ compute the part of $\mathcal{V}_1(S \setminus H)$ enclosed within F.
  - Remove/Disregard every edge of $V_i(H,S)$ that does not remain in $V_{i+1}(H,S)$.

In Figure 4.9 we show the one iteration of the algorithm. We take a Type-1 face F of $\mathcal{V}_2(S)$ and subdivide it with the nearest neighbor Voronoi diagram. Let $V_2(H,S)$ be the Voronoi region of F, where $H = \{s_7, s_8\}$. We only need to consider the line segments S_F that are not in H and contribute to the boundary of the face F, $S_F = \{s_1, s_2, s_3, s_4, s_5, s_6\}$ (see Section 3.5). In Figure 4.9 the order-1 Voronoi diagram $\mathcal{V}_1(S_F)$ contains one Type-2 Voronoi region $\mathcal{V}_1(\{s_4, s_5\}_c, S_F)$. The corresponding edge of the face F should not be removed since it remains in the order-3 Voronoi diagram.

Despite the minor differences described above, the iterative construction can be easily applied to the case of line segments forming a PSLG.

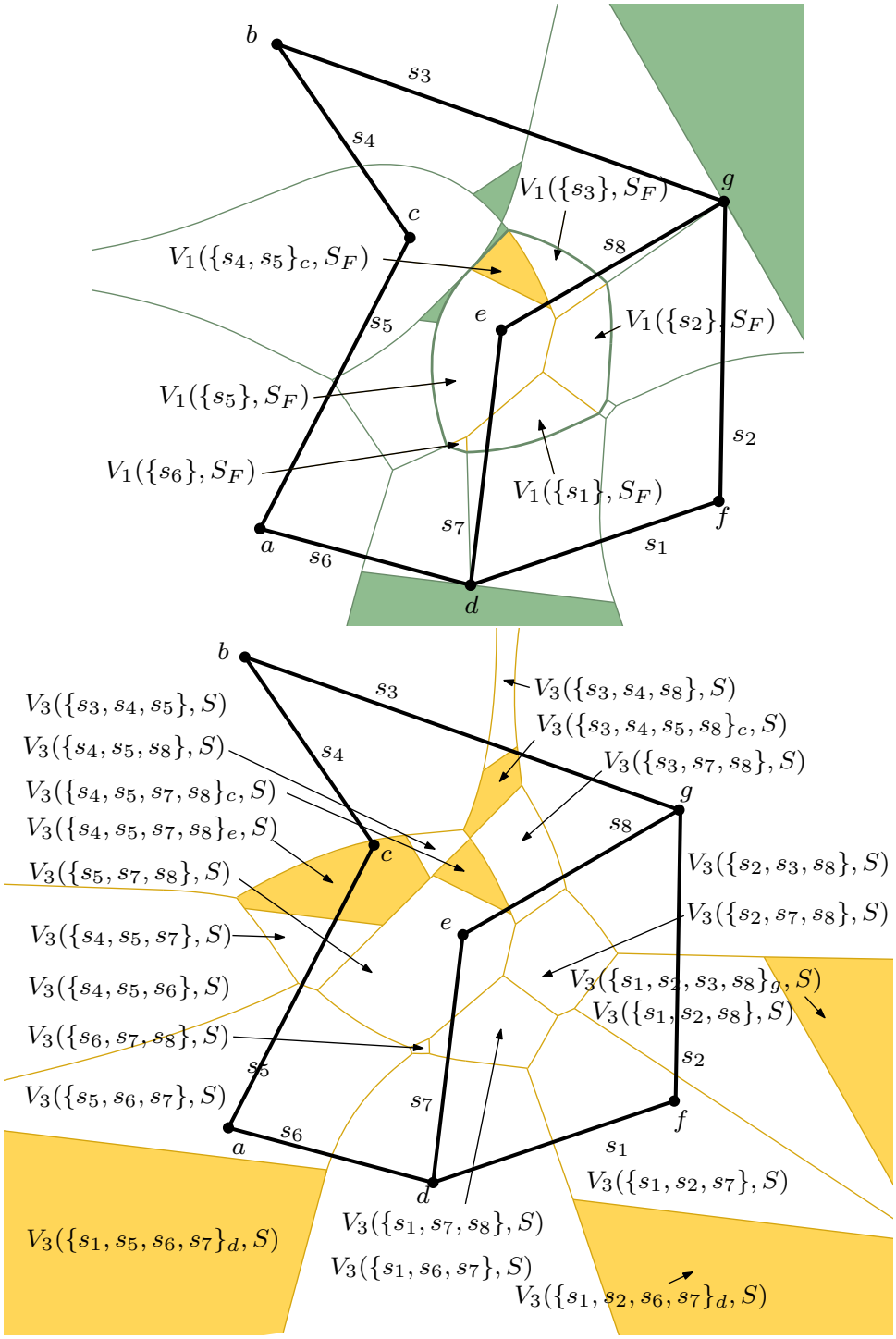

Figure 4.9. Top: The order-2 Voronoi diagram of a PSLG (see Figure 4.5 for more details). The partitioning $\mathcal{V}_1(F)$ of the face $F \subseteq V_2(\{s_7, s_8\}, S)$, where $S_F = \{s_1, \ldots, s_6\}$; Bottom: The order-3 Voronoi diagram of a PSLG, after the procedure is applied for every face F of $\mathcal{V}_2(S)$.

4.5 Summary

In this chapter we have investigated the higher-order Voronoi diagram of line segments forming a planar straight-line graph. Line segments forming a PSLG have very important applications, e.g. see Section 1.1.1.

We have extended the definition of the order-k Voronoi region to the case of line segments forming a PSLG and investigated the properties of the Type-2 Voronoi regions to show the consistency of the extended definition. We have also shown that the relation between the order-k Voronoi diagram and the $(k-1)$-level of distance functions remains under the extended definition. Using the perturbation technique, we have shown that the structural complexity of the order-k Voronoi diagram of a PSLG is $O(k(n-k))$, as in the case of disjoint line segments and points. Moreover, the proof shows that there is an injection from the faces of the order-k Voronoi diagram of a PSLG to the faces of the perturbed order-k Voronoi diagram, which implies that the structural complexity of the former is less than that of the latter.

Finally, we have shown how to extend the standard iterative construction algorithm to the case of line segments forming a PSLG.

Chapter 5

Sweepline Algorithm

In this chapter we discuss the sweepline algorithm for the higher-order Voronoi diagram of line segments. The algorithm sweeps the plane with the horizontal line while constructing the Voronoi diagram from top to bottom. The algorithm constructs all order-i Voronoi diagrams for $i \leq k$ in $O(k^2 n \log n)$ time and $O(kn)$ space. The sweepline algorithm does not need to keep the entire Voronoi diagram and it stores only those parts which are close to the horizontal line. Therefore the algorithm can be used for on-the-fly computations.

The sweepline is a standard technique in computational geometry [13]. It allows us to sequentially construct the structures while moving from top to bottom (the direction is not important). The space above the horizontal line contains the structure already constructed while the space below the horizontal line corresponds to the part yet to be processed. We discretize the movement of the horizontal line and consider the discrete event points that are used to do the computation.

The main difficulty with constructing Voronoi diagrams using the sweepline technique is that the sites may contribute to the structure of the Voronoi diagram even before the sweepline encounters them. Fortune solved this problem by introducing a transformation that allows us to construct the diagram that has the same topology as the nearest neighbor Voronoi diagram [43]. Subsequently, Seidel described a way to avoid the transformation and construct the nearest neighbor Voronoi diagram directly by introducing an additional curve, called a *beach line* [78]. The beach line is the lower envelope of an arrangement of parabolas that continuously change shape as the horizontal line sweeps the plane. The vertices of the beach line move along the edges of the Voronoi diagram, and every time a pair of vertices coincide, a Voronoi vertex is created. By processing the discrete event points that change the topology of the lower

envelope, one can construct the nearest neighbor Voronoi diagram by sweeping the plane with the horizontal line. Interestingly, the beach line can be viewed as an intersection between an arrangement of distance functions in $\mathbb{R}^3$ and a plane with a 45-degree angle with the xy-plane. From this perspective we can say that we actually sweep an arrangement of distance functions with a tilted plane in $\mathbb{R}^3$. For more information on the sweepline approach and the Voronoi diagrams see [33].

Based on the duality of Voronoi diagrams and arrangements, Rosenberger proposed a sweepline algorithm for the construction of order-k Voronoi diagrams of weighted points [76]. The idea of the method is due to Herbert Edelsbrunner [76] and it generalizes the Fortunes sweepline algorithm. In this method, one maintains an array of k x-monotone curves that move one after another and the last curve is used to construct the order-k Voronoi diagram, where the curves correspond to the first k levels of the arrangement of parabolas (see the definition in Section 2.4).

The principles of our algorithm are similar to Rosenberger's,[76] but it provides a simpler way to process the events without geometric transformations. Our algorithm can also be used to construct the order-k Voronoi diagram of polygonal objects such as: line segments and a planar straight-line graph.

In Section 5.1 we describe the sweepline algorithm for the higher-order Voronoi diagram of line segments. We investigate the properties of the algorithm and perform the time and space complexity analysis. In Section 5.2 we extend the algorithm to the line segments forming a PSLG.

5.1 Sweeping Disjoint Line Segments

In this section we describe the sweepline algorithm for disjoint line segments. We make a *general position assumption*: not more than three line segments touch the same disk and there are no endpoints that have the same horizontal or vertical coordinate.

Let ℓ be a horizontal line such that the halfplane ℓ^+ above ℓ intersects at least k line segments in $S^+ \subseteq S$.

Definition 4. *The* wave-curve $w(s)$ *is the locus of points equidistant from the line segment* $s \cap \ell^+$ *and the line* ℓ *(see Figure 5.1).*

Consider the arrangement A of wave-curves $w(s)$, $s \in S^+$. Following the definition of the i-level given in Section 2.1 a point x belongs to i-level A_i if there are at most i wave-curves passing strictly below and more than i wave-curves

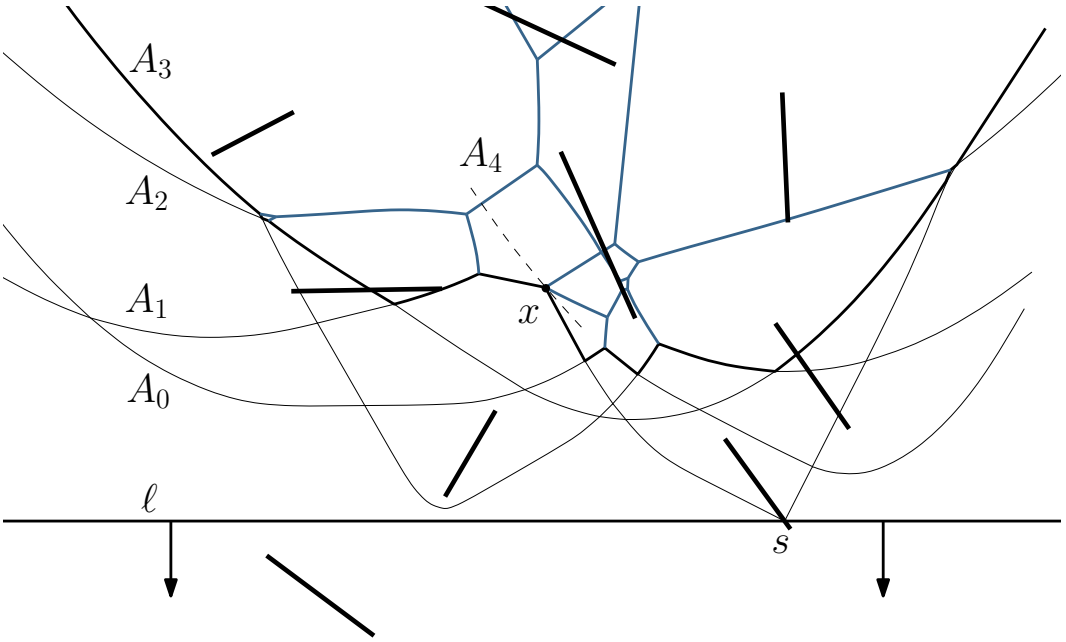

Figure 5.1. Constructing order-4 Voronoi diagram via sweepline technique.

passing below or through the point, i.e. $\pi^-(y) \leq i$ and $\pi^-(y) + \pi^0(y) > i$. The function Π defines an equivalence relation, making two points x and y equivalent iff $\Pi(x) = \Pi(y)$. The relation partitions the i-level into connected components of the equivalence classes.[41]

Definition 5. *The* wave *w is a connected component of the i-level that consists of more than a single point. The* breakpoint *is a connected component of the i-level that is a single point.*

Every point x on a fixed wave w has the same sets $\Pi^-(x)$ and $\Pi^0(x)$. Let $\Pi^-(w) = \Pi^-(x)$, $\Pi^0(w) = \Pi^0(x)$, $\pi^-(w) = \pi^-(x)$, and $\pi^0(w) = \pi^0(x)$, where x is any point on the wave w. Disjoint line segments correspond to the wave-curves that intersect transversely and each wave w of A_i has exactly i wave-curves below, i.e. $\pi^-(w) = i - 1$.

Note. It may look like we are giving an overcomplicated definition of the breakpoints and the waves. However, this definition will be useful for arrangements with non-transversal intersections in Section 5.2.

The algorithm sweeps the plane with the horizontal line while maintaining the levels $A_0, \ldots, A_{k-1}$. We store each level as an ordered list of waves that

allows search/insertion/deletion in logarithmic time. We store only the adjacency information of the waves and avoid storing their explicit shapes. Following the standard sweepline technique for the Voronoi diagrams[43, 76] there are two types of events that change the topology of the levels: *site-events* and *circle-events*. *Site-events* occur when the horizontal line encounters new sites, see Figure 5.2. *Circle-events* occur when the topology of the arrangement changes locally, which corresponds to a moment when the horizontal line touches the bottommost point of the disk that touches three sites, see Figure 5.3. The events are kept in a priority queue Q that allows insertion/deletion in logarithmic time. Inside Q the site-events are ordered by the y-coordinate of the topmost endpoint and the circle-events are ordered by the y-coordinate of the bottommost endpoint of the disk. We process events one by one as they appear in Q. Algorithm 1 summarizes the plane sweep at a high level.

Algorithm 1 Sweepline

```
1:  function Sweepline(S, k)
2:      Let V be empty set of vertices
3:      Let Q be S sorted by the y-coordinate of topmost endpoint
4:      Let A_i = (), for i = 0, ..., k − 1
5:      while Q not empty do
6:          x ← TopMost(Q)
7:          if x is a site-event then
8:              ProcessSiteEvent(x, k, A)
9:          else
10:             ProcessCircleEvent(x, k, A, V)
11:         end if
12:     end while
13:     return V
14: end function
```

A site-event occurs when a new wave-curve is introduced into the levels $A_0, \ldots, A_{k-1}$. A circle-event occurs when two or more breakpoints coincide. We keep track of circle-events by computing them in advance for every triple of consecutive waves. For every change in the topology of the arrangement we recompute the circle-events of the waves that are affected by the change. We encapsulate this procedure in the operation $UpdateTriples$ which is used in the Procedures 2 and 3. $UpdateTriples(A_i, r, m)$ adds circle-events that correspond to the newly created consecutive triples and removes those that do not correspond to consecutive triples anymore, where A_i is the level and $r, \ldots, m$ are the

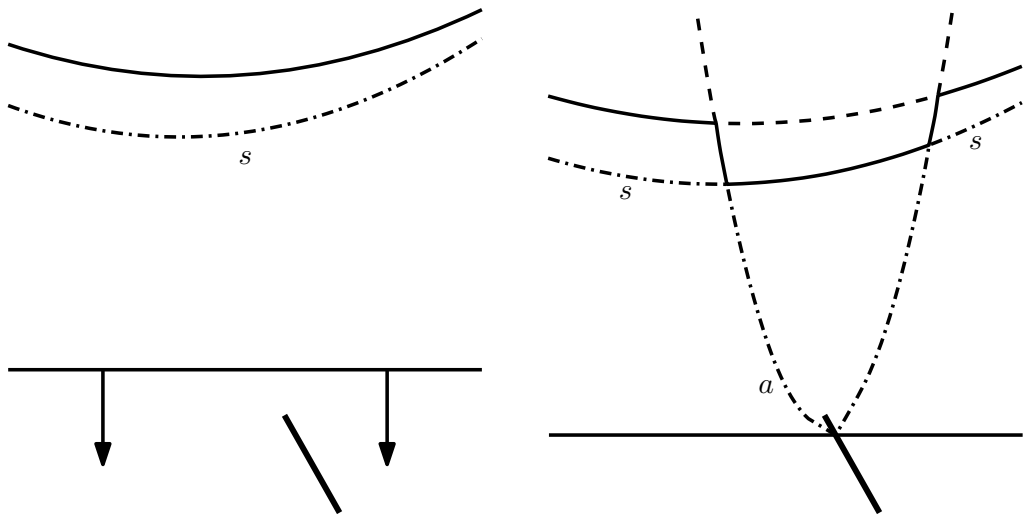

Figure 5.2. The site-event.

positions that should be considered during the update. We skip those triples that corresponds to the disks that are completely above the current position of the horizontal line.

We process site-events and circle-events by performing substitutions of waves in levels $A_0, \ldots, A_{k-1}$, see Procedures 2 and 3. The operation $Substitute(A_i, r, (a,b,c), (d,e,f))$ substitutes in level A_i at position r the subsequence a,b,c of waves with the subsequence d,e,f. The function $FindPosition(A_i, x)$ finds the position of the event x in level A_i and returns the position and the wave at this position. If x is a circle-event, then the function returns the index of the leftmost wave and the leftmost wave involved in the event.

As the horizontal line moves down the breakpoints of the $(k-1)$-level that do not belong to the $(k-2)$-level trace the edges of the order-k Voronoi diagram. The collision of two such breakpoints is a circle-event which has the center of the circle incident to two edges of the order-k Voronoi diagram. Subsequently the center of such a circle-event is an order-k Voronoi vertex. Note, however, that only some of the circle-events correspond to the order-k Voronoi vertices.

We can construct the order-k Voronoi diagram by maintaining the $(k-1)$-level. The levels $0, \ldots, k-2$ are needed to correctly maintain the topology of the $(k-1)$-level.

The following lemmas establish the correctness of the approach by showing that the portion of the order-k Voronoi diagram above the $(k-1)$-level does not

change as the sweepline moves down and the breakpoints of the $(k-1)$-level move along the edges of the order-k Voronoi diagram.

Lemma 5.1.1. *Let $x \in \ell^+$ and let $w(s_1), \ldots, w(s_m)$ be the wave-curves below or passing through the point x, and $m > 0$. Then line segments $s_1, \ldots, s_m$ are the m closest line segments to point x among the line segments in S.*

Proof. Consider the disk centered at the point x that touches the horizontal line ℓ at the point y. Let $p_1, \ldots, p_m$ be the intersection points of the wave-curves with the line segment xy. For each p_j, $j = 1, \ldots, m$ consider the disk D_j centered at the point p_j and touching the horizontal line ℓ. The definition of the wave-curve implies that the line segment s_j touches the disk D_j, $D_j \subseteq D_{j+1}$ and $D_j \subset \ell^+$. Therefore, the disk D_m touches or intersects the line segments $s_1, \ldots, s_m$.

Consider the ray emanating from point y through point x. Let $w(s')$ be the wave-curve that intersects the ray at point p' above x. Consider the disk D' centered at p' and touching the horizontal line ℓ. $D_m \subset D'$ and therefore the line segment s' does not touch or intersect the disk D_m.

The line segments $S \setminus S^+$ are below the horizontal line ℓ and since $D_m \subset \ell^+$, they do not touch or intersect the disk D_m. Therefore, disk D_m touches or intersects the line segments $s_1, \ldots, s_m$ and does not touch or intersect any other line segments. □

Lemma 5.1.1 implies a relation between the waves of the $(k-1)$-level and the regions of the order-k Voronoi diagram. Suppose x belongs to a wave on the level A_{k-1}. Let H be the set of sites that correspond to the wave-curves that pass below or through the point x, then Lemma 5.1.1 implies that H are the k closest sites to x, i.e. x belongs to the region $V_k(H, S)$.

Similarly, we can show a relation between the breakpoints of the $(k-1)$-level and the edges of the order-k Voronoi diagram. Suppose x is a breakpoint common to the levels A_{k-1} and A_k. Suppose x is an intersection point of two wave-curves $w(s)$ and $w(t)$. Let H be the set of sites strictly below point x, then Lemma 5.1.1 implies that $H \cup \{s, t\}$ are the $k+1$ nearest sites to x. Moreover, sites s and t are equidistant to point x and further from point x than the rest of the sites in H. Therefore, x belongs to the edge of the regions $V_k(H \cup \{s\}, S)$ and $V_k(H \cup \{t\}, S)$.

Suppose x is an intersection point of three wave-curves $w(s)$, $w(t)$ and $w(r)$ and it belongs to the levels A_{k-1}, A_k, A_{k+1}. Let H be the set of sites strictly below point x, then Lemma 5.1.1 implies that $H \cup \{s, t, r\}$ are the $k+2$ nearest sites to x. Since s, t, r are equidistant to x, x is a *new* order-k Voronoi vertex that is incident to the order-k Voronoi regions $V_k(H \cup \{s\}, S)$, $V_k(H \cup \{t\}, S)$ and $V_k(H \cup \{r\}, S)$.

Procedure 2 Process Site-Event

```
1: procedure ProcessSiteEvent(x, k, A)
2:     Let a be the site associated with event x
3:     s, r ← FindPosition(A_0, x)
4:     Substitute(A_0, r, (s), (s, a, s))
5:     UpdateTriplets(A_0, r, r + 2)
6:     s' ← s
7:     for i ← 1, ..., k − 1 do
8:         if L_i is empty then
9:             A_i = (x, s', x)
10:            exit for loop
11:        else
12:            s, r ← FindPosition(A_i, a)
13:            Substitute(A_i, r, (s), (s, a, s', a, s))
14:            UpdateTriples(A_i, r, r + 4)
15:            s' ← s
16:        end if
17:    end for
18: end procedure
```

Similarly, one can show that if x belongs to the levels A_{k-2}, A_{k-1}, A_k then it is an *old* order-k Voronoi vertex. In Figure 5.1 x is an intersection point of three levels A_3, A_4, A_5, and therefore x is a *new* Voronoi vertex of the order-4 Voronoi diagram.

The following lemma shows that the circle-events cause the waves to disappear from the lower levels and as the result appear on the upper levels.

Lemma 5.1.2. *Consider a circle-event at point $v \in A_{i-1}$ such that v is a new Voronoi vertex of $\mathcal{V}_i(S)$. At this event a single wave disappears from A_{i-1} and appears in A_{i+1}. Moreover two adjacent waves swap their positions in A_i.*

Proof. Let v be an intersection point of the wave-curves $w(s)$, $w(t)$ and $w(r)$. Consider the arrangement A' of the three wave-curves $w(s)$, $w(t)$ and $w(r)$. The part of the $(i-1)$-level in the arrangement A around point v corresponds to the 0-level in the arrangement A'. The 0-level in the arrangement A' is the *beach line* of the Fortune's algorithm.[43] According to Fortune's algorithm the circle-event makes a wave disappear from the *beach line*. Therefore, during the circle-event a single wave disappears from A_{i-1}. If now one considers the three wave-curves separately from the rest of the wave-curves, then the rest of the

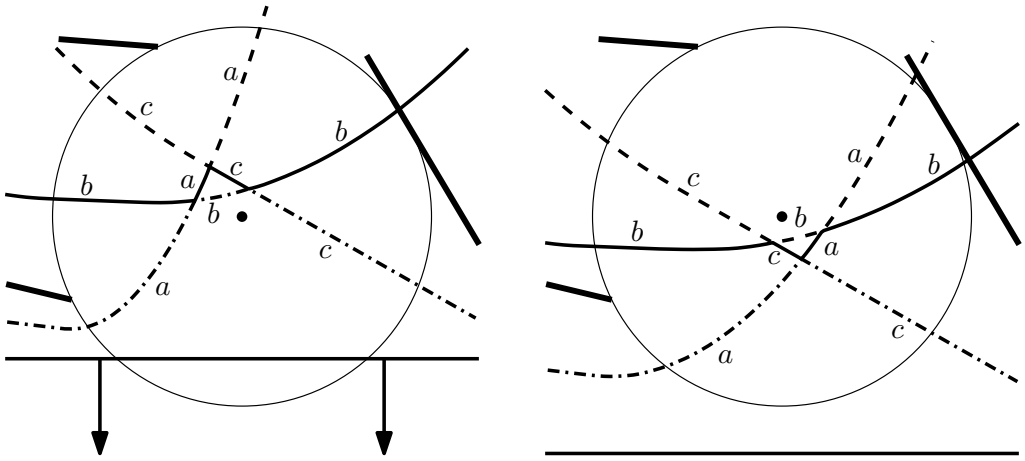

Figure 5.3. The circle-event on levels A_{bot},A_{mid},A_{top}.

claim will follow. □

Lemma 5.1.2 implies that a new wave may be introduced to the i-level by a site-event or from the levels below the i-level. This proves that the maintenance of the $(k-1)$-level requires only the maintenance of the levels below it. Therefore the first k levels are sufficient for the construction of the order-k Voronoi diagram.

The rest of this section provides space and time complexity bounds for the algorithm.

Lemma 5.1.3. *The maximum size of queue Q and the maximum total complexity of levels $A_0,\ldots,A_{k-1}$ are $O(nk)$.*

Proof. Since the wave-curves are Jordan curves the following bound holds:[30, 84]

$$g_{\leq k-1}(n) = O\left((k-1)^2 g_0(\lfloor n/(k-1)\rfloor)\right) \tag{5.1}$$

where $g_{\leq k-1}(n)$ is the maximum complexity of levels $A_0,\ldots,A_{k-1}$ and $g_0(m)$ is the maximum complexity of the lower envelope of m wave-curves. The lower envelope of waves corresponds to *beach line* of the order-1 Voronoi diagram,[43] therefore $g_0(m) = O(m)$. Thus the maximum complexity of $A_0,\ldots,A_{k-1}$, $g_{\leq k-1}(n)$ is equal to $O(nk)$.

The number of site-events is $O(n)$. Every circle-event in event queue Q corresponds to a triple of adjacent waves at some level A_i, $0 \leq i \leq k-1$. Therefore the number of circle-events is proportional to the total size of levels $A_0,\ldots,A_{k-1}$, which is $O(nk)$. Thus Q is of size $O(nk)$. □

Procedure 3 Process Circle-Event

```
1: procedure ProcessCircleEvent(x, k, A, V)
2:     Let A_bot, A_mid, A_top be the three levels that involve the event x, from bottom to top
3:     a, botPos ← FindPosition(A_bot, x)
4:     b, midPos ← FindPosition(A_mid, x)
5:     c, topPos ← FindPosition(A_top, x)
6:     if bot = k − 1 or bot = k − 2 then
7:         V = V ∪ {x}                ▷ vertex of x with the three associated edges
8:         Substitute(A_bot, botPos, (a, b, c), (a, c))
9:         UpdateTriples(A_bot, botPos, botPos + 2)
10:    else if bot ≤ k − 2 then
11:        Substitute(A_mid, midPos, (b, a, c, b), (b, c, a, b))
12:        UpdateTriplets(A_mid, midPos, midPos + 3)
13:    else if bot ≤ k − 3 then
14:        Substitute(A_top, topPos, (c, a), (c, b, a))
15:        UpdateTriplets(A_top, topPos, topPos + 2)
16:    end if
17: end procedure
```

Theorem 5.1.4. *The algorithm can be implemented to run in $O(k^2 n \log n)$ time and $O(nk)$ space.*

Proof. The site-events correspond to the insertion of the new wave-curves in levels $A_0, \ldots, A_{k-1}$. The number of site-events is bounded by the number of sites, $O(n)$. When a new line segment intersects a halfplane ℓ^+ we insert it in lists $A_0, \ldots, A_{k-1}$. This requires a binary search on every list and therefore it takes $O(\log |A_i|)$ per list, where $|A_i|$ denotes the size of the list. Since the maximum complexity of A_i is bounded by the structural complexity of the order-i Voronoi diagram, we need $O\left(\log\left(i(n-i)\right)\right) = O(\log n)$ per level A_i, or $O(k \log n)$ for all levels. Therefore it takes $O(nk \log n)$ time to process all the site-events.

The circle-events correspond to the Voronoi vertices of the order-i Voronoi diagrams, $i = 0, \ldots, k-1$. Every such event requires constant time. Since the number of order-i Voronoi vertices is bounded by $O(i(n-i))$ thus it implies that the total number of circle-events is bounded by $\sum_{i=1}^{k} O(i(n-i)) = O(k^2 n)$. Every site-event and circle-event requires an update of the triples that involve the line segments that are adjacent to the places where the changes occurred. Insertions and deletions into the event queue Q require $O(\log |Q|)$ time per each inserted/removed circle-event, where $|Q|$ - is the size of the queue. Lemma 5.1.3

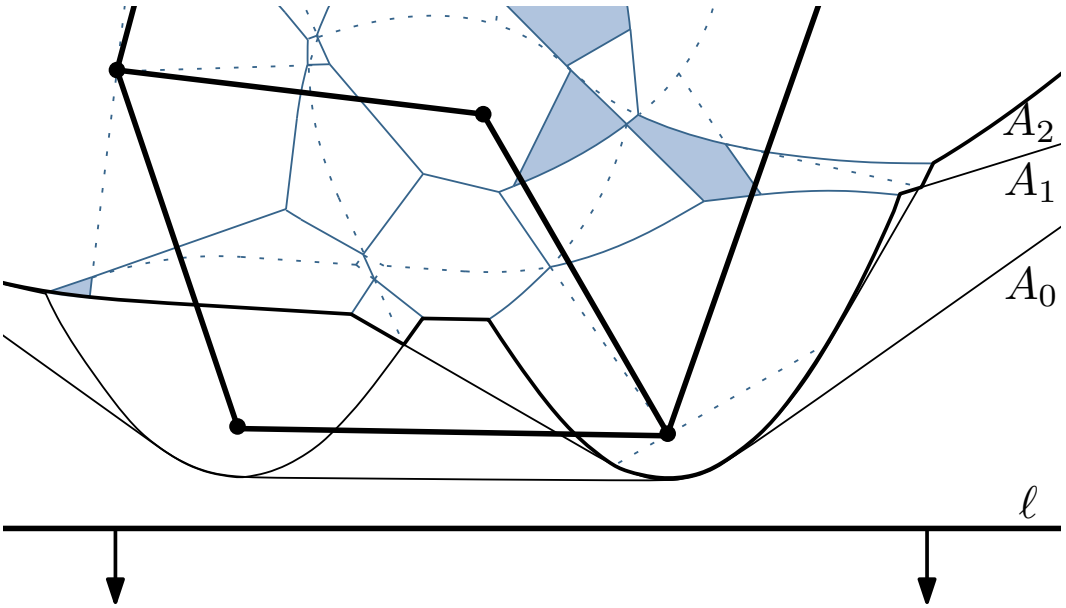

Figure 5.4. Constructing the order-3 Voronoi diagram via the sweepline technique. The dotted lines depict the internal edges. The Type-2 regions are depicted shaded.

implies that the size of the queue is $O(nk)$. Therefore it takes $O(\log(nk)) = O(\log n)$ time per event. And the total running time is $O\left(k^2 n \log n\right)$.

During the execution of the algorithm we store the event queue Q, lists $A_0, \ldots, A_{k-1}$ and we output the order-k Voronoi diagram $\mathcal{V}_k(S)$. Then Lemma 5.1.3 implies the total space complexity. □

5.2 Sweeping a Planar Straight-Line Graph

In this section we adopt the algorithm for a PSLG. The line segments forming a PSLG correspond to the wave-curves that may intersect non-transversely. There are two types of breakpoints (see Figure 5.11):

- *X-breakpoints* are incident to four waves;
- *Y-breakpoints* are incident to three waves.

Note, however, that some of the breakpoints of A_{k-1} do not move along the edges of the order-k Voronoi diagram. The breakpoints may move along the bisectors

between the elementary sites that do not correspond to the edges of the order-k Voronoi diagram.

The plane sweep construction for a PSLG requires to explicitly distinguish two types of Voronoi edges.

- *External* Voronoi edges bound the order-k Voronoi regions.
- *Internal* Voronoi edges belong to the interior of the order-k Voronoi regions and they represent the farthest subdivision inside the order-k Voronoi regions.

In Figure 5.4 some of the external edges of the order-2 Voronoi diagram are the internal edges of the order-3 Voronoi diagram in Figure 5.4 (see the dotted lines). Type-2 Voronoi regions do not contain internal edges, since there is no farthest subdivision inside the Type-2 faces, see Chapter 4.

The same wave may appear on multiple levels. We store each wave separately and only link it with the corresponding positions at each level. With each wave w we store: the set $\Pi^0(w)$ of the wave-curves that pass through the wave, and the number $\pi^-(w)$ of the wave-curves strictly below the wave. The following lemma states that this information is enough to determine the type of the order-k Voronoi edge.

Lemma 5.2.1. *Let x be a breakpoint on the $(k-1)$-level which is incident to waves $w_1,\ldots,w_r$, where r equals 3 or 4. Suppose we know the sets $\Pi^0(w_1),\ldots,\Pi^0(w_r)$ and the numbers $\pi^-(w_1),\ldots,\pi^-(w_r)$. Then we can determine whether the point x belongs to an edge of the order-k Voronoi diagram, and if it does whether it is an internal or an external edge.*

Proof. Consider the disk centered at point x that touches the horizontal line ℓ. Using the sets $\Pi^0(w_1),\ldots,\Pi^-(w_r)$ of the incident waves we can determine the line segments that touch the disk. We can also determine the number of line segments intersected by the interior of the disk by using the numbers $\pi^-(w_1),\ldots,\pi^-(w_r)$ of the incident waves. Here we describe the possible cases that may occur and the corresponding type of the order-k Voronoi edge.

Point x belongs to the internal edge if: (1) The interior of the disk intersects $k-2$ line segments and the boundary touches two line segments at different points; (2) The interior of the disk intersects $k-2$ line segments and the boundary touches two line segments at the same endpoint, but one of the line segments is also tangent to the disk.

Point x belongs to the external edge if: (3) The disk touches or intersects at least $k+1$ line segments and the boundary of the disk touches at least two line

segments in two different points; (4) The interior of the disk intersects $k-1$ line segments and the boundary touches two line segments at the same endpoint, but one of the line segments is also tangent to the disk.

(5) Point x does not belong to an edge of the order-k Voronoi diagram if the disk touches or intersects at least $k+1$ line segments and the boundary touches the line segments at the same point (and it is not the case (4)). □

While the sweep is performed the neighboring breakpoints may collide producing a circle-event. If a pair of breakpoints of the $(k-1)$-level that correspond to external or internal edges meet at the point x then x is an order-k Voronoi vertex. If the circle-event involves only internal Voronoi edges then the corresponding Voronoi vertex is internal, i.e. belongs to the interior of the order-k Voronoi region.

To summarize, the algorithm sweeps the plane with the horizontal line, while maintaining k levels $A_0, \ldots, A_{k-1}$. The breakpoints of the $(k-1)$-level move along the external and the internal edges of the order-$(k-1)$ Voronoi diagram of a planar straight-line graph. We use Lemma 5.2.1 to determine the type of the edge associated with every breakpoint on the $(k-1)$-level. When the circle-event occurs, several breakpoints meet at a common point. We can determine the type of the vertex that is created during the circle-event using the knowledge of the type of the associated edges. The algorithm outputs Voronoi vertices and incident Voronoi edges every time it handles events at which Voronoi edges meet. In order to process the events the algorithm computes the small subarrangements of wave-curves separately and then inserts them into the main arrangement.

A high-level description of the sweepline algorithm for a planar straight-line graph is the same as for disjoint line segments, see Algorithm 1. However, there is a major difference in how we process the events. Since the line segments are abutting the site-events may involve more than a single line segment. Moreover, the non-transversal intersections may cause the circle-events to involve more than three levels. In the following we describe in detail the different cases that may occur with site-events and circle-events.

5.2.1 Processing Site-Events

Unlike disjoint line segments where it is sufficient to process only the top endpoint of the line segment, in the case of line segments forming a planar straight-line graph we need to process all vertices of the PSLG, i.e. all segment endpoints.

With every site-event p we store: (1) The y-coordinate of the endpoint p; (2) The set of line segments $I(p)$ incident to p.

Let p be an endpoint that corresponds to a site-event. Let $I(p)$ be the set of line segments incident to the point p. Let $I^-(p)$ and $I^+(p)$ be the sets of line segments incident to point p that are below and above the horizontal line through p, respectively. Let $s_1, \ldots, s_m$ be the line segments in $I^+(p)$ ordered from left to right, $t_1, \ldots, t_r$ be the line segments in $I^-(p)$ ordered from left to right and $w_1, \ldots, w_m$ and $v_1, \ldots, v_r$ be the corresponding wave-curves. Then before the line ℓ hits endpoint p the line segments in $I^-(p)$ do not contribute the wave-curves to the arrangement. First, we describe the two special cases ($I^-(p) = \emptyset$ and $I^+(p) = \emptyset$) which give us an insight on the general case that we describe later.

Site-Events with Empty Bottom

Consider the case when the set $I^-(p)$ is empty. The way the *topology changes at the moment of the site-event depends only on the ordering of the line segments* $s_1, \ldots, s_m$ and does not depend on the angles between them, i.e. if we rotate the line segments $s_1, \ldots, s_m$ while preserving their order, the topology will not change. Figure 5.7 shows a site-event with empty bottom, where $m = 4$ and $i = 1$ and the labels denote the sets Π^0 of the waves on the i-level.

Consider also Figure 5.6 which shows two site-events with empty bottoms. The two top figures show the waves before and after the first site event occurs. The two bottom figures show the waves before and after the second site event occurs. Note in both cases the number of line segments involved is the same, however, the angles between the line segments are different. However, the way the topology changes in both cases is the same.

Site-Events with Empty Top

Consider the case when the set $I^+(p)$ is empty. As in the previous case, the *topological changes do not depend on the angles between the line segments* $t_1, \ldots, t_r$. The difference with the previous case is that the line segments $t_1, \ldots, t_r$ do not contribute to the arrangement before the site-event occurs. See Figure 5.7, where $r = 4$, $i = 1$ and a_1, a_2 are the wave-curves of the waves above the point p, before the site-event occurs.

General Site-Events

Consider the general case, when $I^-(p)$ and $I^+(p)$ are not empty. Unfortunately, in this case the site-event creates the topological changes that *depend not only on*

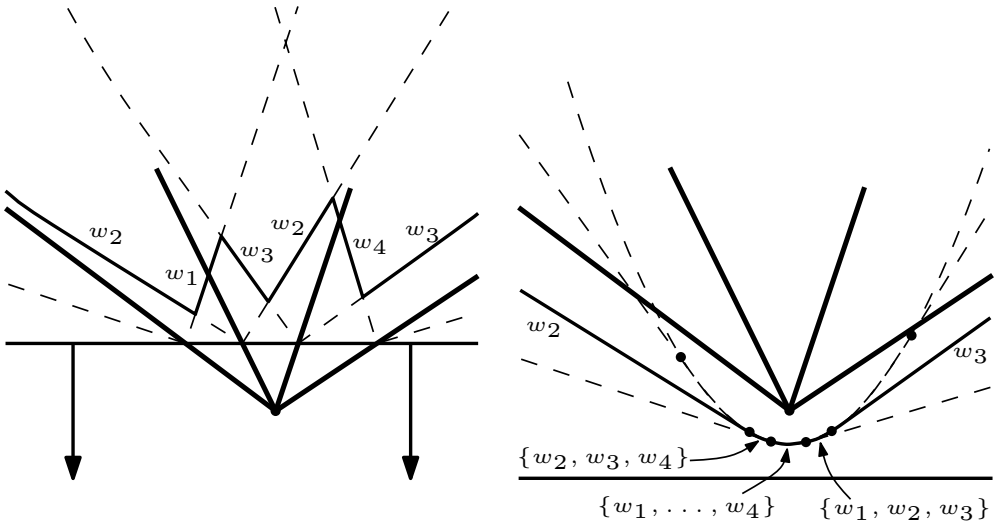

Figure 5.5. Wave-curves of the line segments $I^+(p)$ before (left figure) and after (right figure) the site-event occurs. The 1-level is depicted with bold. The labels show the sets Π^0 of the waves.

the ordering of the line segments in $I^-(p)$ and $I^+(p)$ but on the angles between the line segments. Figure 5.9 illustrates the general site-event. The 1-level is shown in bold, and its structure after the event occurs depends on the angles between the line segments.

Consider also the Figure 5.8 which shows two general site-events. In both cases the number of line segments involved is the same: there is one line segment s_1 in $I^+(p)$ and one line segment t_1 in $I^-(p)$. The difference is in the angles between s_1 and t_1. In both cases the 0-level is the same before the site-event occurs. However, the 0-level is different after the site-event occurs. In the first case, 0-level is composed of two waves with the following sets Π^0: $\{w_1\}, \{v_1\}$. In the second case, 0-level is composed of two waves with the following sets Π^0: $\{v_1\}, \{w_1\}$. The 1-level is also different in both cases.

In this case we do not try to derive the changes from the order of the line segments in $I^-(p)$ and $I^+(p)$. Instead we consider the moment in time right after the site-event occurs and compute the arrangement L of wave-curves in $I(p)$. Then we remove those waves that correspond to the line segments in $I^-(p)$ and insert the arrangement L in the arrangement A.

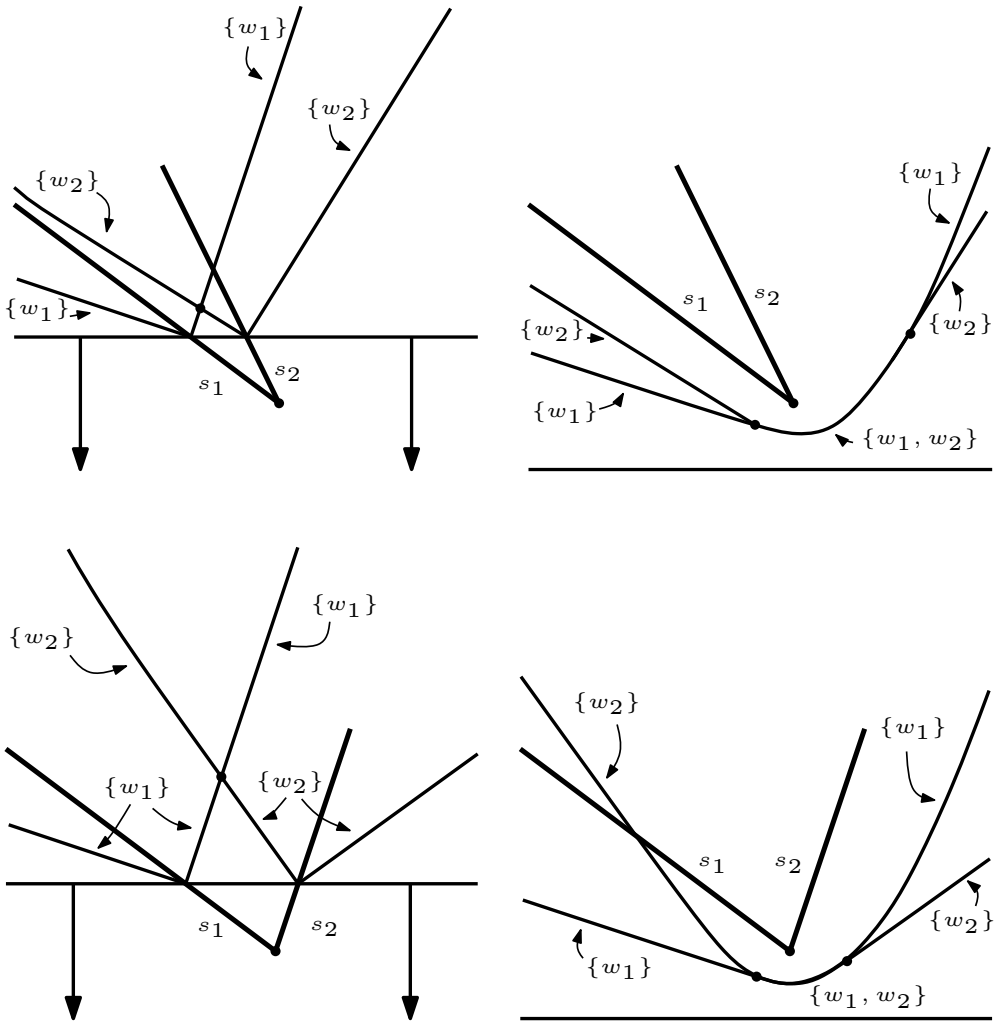

Figure 5.6. Wave-curves of the line segments $I^+(p)$ before (left figures) and after (right figures) the site-event occurs. Before the site-event occurs, the 0-level is composed of the waves with the following sets Π^0: $\{w_1\}, \{w_2\}$. The 1-level is composed of the waves with the following sets Π^0: $\{w_2\}, \{w_1\}$. After the site-event occurs, the 0-level is composed of the waves with the following sets Π^0: $\{w_1\}, \{w_1, w_2\}, \{w_2\}$. The 1-level is composed of the waves with the following sets Π^0: $\{w_2\}, \{w_1, w_2\}, \{w_1\}$. The way the topology of the levels changes during the site-event does not depend on the angles between the line segments in $I^+(p)$.

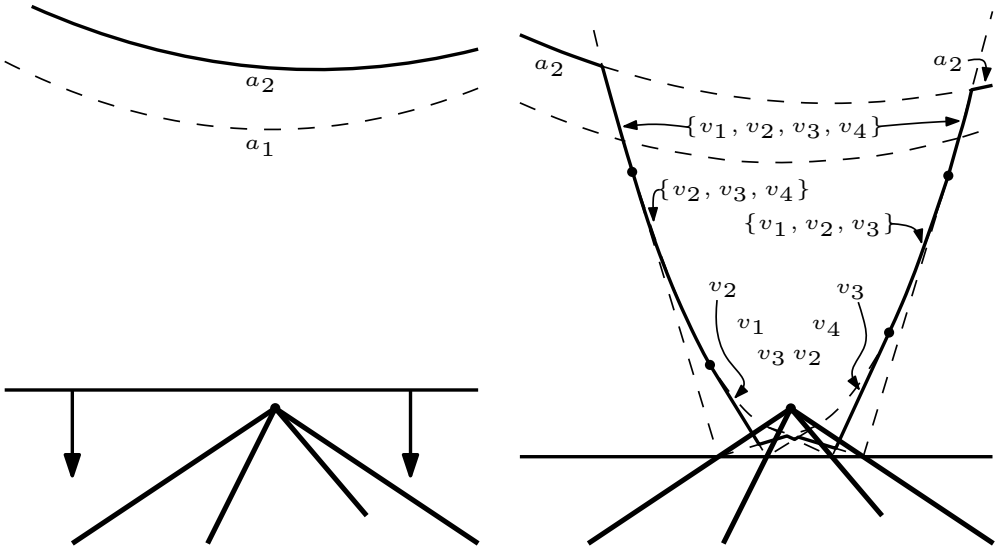

Figure 5.7. Wave-curves of the line segments $I^-(p)$ before (left figure) and after (right figure) the site-event occurs. The 1-level is depicted with bold. The labels show the sets Π^0 of the waves. a_1 and a_2 are the wave-curves strictly below point p, before the site-event occurs.

5.2.2 Processing Circle-Events

During the circle-event the topological change is local and it involves a number of levels equal to the number of line segments that touch the disk. For instance, in Figure 5.10 there are 6 levels involved.

With every circle-event we store: (1) The y-coordinate of the bottommost point of the disk; (2) The line segments that touch the disk; (3) The pointers to the levels that involve the event.

Under the *weak general position assumption* a circle-event involves at most one *Y-breakpoint*. The circle-event may involve face appearance or disappearance in the arrangement. During the circle-event two or three breakpoints meet and are replaced with two or three other new breakpoints. Consequently, there are five cases (two pairs of which are symmetric with respect to the y-axis) of circle-events that can be characterized by the topological changes associated to them:

1. Three X-breakpoints are replaced with other three X-breakpoints, see Figure 5.10;

2. One Y-breakpoint and two X-breakpoints (considered from left to right) are

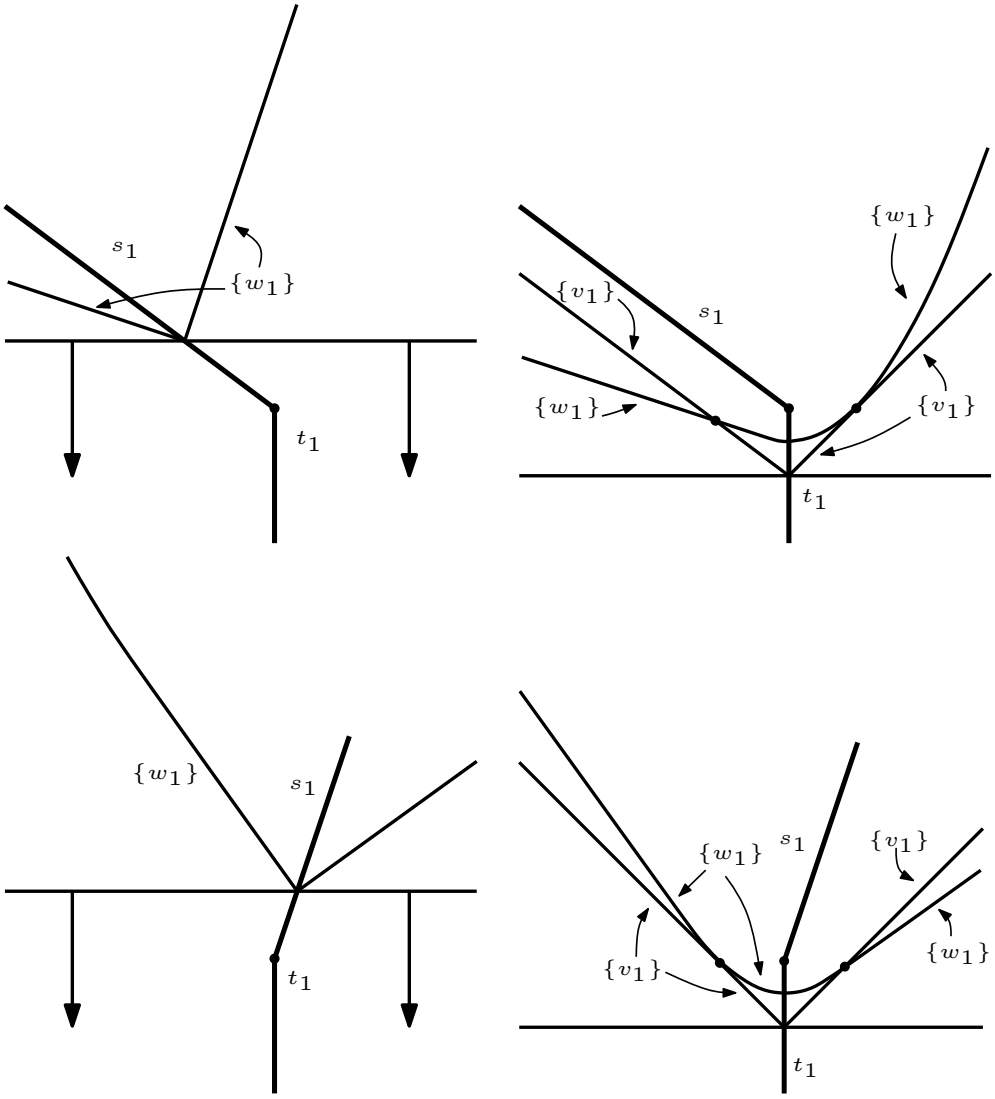

Figure 5.8. Wave-curves of the line segments $\{s_1\} = I^+(p)$, $\{t_1\} = I^-(p)$ before (left figures) and after (right figures) the site-event occurs. Before the site-event occurs, the 0-level is composed of a single wave with the following set Π^0: $\{w_1\}$. What happens with the waves after the site-event occurs depends on the angles between the line segments. In the top case, after the site event occurs, the 0-level is composed of the waves with the following sets Π^0: $\{w_1\}, \{v_1\}$. The 1-level is composed of the waves with the following sets Π^0: $\{v_1\}, \{w_1\}$. In the bottom case, after the site event occurs, the 0-level is composed of the waves with the following sets Π^0: $\{v_1\}, \{w_1\}$. The 1-level is composed of the waves with the following sets Π^0: $\{w_1\}, \{v_1\}$.

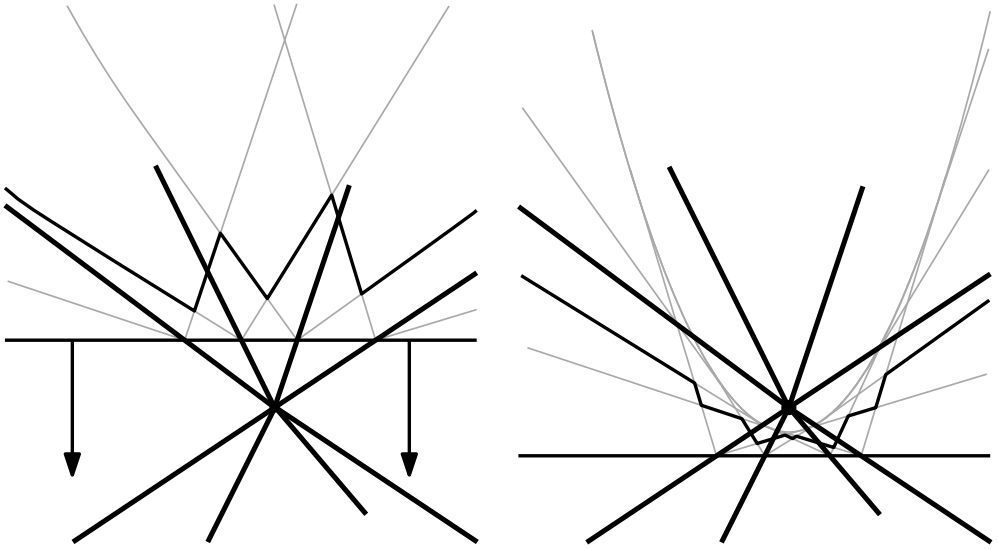

Figure 5.9. Wave-curves before and after the general site-event. The 1-level of wave-curves is bold. The way the topology of the 1-level changes during the general site-event depends on the angles between the line segments.

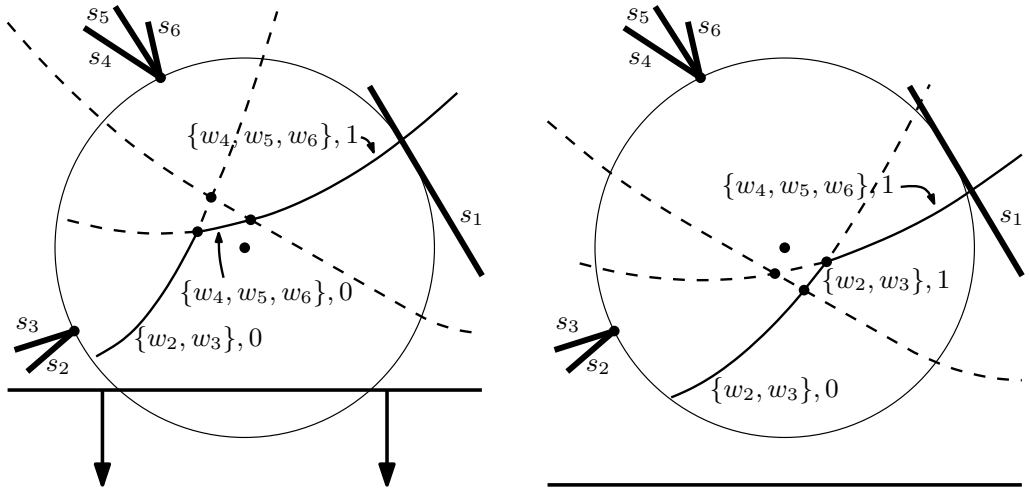

Figure 5.10. The case (1) of the circle-event in arrangement with non-transversal intersections. The 1-level is depicted with bold. The waves of the 1-level are labeled with the pairs Π^0, π^-.

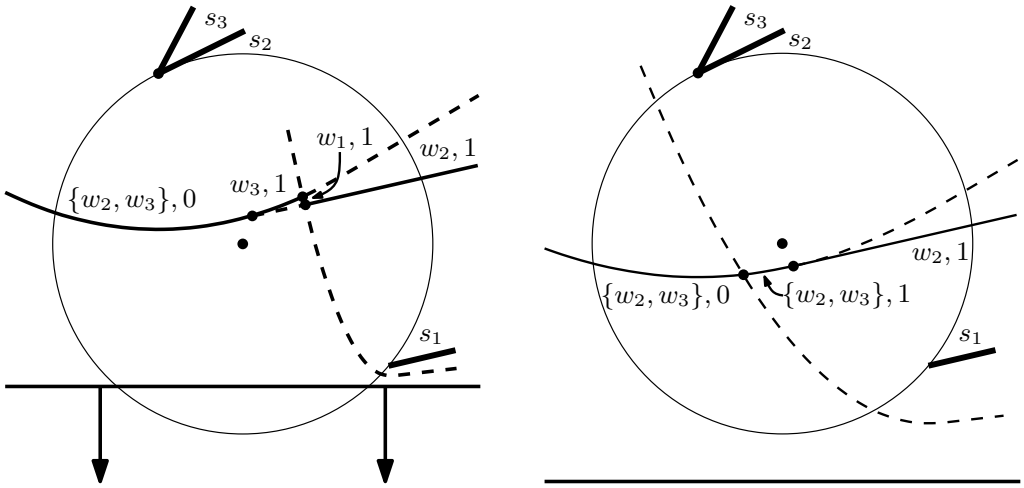

Figure 5.11. The case (2) of the circle-event in arrangement with non-transversal intersections. The 1-level is depicted with bold. The waves of the 1-level are labeled with the pairs Π^0, π^-.

replaced with one X-breakpoint and one Y-breakpoint. A face disappears, see Figure 5.11;

3. One X-breakpoint and one Y-breakpoint (considered from left to right) are replaced with one Y-breakpoint and two X-breakpoints. A face appears, see Figure 5.12.

We process the circle-event in the following way: First, we remove all waves that are incident to any two breakpoints of the circle-event. Then we update the adjacency relations between the waves that are incident to one breakpoint of the circle-event. Finally, we consider those wave-curves that participate in the circle-event and compute their arrangement after the circle-event occurs. Then we insert the new computed waves that are incident to any two breakpoints of the circle-event.

5.2.3 Analysis

The sweepline algorithm can be applied to any kind of planar straight-line graph. However, it is useful only for the case when the vertices of PSLG have $O(1)$ degree. For instance, consider the case when n line segments are all incident to the same endpoint. In this case the vertex of the Voronoi diagram that corresponds

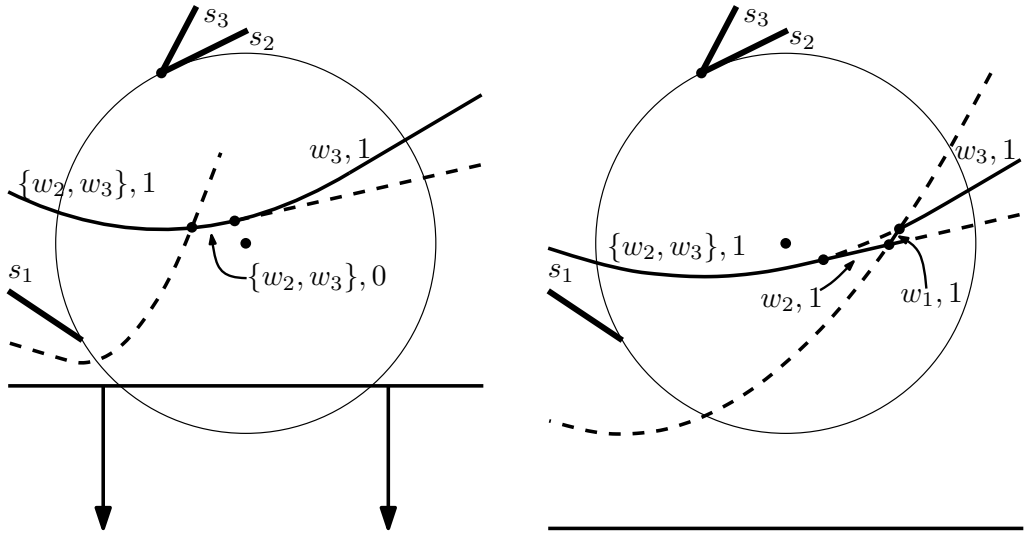

Figure 5.12. The case (3) of the circle-event in arrangement with non-transversal intersections. The 1-level is depicted with bold. The waves of the 1-level are labeled with the pairs Π^0, π^-.

to the endpoint contributes the major complexity of the Voronoi diagram. However if we process PSLG with a sweepline approach we should do all the computations associated with the vertex at the same moment of time. This negates the benefits of the sweepline approach which performs the sequential construction of the Voronoi diagram.

Theorem 5.2.2. *If the vertices of a PSLG have $O(1)$ degree then the algorithm can be implemented to run in $O(k^2 n \log n)$ time.*

Proof. The structural complexity in case of a PSLG is $O(k(n-k))$. However, most of the results on the structural complexity of the k-level of Jordan curves require tranvsersality, to the best of our knowledge.[39, 84] Therefore, in the case of the wave-curves that correspond to the line segments of a PSLG we use a naive structural complexity bound of the k-level, $O(n^2)$. Since we assume that the vertices of a PSLG have $O(1)$ degree it implies that the time required to process the events is not greater than in the case of disjoint line segments. Therefore it takes $O(nk \log n^2) = O(nk \log n)$ total time to process all site-events.

The number of circle-events is bounded by the number of Voronoi vertices in order-i Voronoi diagrams, for $i = 1, \ldots, k$. Since the structural complexity of the order-i Voronoi diagram is $O(i(n-i))$ the total number of circle-events is bounded by $\sum_{i=1}^{k} O(i(n-i)) = O(k^2 n)$. The processing takes $O(\log n^2)$ time

per each circle-event. Therefore the total time complexity of the algorithm is $O(k^2 n \log n)$. □

5.3 Summary

In this chapter we have presented a sweepline algorithm for higher-order Voronoi diagrams of line segments. The sweepline algorithm has $O(k^2 n \log n)$ time complexity and $O(kn)$ space complexity, similarly to the iterative algorithm. The idea of the algorithm is similar to that of Rosenberger's, but has a different way of processing the events and it can be applied to the case of line segments. It constructs not only the order-k Voronoi diagram, but all order-i Voronoi diagrams for $i \leq k$. The algorithm is simple to implement and it allows the top-to-bottom sequential construction of the diagram. It does not require to maintain the entire diagram during the construction and it can therefore be applied for on-the-fly computations (see Section 1.1.1).

We have also adopted the algorithm to handle the case of line segments forming a planar straight-line graph. The algorithm can be applied to any planar straight-line graph. However, when PSLG has vertices of an arbitrary degree, the benefits of the sweepline algorithm can be negated by the high-degree vertices. We provide the time complexity analysis for the case of a PSLG with the $O(1)$ vertices degree and prove $O(k^2 n \log n)$ time complexity and $O(n^2)$ space complexity.

Chapter 6

Algorithms for Higher-Order Abstract Voronoi Diagrams

In this chapter, we develop a randomized divide and conquer algorithm, a random walk algorithm and a randomized iterative algorithm to compute the order-k abstract Voronoi diagram in $O(kn^{1+\varepsilon})$, $O(n^2 2^{\alpha(n)} \log n)$ and $O(nk^2 \log n)$ expected time complexities, respectively.

Abstract Voronoi diagram is a high-level framework for Voronoi diagrams suggested by Klein [53] and later improved by Klein et al. [54]. In this framework the Voronoi diagram is defined in terms not dependent on the concrete geometry, metric space or shapes of the sites. Instead abstract Voronoi diagrams are defined using bisecting curves. For a set of n abstract sites we know nothing except their pairwise bisectors. We assume that these bisecting curves are *nice*, in the sense that they assume the axioms listed below. Once a bisector system of a concrete Voronoi diagram is shown to satisfy these axioms, combinatorial properties and algorithms to construct abstract Voronoi diagrams are directly applicable.

Let us assume for a set of n sites S the following bisecting system $\mathcal{J} = \{J(p,q) \mid p,q \in S, p \neq q\}$. A bisector $J(p,q)$ partitions the plane into two domains $D(p,q)$ and $D(q,p)$ where $D(p,q)$ are points considered to be closer to p than q.

Definition 6. *The nearest neighbor Voronoi region of site p is*

$$V(p,S) = \bigcap_{q \in S, q \neq p} D(p,q).$$

The nearest neighbor Voronoi diagram is

$$\mathcal{V}(S) = \bigcup_{p \in S} \partial V(p, S),$$

where ∂ denotes the boundary.

The bisector system $\mathcal{J}$ satisfies the following axioms for any $S' \subseteq S$:

(A1) Each first-order Voronoi region $V_1(p, S')$ is pathwise connected.

(A2) Each point in the plane belongs to the closure of $V_1(p, S')$, for a site $p \in S'$.

The axioms are strong enough to prove that for any three different sites p, q, r the following holds [53, 54]:

$$D(p,q) \cap D(q,r) \subseteq D(p,r) \tag{6.1}$$

This property is enough to prove that the structural complexity of the abstract Voronoi diagram is linear, $O(n)$.

Without any further assumptions one can apply a randomized incremental approach to construct the abstract Voronoi diagram, see [55, 61]. The algorithm has $O(n \log n)$ expected time complexity and $O(n)$ expected space complexity.

The abstract counterpart for the farthest Voronoi diagram was studied by Rasch [75] and Mehlhorn et al. [62]. The farthest abstract Voronoi diagram has a tree structure and $O(n)$ structural complexity. It can be also constructed with a randomized algorithm in $O(n \log n)$ expected time.

The axioms (A1)–(A2) are satisfied for many concrete Voronoi diagrams, like: power diagrams[1], additively weighted points[1], non-intersecting line segments, disjoint convex polygons. Therefore, the structural complexity bound and the randomized incremental algorithm are applicable to all these concrete cases.

There are other concrete Voronoi diagrams which are not covered by the abstract Voronoi diagrams framework, because their bisectors do not satisfy the axioms (A1)–(A2). For example, multiplicatively weighted points[1], intersecting line segments, line segments forming a planar straight-line graph, intersecting convex polygons, intersecting circles, etc. Also disjoint simple polygons do not satisfy the axioms, since they may have closed bisectors.

[1] In power diagrams, additively weighted points and multiplicatively weighted points the distance between the point p and the weighted point s is measured as $d_{\text{power}}(p,s) = d^2(p,s) - w(s)$, $d_{\text{add}}(p,s) = d(p,s) - w(s)$ and $d_{\text{mult}}(p,s) = d(p,s)/w(s)$, respectively, where $d(p,s)$ is the regular distance measure and $w(s)$ is the weight of the point s.

6.1 Higher-Order Abstract Voronoi Diagrams

Higher-order abstract Voronoi diagrams were recently introduced by Bohler et al. [17]. The concept of the higher-order abstract Voronoi diagram extends the ordinary abstract Voronoi diagram to an arbitrary order k. In addition to axioms (A1)–(A2) the extended framework assumes the following axioms for any $S' \subseteq S$:

(A3) No first-order Voronoi region $V_1(p, S')$ is empty.

(A4) Each curve $J(p,q)$, where $p \neq q$, is unbounded. After stereographic projection to the sphere, it can be completed to be a closed Jordan curve through the north pole.

(A5) Any two curves $J(p,q)$ and $J(s,t)$ have only finitely many intersection points, and these intersections are transversal.

Definition 7. *The order-k Voronoi region associated with H is*

$$V_k(H,S) = \bigcap_{p\in H, q\in S\setminus H} D(p,q).$$

The order-k Voronoi diagram is

$$\mathcal{V}_k(S) = \bigcup_{|H|=k} \partial V_k(H,S),$$

where ∂ denotes the boundary.

The number of faces in the order-k abstract Voronoi diagram of n sites is less or equal than $2k(n-k)$ and the bound is tight [17].

Even though we are working in an abstract setting where no notion of distance is defined, it is sometimes easier to think in "nearest" terms about a particular site. Therefore we define the following relation that establishes a linear order on the sites in S for a fixed point $x \in \mathbb{R}^2$.

Definition 8. *For a point $x \in R^2$ and two sites $p, q \in S$, $p <_x q$, $p =_x q$, or $p >_x q$ if $x \in D(p,q)$, $x \in J(p,q)$, or $x \in D(q,p)$, respectively.*

Eq. (6.1) implies the transitivity of the $\leq_x$ relation. Suppose $p \leq_x q$ and $q \leq_x r$, then $x \in D(p,q)$ and $x \in D(q,r)$. Since $D(p,q) \cap D(q,r) \subseteq D(p,r)$, $x \in D(p,r)$ and thus $p \leq_x r$. We define an ordered sequence on S, $\pi_x^S = (s_1, \ldots, s_n)$, given x, satisfying $s_1 \leq_x s_2 \leq_x \ldots \leq_x s_n$. We say that site s is *k-nearest* to point x if s occupies the k-th position in the sequence π_x^S.

For each point $x \in V_k(H,S)$ and $\pi_x^S = (s_1,\ldots,s_n)$, $H = \{s_1,\ldots,s_k\}$, and $s_k <_x s_{k+1}$. If $V_k(H_1,S)$ and $V_k(H_2,S)$ share an edge e, then for any point $x \in e, H_1 \cap H_2 = \{s_1,\ldots,s_{k-1}\}$ and $s_{k-1} <_x s_k =_x s_{k+1}$, see [17, Lemma 5]. For simplicity, throughout this chapter, we make a general position assumption that the degree of any Voronoi vertex is exactly three.

Similarly to Section 3.2 we define two types of Voronoi vertices:

Definition 9. *Let v be a Voronoi vertex among $V_k(H_1,S)$, $V_k(H_2,S)$, and $V_k(H_3,S)$, and let $H = H_1 \cap H_2 \cap H_3$ then v can be categorized into two types:* new *when $|H| = k-1$ and* old *when $|H| = k-2$.*

A *new* Voronoi vertex of $\mathcal{V}_k(S)$ is an *old* Voronoi vertex of $\mathcal{V}_{k+1}(S)$.

Let v be a Voronoi vertex as in the Def. 9, we can show that $H = \{s_1,\ldots,s_t\}$ and $s_t <_v s_{t+1} =_v s_{t+2} =_v s_{t+3} <_v s_{t+4}$, where $t = |H|$ and $\pi_v^S = (s_1,\ldots,s_n)$. Each Voronoi vertex is defined by the three sites $s_{t+1}, s_{t+2}, s_{t+3}$.

The following is the definition of the key substructure of the higher-order Voronoi diagram which is going to be used by the algorithm in Section 6.4.

Definition 10. *The k-*neighborhood *of a site p in S, denoted by $VN_k(p,S)$, is the union of closures of $V_k(H,S)$ for all $H \subset S$, such that $p \in H$ and $|H| = k$, i.e.,*

$$VN_k(p,S) = \bigcup_{p\in H, H\subset S, |H|=k} \overline{V_k(H,S)},$$

where $\overline{X}$ denotes the topological closure of the set X.

Each edge of $\partial VN_k(p,S)$ belongs to $J(p,q)$ for a site $q \in S \setminus \{p\}$, and each edge of $\mathcal{V}_k(S)$ belongs to $\partial VN_k(p,S)$ for a site $p \in S$, the latter implies

$$\mathcal{V}_k(S) = \bigcup_{p\in S} \partial VN_k(p,S). \tag{6.2}$$

Unlike order-k Voronoi regions of point-sites, abstract order-k Voronoi regions may be disconnected. In fact one region may disconnect into $\Omega(n)$ disjoint faces, for $k > 1$ (see e.g. Section 3.1 for line segments). Nevertheless, the *k-neighborhood* is connected, and this is the major property used in Section 6.4.

Lemma 6.1.1. *$VN_k(p,S)$ is simply connected and there is no finite set of points whose removal would make $VN_k(p,S)$ disconnected.*

Proof. First we show that $VN_k(p,S)$ is path connected. For the sake of contradiction suppose there exists a curve L that contains at most finitely many

points of $VN_k(p,S)$ and separates parts of it. Consider point $x \in VN_k(p,S)$. Site p is one of the k closest sites to point x, therefore there are at least $n-k$ sites $\{q_1,\ldots,q_{n-k}\} = Q \subset S$ that are further from x than the site p, i.e. $x \in \overline{\bigcap_{i=1}^{n-k} D(p,q_i)} = \overline{V_1(p,Q\cup\{p\})}$. The region $V_1(p,Q\cup\{p\})$ does not intersect L, otherwise L would intersect more than finitely many points of $VN_k(p,S)$. Since L separates parts of $VN_k(p,S)$ there are at least two regions $V_1(p,Q\cup\{p\})$ and $V_1(p,Q'\cup\{p\})$ separated by L and $Q \neq Q'$. Since $V_1(p,S) \subseteq V_1(p,Q\cup\{p\}) \cap V_1(p,Q'\cup\{p\})$, the region $V_1(p,S)$ is empty, which is a contradiction to (A3).

Next we show that there can be no holes in $VN_k(p,S)$. For the sake of contradiction suppose there is a hole F entirely surrounded by $VN_k(p,S)$. Then the boundary ∂F is a subset of $\partial VN_k(p,S)$. Therefore the edges on the boundary ∂F correspond to the bisectors $J(p,g)$, $g \in G \subset S$. If one of the bisectors $J(p,g')$ goes through the interior of F then consider $F \cap D(g',p)$, which is not empty, and so on until we have F' bounded by edges of the bisectors $J(p,g)$, $g \in G' \subseteq G$. The construction of F' implies that $F' \subseteq \bigcap_{g\in G'} D(g,p)$. Consider the farthest Voronoi diagram $\mathcal{V}_m(G'\cup\{p\})$, where $m = |G'|$. Set F' is a bounded face of the farthest Voronoi region $\bigcap_{g\in G'} D(g,p) = V_m(G, G\cup\{p\})$, which contradicts the fact that there are no bounded faces in farthest Voronoi diagrams [17, Lemma 7]. □

Alternatively we can prove the part of Lemma 6.1.1 about the path connectivity in the following way. The definition of $VN_k(p,S)$ implies that p is at most k-nearest for every point in $VN_k(p,S)$. Therefore

$$VN_k(p,S) = \bigcup_{p\in H, H\subset S, |H|=k} \overline{V_1(p,\{p\}\cup(S\setminus H))}.$$

Axiom (A1) implies that $V_1(p,\{p\}\cup(S\setminus H))$ is path connected. Thus $VN_k(p,S)$ is also path connected as the union of path connected sets.

Axiom (A5) implies that any two bisectors have finitely many intersection points. However, the following lemma states that if two bisectors have a site in common then they intersect at most twice.

Lemma 6.1.2. *Let $J(p,q)$ and $J(p,r)$ be a pair of bisectors, where $r \neq q$, then the bisectors intersect at most twice.*

Proof. Consider the order-1 Voronoi diagram $\mathcal{V}_1(\{p,q,r\})$. Axiom (A1) implies that there are exactly 3 faces in $\mathcal{V}_1(\{p,q,r\})$. Euler's formula implies that there are at most two Voronoi vertices in $\mathcal{V}_1(\{p,q,r\})$. Axiom (A3) implies that bisector $J(q,r)$ must pass through every intersection point of bisectors $J(p,q)$, $J(p,r)$

and each intersection point corresponds to a Voronoi vertex of $\mathcal{V}_1(\{p,q,r\})$. Since the number of Voronoi vertices is at most two, bisectors $J(p,q)$ and $J(p,r)$ intersect at most twice. □

6.2 Randomized Divide and Conquer Algorithm

Despite the recent results on nearest neighbor Voronoi diagrams, no efficient algorithmic techniques have been available that are directly applicable to compute the order-k abstract Voronoi diagrams. The standard iterative approach [59], which is directly applicable, is efficient for small values of k only. There are efficient randomized algorithms for point sites in the Euclidean metric, most of them depend on specific geometric transformations, see Section 2.3. It is nontrivial to get rid of the geometric transformations and to extend them to the abstract version.

For nearest neighbor Voronoi diagrams Mehlhorn et al. [61] proposed a randomized incremental construction algorithm, which is based on the Clarkson-Shor incremental construction technique [31]. The extension of this approach to handle higher-order abstract Voronoi diagrams is stated as an open problem [61].

In this chapter, we develop a randomized divide and conquer algorithm to compute the order-k abstract Voronoi diagram in expected $O(kn^{1+\varepsilon})$ basic operations, based on Clarkson's random sampling technique and one additional axiom:

(A6) The number of vertical tangencies of a bisector is a constant.

We assume the existence of a sufficiently large closed curve Γ such that no two bisectors intersect outside Γ. Γ can be viewed as $J(\infty,p)$, for any $p \in S$. The usage of Γ is to deal with unbounded Voronoi edges and faces. Hereafter we consider only the part of the Voronoi diagram enclosed inside Γ, without explicit indication.

We consider the following basic operations:

1. For an arbitrary point x, determine if x is in $D(p,q)$, $J(p,q)$ or $D(q,p)$.
2. Given a point x on $J(p,q)$, determine the next vertical tangent point or the next intersection with $J(s,t)$ or a straight line along one direction of $J(p,q)$.
3. For two points x,y on $J(p,q)$, determine the in-front/behind relation along one direction of $J(p,q)$;

4. Compare two points x, y by x-coordinate, where x, y are the intersection points or the points of vertical tangency of the bisectors.

The first two operations allow to compute the Voronoi diagram for a constant number of sites in $O(1)$ time, the second one allows a vertical trapezoidal decomposition of a region, and the last one allows to link Voronoi vertices along one bisector. For point sites in any convex distance metric or the Karlsruhe metric, and for disjoint line segments or disjoint convex polygons of constant size in the L_p norms or under the Hausdorff metric, our algorithm achieves expected $O(kn^{1+\varepsilon})$ time, as all basic operations take $O(1)$ time.

In order to apply Clarkson's technique [30], we define a vertical decomposition of the order-k Voronoi diagram. Then, we prove that our vertical trapezoidal decomposition of $V_k(H,S)$ allows a divide and conquer algorithm and an expected time analysis. To compute the sub-diagram when the sub-instance is small enough we propose two sub-algorithms for order-k abstract Voronoi diagrams. For the first one, we combine Lee's [59] iterative method for point sites in the Euclidean metric and randomized incremental construction for first-order abstract Voronoi diagrams [55] to achieve expected $O(k^2 n \log n)$ operations. For the second one, we first prove that certain properties for points in the Euclidean metric also hold for the abstract version, and then adopt Har-Peled's random walk technique [48] to make an $O(n^2 2^{\alpha(n)} \log n)$-operation randomized algorithm, where $\alpha(\cdot)$ is the inverse of the Ackermann function.

Our algorithm displays the essence of Clarkson's randomized divide and conquer algorithm for Euclidean order-k Voronoi diagrams [30], but does not require many geometric constraints. Clarkson's method depends on many properties of planes in 3D space, which correspond to circles in the plane, and thus, would be more restricted in geometric sites and distance metrics. Instead, our algorithm uses the viewpoint of Voronoi diagrams to define all the sub-structures and conflict relations, and thus, it relies on the properties of a bisector system that satisfies the 6 axioms, rather than geometric sites and distance measures.

6.2.1 Refined Diagram

We first define a refined version of the order-k Voronoi diagram and then partition it into vertical decomposition.

Definition 11. *The refined order-k Voronoi diagram $\mathbb{V}_k(S)$ of S is derived by superimposing $\mathcal{V}_k(S)$ and $\mathcal{V}_{k+1}(S)$. It is defined as:*

$$\mathbb{V}_k(S) = \mathcal{V}_k(S) \cup \bigcup_{H \subset S, |H|=k} \mathcal{V}_1(S \setminus H) \cap V_k(H,S).$$

A region $\mathbb{V}_k(p,H,S)$ of $\mathbb{V}_k(S)$ is associated with a site $p \in S$, which is called the dominator, and a k-element subset $H \subset S$. For any point $x \in \mathbb{V}_k(p,H,S)$, H is the set of k nearest sites to x and p is the $(k+1)$-nearest site to x.

This definition is also equivalent to

$$\mathbb{V}_k(S) = \mathcal{V}_k(S) \cup \mathcal{V}_{k+1}(S). \tag{6.3}$$

Subsequently the following lemma implies time complexity bounds for the construction of $\mathbb{V}_k(S)$ from $\mathcal{V}_k(S)$.

Lemma 6.2.1. *$\mathcal{V}_{j+1}(S)$ can be computed from $\mathcal{V}_j(S)$ in expected $O(j(n-j)\log n)$ operations.*

Proof. Consider a face F of $\mathcal{V}_j(S)$. Suppose F is a face of the region $V_j(H,S)$. Consider ℓ regions $V_j(H_i,S)$ adjacent to the face F, $i = 1,\ldots,\ell$. For every edge e on the boundary ∂F we can determine in $O(1)$ time the site that belongs to H_i but does not belong to H in the following way: suppose the edge e corresponds to the bisector $J(p,q)$ then if $F \subseteq D(p,q)$ then take q otherwise take p. The set $Q = \bigcup_{i=1,\ldots,\ell} H_i \setminus H$ can be computed in $O(|\partial F|)$ operations, where $|\partial F|$ denotes the number of edges along the boundary of F.

We want to show that $\mathcal{V}_1(Q) \cap F = \mathcal{V}_{j+1}(S) \cap F$, which is also equal to $\mathcal{V}_1(S \setminus H) \cap F$. Consider $\mathcal{V}_1(S \setminus H) \cap F$. Take an arbitrary point $x \in F$ and suppose $x \in V_1(s, S \setminus H)$. For the sake of a contradiction assume $s \notin Q$. This means that for any $q \in Q$, $s <_x q$. Therefore q is the $(j+1)$-closest site to the point x among the sites in S, i.e. $x \in V_{j+1}(H \cup \{s\}, S)$. Let F' be the face of $V_{j+1}(H \cup \{s\}, S)$ that contains x. Since $s \notin Q$, F' does not intersect ∂F. Thus F' is completely enclosed in F and does not intersect any edge of $\mathcal{V}_j(S)$, i.e. $F' \cap \mathcal{V}_j(S)$ is empty. The portion of $\mathcal{V}_j(S)$ enclosed in F' is exactly the farthest Voronoi diagram $\mathcal{V}_j(H \cup \{s\})$ enclosed in F' [17, Lemma 12], which is not empty [17, Lemma 13], a contradiction. Hence $s \in Q$, and since x is taken arbitrarily in F, $\mathcal{V}_1(S \setminus H) \cap F = \mathcal{V}_1(Q) \cap F = \mathcal{V}_{j+1}(S) \cap F$.

Therefore, we can construct $\mathcal{V}_{j+1}(S)$ by constructing $\mathcal{V}_1(Q) \cap F$ for each of the faces in $\mathcal{V}_j(S)$. The algorithm in [55] computes $\mathcal{V}_{j+1}(S) \cap F$ in expected $O(|Q|\log|Q|) = O(|\partial F|\log|\partial F|)$ operations. Therefore, the expected number of operations for computing $\mathcal{V}_{j+1}(S)$ from $\mathcal{V}_j(S)$ is bounded by

$$\sum_{F \text{ face of } \mathcal{V}_j(S)} O(|\partial F|\log|\partial F|) = O(|\mathcal{V}_j(S)|\log|\mathcal{V}_j(S)|),$$

where $|\mathcal{V}_j(S)|$ is the structural complexity of $\mathcal{V}_j(S)$. Since the structural complexity of $\mathcal{V}_j(S)$ is $O(j(n-j))$ [17], the $O((j(n-j)\log n)$ expected running time follows. □

Definition 12. *The* vertical decomposition *of $\mathbb{V}_k(S)$, denoted by $\mathbb{V}_k^{\triangle}(S)$, is the subdivision of the plane into (pseudo-)trapezoids obtained by shooting vertical rays up and down from each vertex in $\mathbb{V}_k(S)$ and each vertical tangent point of each edge in $\mathbb{V}_k(S)$, until the intersection with an edge or all the way to infinity.*

Lemma 6.2.2. *Let A be a subset of the arrangement of curves $\mathcal{J}$. The vertical trapezoidal decomposition $A^{\triangle}$ can be constructed from A in $O(m \log m)$ operations, where m is the complexity of A.*

Proof. Because each bisector has a constant number of points of vertical tangency, we can take $O(m)$ operations to compute all points of vertical tangency in A. Thus in $O(m \log(m))$ operations we can create a list sorted according to the x-coordinate $v_1, \ldots, v_{m'}$, $m' \in O(m)$ of all points of interest, intersections between bisectors and points of vertical tangency, of $\mathcal{V}(S)$. Further let v_1 be the leftmost point among $v_1, \ldots, v_{m'}$. Now we sweep the plane with a sweepline L from left to right and keep the edges intersected by the sweepline, which for each time step stores the edges intersected by L sorted by their y-coordinate.

Each time the sweepline hits a point v_i we have an event where a vertical line segment from v_i to the next edge above and below v_i is inserted into $A^{\triangle}$. Further the edges incident to v_i to the left are deleted from the sweepline structure and the edges to the right are inserted into the sweepline structure. Using a balanced binary tree with connected leaves to store the order of the edges along the sweepline, insertion and deletion of edges in the takes $O(\log m)$ operations, the insertion of vertical line segments in $A^{\triangle}$ takes constant time. Thus the whole algorithm takes $O(m \log m)$ operations. □

Lemma 6.2.3. *$\mathbb{V}_k^{\triangle}(S)$ can be constructed from $\mathcal{V}_k(S)$ in expected $O(k(n-k)\log n)$ operations.*

Proof. By Lemma 6.2.1, $\mathbb{V}_k(S)$ can be constructed from $\mathcal{V}_k(S)$ in expected $O(k(n-k)\log n)$ operations. Eq. 6.3 implies that the structural complexity of $\mathbb{V}_k(S)$ and $\mathcal{V}_k(S)$ are asymptotically the same, i.e. $O(k(n-k))$. Therefore by Lemma 6.2.2 the expected number of operations to compute $\mathbb{V}_k^{\triangle}(S)$ from $\mathbb{V}_k(S)$ is

$$O(k(n-k)\log(k(n-k))) = O(k(n-k)\log n).$$

□

A trapezoid $\triangle$ of $\mathbb{V}_k^{\triangle}(S)$ in $\mathbb{V}_k(p,H,S)$ is defined by the dominator p and 1-4 other sites. Vertical boundaries of the trapezoid may be defined either by an intersection point or by a point of vertical tangency. Moreover, one of the vertical

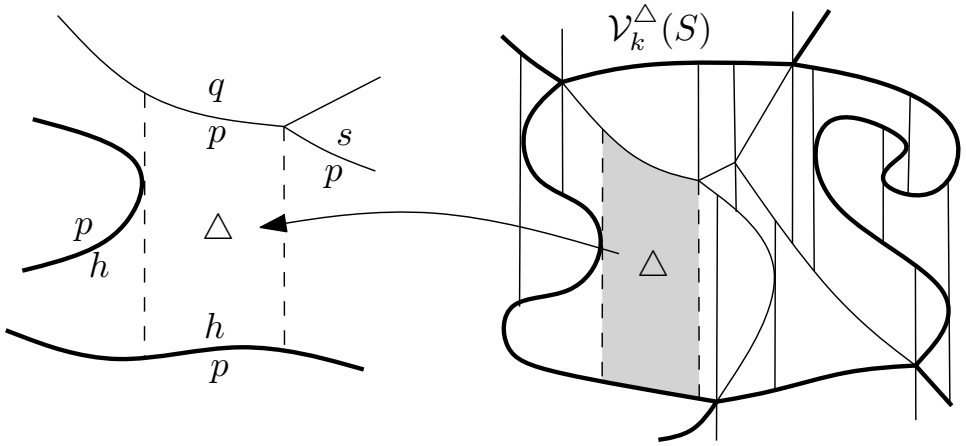

Figure 6.1. Trapezoid $\triangle$ of $\mathbb{V}_k^{\triangle}(S)$. $\mathcal{V}_k(S)$ is depicted in shaded.

boundaries may be degenerate. Let $d(\triangle)$ be the dominator of the trapezoid and $B(\triangle)$ be the set of sites that together with the dominator define the boundaries of the trapezoid $\triangle$. Then $1 \leq |B(\triangle)| \leq 4$ and for any point $x \in \triangle$, $H \setminus B(\triangle)$ are the $k - |H \cap B(\triangle)|$ nearest sites to x.

In Figure 6.1, the top and bottom edges of $\triangle$ are defined by $J(p,q)$ and $J(p,h)$, respectively, and the left and right edges are defined by a vertical tangent point of $J(p,h)$ and an intersection between $J(p,q)$ and $J(p,s)$, respectively. In other words, $B(\triangle) = \{q,h,s\}$ and $d(\triangle) = p$.

In order to apply Clarkson's abstract framework [30] to the abstract Voronoi diagrams framework we define the notion of conflicts between the trapezoids and the sites in such a way that it efficiently "brackets" the space. The property of the conflict relations to "bracket" the space is the essential part of the divide and conquer algorithm.

Definition 13. *For a trapezoid $\triangle$ of $\mathbb{V}_k^{\triangle}(S)$, a site $s \notin B(\triangle)$* strongly conflicts *with $\triangle$, if $\overline{\triangle} \subset D(s, d(\triangle))$. A site $s \notin B(\triangle)$* weakly conflicts *with $\triangle$, if $\overline{\triangle} \cap D(s, d(\triangle)) \neq \emptyset$. The set of sites $X \subseteq S$ that* strongly*, resp.* weakly conflict *with $\triangle$ is denoted by $X \wedge_s \triangle$, resp. $X \wedge_w \triangle$.*

In general, the set of *strong conflicts* is different from the set of *weak conflicts*, and $X \wedge_s \triangle \subseteq X \wedge_w \triangle$. In Figure 6.2, set $S = \{p_1, \ldots, p_7, s_1, \ldots, s_4\}$ is the set of line segments in Euclidean space. $R = \{p_1, \ldots, p_7\}$ is the subset of S and $\triangle$ is the trapezoid of $\mathcal{V}_3^{\triangle}(R)$ in $\mathbb{V}_3(p_1, \{p_2, p_3, p_4\}, R)$. The dominator $d(\triangle)$ of the trapezoid $\triangle$ is p_1. The set of the sites $B(\triangle)$ that define the boundaries of the trapezoid $\triangle$ is $\{p_2, p_3, p_5, p_6\}$. Since the sites p_2, p_3, p_5, p_6 define the boundary of the trapezoid they cannot conflict with the trapezoid. However, the site p_4

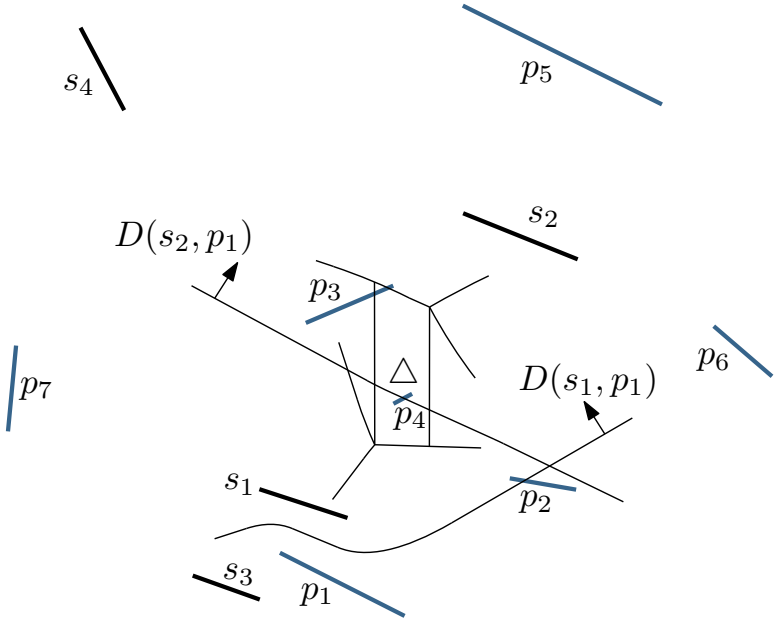

Figure 6.2. Trapezoid $\triangle \in \mathbb{V}_3(p_1, \{p_2, p_3, p_4\})$, where $p_1, \ldots, p_7$ are line segments.

strongly conflicts with $\triangle$, since $\overline{\triangle} \subset D(p_4, p_1)$. Sites that do not belong in R can also conflict with the trapezoid. Here, site s_1 strongly conflicts with $\triangle$, since $\overline{\triangle} \subset D(s_1, p_1)$. However, site s_2 weakly conflicts with $\triangle$, because the dominance region $D(s_2, p_1)$ does not enclose $\overline{\triangle}$, but only intersects $\overline{\triangle}$. Thus, $S \wedge_s \triangle = \{p_4, s_1\}$, $S \wedge_w \triangle = \{p_4, s_1, s_2\}$. In Lemmas 6.2.4, 6.2.5 we use *weak* and *strong conflicts* for the upper and lower bounds, respectively.

Lemma 6.2.4. *Let R be a subset of S and β be a positive integer. Then for any trapezoid $\triangle$ of $\mathbb{V}^{\triangle}_{\beta}(R)$,*

1. $|R \wedge_s \triangle| \geq \beta - 4$,
2. $|R \wedge_w \triangle| \leq \beta$.

Proof. Let $\triangle$ be a trapezoid enclosed in the region $V_\beta(H, R)$. We want to prove that $H \setminus B(\triangle) \subseteq R \wedge_s \triangle$ and $R \wedge_w \triangle \subseteq H$, which will immediately imply the statement of the lemma, since $|B(\triangle)| \leq 4$ and $|H| = \beta$.

For each point x in the trapezoid $\triangle$, H is the set of β nearest sites and $d(\triangle)$ is the $(\beta+1)$-nearest site. Therefore for each site $p \in H \setminus B(\triangle)$, p is closer to x than $d(\triangle)$, thus $\overline{\triangle} \subset D(p, d(\triangle))$. We also include the boundary of the trapezoid since we have excluded sites in $B(\triangle)$ that define the boundary. Therefore sites in $H \setminus B(\triangle)$ are in the strong conflict with the trapezoid $\triangle$, i.e. $H \setminus B(\triangle) \subseteq R \wedge_s \triangle$.

Let p be the site in $R \wedge_w \triangle$. The definition of the weak conflict implies that there is a point x in $\overline{\triangle}$ such that $p <_x d(\triangle)$. If x belongs in $\triangle$ then p is one of the β nearest sites to x, i.e. $p \in H$. If x is in $\partial\triangle$ then p can not be a site that defines the boundary, because the dominance region $D(p, d(\triangle))$ does not include the bisector $J(p, d(\triangle))$ that corresponds to the boundary. Thus p is one of the β nearest sites to x, i.e. $p \in H$. Therefore $R \wedge_w \triangle \subseteq H$. □

Lemma 6.2.4 and [30, Corollaries 4.3 and 4.4] imply the following.

Lemma 6.2.5. *Let R be an r-element random sample of S. Then with probability at least 1/2, as $r \to \infty$, for any trapezoid $\triangle$ of $\mathbb{V}^{\triangle}_{\beta}(R)$,*

1. $|S \wedge_s \triangle| \geq |S|/(r-5)$,
2. $|S \wedge_w \triangle| \leq \alpha|S|$,

where $\beta = O(\log r / \log\log r)$ and $\alpha = O(\log r / r)$.

Proof. The proof follows [30, Lemma 5.4]. Let $\mathcal{F}_R$ be the family of trapezoids defined by at most 5 sites in R. First we want to prove that $|\mathcal{F}_R| = O(r^5)$. Each trapezoid is defined by the dominator and 1-4 other sites. Let p be the dominator of the trapezoid $\triangle$, then the boundary of the trapezoid is defined by the bisectors $J(p,q)$, where $q \in B(\triangle)$. Lemma 6.1.2 implies that the bisectors that define the boundary intersect $O(1)$ number of times, i.e. there are $O(1)$ ways to define a trapezoid given a dominator and 1-4 other sites. Thus there are $O(1)$ ways to define a trapezoid given 2-5 sites. Therefore $|\mathcal{F}_R| = O(r^5)$.

Consider the following probability:

$$Prob\{\exists\triangle \in \mathbb{V}^{\triangle}_{\beta}(R)\ |S \wedge_s \triangle| < n/(r-5)\}.$$

Lemma 6.2.4 implies that for every trapezoid $\triangle$ in $\mathbb{V}^{\triangle}_{\beta}(R)$ the number of *strong conflicts* with R is at least $\beta - 4$. Therefore, we can apply [30, Corollary 4.3] to derive

$$Prob\{\exists\triangle \in \mathcal{F}_R\ |R \wedge_s \triangle| \geq \beta - 4 \text{ and } |S \wedge_s \triangle| < n/(r-5)\} \leq O(r^5)(e/(\beta-4))^{\beta-4}.$$

Similarly consider the following probability:

$$Prob\{\exists\triangle \in \mathbb{V}^{\triangle}_{\beta}(R)\ |S \wedge_w \triangle| > \alpha n\}.$$

Lemma 6.2.4 implies that every trapezoid $\triangle$ in $\mathbb{V}^{\triangle}_{\beta}(R)$ is in *weak conflict* with at most β sites in R. [30, Corollary 4.4] implies

$$Prob\{\exists\triangle \in \mathcal{F}_R\ |R \wedge_w \triangle| \leq \beta \text{ and } |S \wedge_w \triangle| > \alpha n\} \leq O(r^5)e^{-\alpha r}(e\alpha r/\beta)^{\beta}.$$

For suitable $\alpha = O(\log r / r)$ and $\beta = O(\log r / \log\log r)$ these two probabilities are less than $1/4$ each. Therefore the claim of the lemma follows. □

Lemma 6.2.6. *Let R be a subset of S such that for any trapezoid $\triangle$ of $\mathbb{V}_\beta^\triangle(R)$, $|S \wedge_s \triangle| > k$. Let v be a Voronoi vertex of $\mathcal{V}_k(S)$. Then there exists a trapezoid $\triangle \in \mathbb{V}_\beta^\triangle(R)$ such that v is also a Voronoi vertex of $\mathcal{V}_k(S \wedge_w \triangle)$.*

Proof. Let v be a Voronoi vertex incident to three Voronoi regions $V_k(H_1, S)$, $V_k(H_2, S)$ and $V_k(H_3, S)$. Consider the trapezoidal decomposition $\mathbb{V}_\beta^\triangle(R)$. There is a trapezoid $\triangle$ of $\mathbb{V}_\beta^\triangle(R)$ such that $\overline{\triangle}$ contains v. We want to prove that $H_1 \cup H_2 \cup H_3 \subseteq S \wedge_w \triangle$.

Let H be $H_1 \cup H_2 \cup H_3$ and $t = |H|$. By Definition 8 and Definition 9, v is either a *new* or an *old* vertex, therefore t is either $k+1$ or $k+2$. Consider the sequence π_v^S, in the sequence the sites have the following relations: $s_1 \leq_v \ldots \leq_v s_{t-3} <_v s_{t-2} =_v s_{t-1} =_v s_t <_v s_{t+1} \ldots$, where $H = \{s_1, \ldots, s_t\}$.

Definition 13, implies that for each site $p \in S \wedge_s \triangle$, $p <_v d(\triangle)$. Therefore in the sequence π_v^S the sites in $S \wedge_s \triangle$ are to the left of the site $d(\triangle)$. Denote as d the index of the dominator in the sequence, i.e. $s_d = d(\triangle)$. Then $d > k+1$ since all the sites in $S \wedge_s \triangle$ are to the left of the dominator and $|S \wedge_s \triangle| > k$.

Take a site s_i, where $i < d$. Site s_i has index less than d, therefore $s_i \leq_v d(\triangle)$, or $v \in D(s_i, d(\triangle))$. Since $v \in \overline{\triangle}$, $v \in \overline{\triangle} \cap D(s_i, d(\triangle))$ and therefore s_i is in weak conflict with the trapezoid $\triangle$. Therefore for any site s_i, for $i < d$, $s_i \in S \wedge_w \triangle$.

Consider now the sites $s_1, \ldots, s_t$. Since t can be equal to $k+1$ or $k+2$, $t \leq k+1 < d$. Therefore $s_1, \ldots, s_t \in S \wedge_w \triangle$. Recall that $H = H_1 \cup H_2 \cup H_3 = \{s_1, \ldots, s_t\}$ then $H \subseteq S \wedge_w \triangle$.

Consider now two order-k Voronoi diagrams: $\mathcal{V}_k(S)$ and $\mathcal{V}_k(S \wedge_w \triangle)$. Since the order-$k$ Voronoi vertex v is defined by the sites in H_1, H_2 and H_3 it is also present in the second diagram, because $H_1 \cup H_2 \cup H_3 \subseteq S \wedge_w \triangle$. □

6.2.2 Computing the Voronoi vertices

Lemma 6.2.6 indicates that if for any $\triangle$ trapezoid of $\mathbb{V}_\beta^\triangle(R)$, $|S \wedge_s \triangle| > k$, then computing the Voronoi vertices of $\mathcal{V}_k(S)$ can be transformed into computing the Voronoi vertices of $\mathcal{V}_k(S \wedge_w \triangle)$ for each $\triangle$ trapezoid of $\mathbb{V}_\beta^\triangle(R)$.

Lemma 6.2.5 states that on average it takes two trials to generate the sample R such that $|S \wedge_s \triangle| \geq |S|/(r-5)$, where the size r of the random sample R is any sufficiently large constant. Therefore, if $|S|/(r-5) > k$, then we need two trials on average to generate a random sample that satisfies the conditions of Lemma 6.2.6. The condition $|S \wedge_w \triangle| \leq \alpha|S|$ in Lemma 6.2.5 bounds the depth of the recursion. Following Clarkson [30], the algorithm to compute the Voronoi vertices of $\mathcal{V}_k(S)$ is summarized as follows:

- If $|S|/(r-5) \leq k$, compute the Voronoi vertices of $\mathcal{V}_k(S)$ by the algorithm in Section 6.4.
- Otherwise ($|S|/(r-5) > k$)
  1. Choose $R \subset S$ of size r until R satisfies the conditions of Lemma 6.2.5
     (a) Construct $\mathcal{V}_\beta(R)$ by the algorithm in Section 6.3 and Compute $\mathbb{V}^{\triangle}_{\beta}(R)$ from $\mathcal{V}_\beta(R)$ (Lemma 6.2.3).
     (b) Check each trapezoid in $\mathbb{V}^{\triangle}_{\beta}(R)$ to satisfy the conditions of Lemma 6.2.5.
  2. For each trapezoid $\triangle \in \mathbb{V}^{\triangle}_{\beta}(R)$
     (a) Recursively compute the Voronoi vertices $\mathcal{V}_k(S \wedge_w \triangle)$.
     (b) Select vertices of $\mathcal{V}_k(S \wedge_w \triangle)$ that are vertices of $V_k(S)$.

Here we address the question of choosing the constant r. Lemma 6.2.5 states that as r tends to infinity with probability at least $1/2$ the numbers of weak and strong conflicts are bounded by some values. Therefore we can find a large value of r such that the probability is at least $1/2$. This value can be computed by solving $O(r^5)(e/(\beta-4))^{\beta-4} < 1/4$ and $O(r^5)e^{-\alpha r}(e\alpha r/\beta)^{\beta} < 1/4$. In the time analysis we also sometimes consider r to be tending to infinity. This is done in order to simplify the analysis and allow some terms to dominate over another. The larger the value of r is taken, the greater is the chance to successfully generate the sample R. The larger the value of r, the smaller is the value of ε and as the result, the smaller is the asymptotic running time of the algorithm. However, the larger the value of r, the larger is the constant hidden in the O-notation of the running time, see the next section.

6.2.3 Analysis

Lemma 6.2.7. *$\mathcal{V}_k(S)$ can be computed from its Voronoi vertices in $O(k(n-k)\log n)$ operations.*

Proof. For points-sites, a vertex is uniquely defined by three sites [59]. Also for point-sites two vertices are adjacent iff their corresponding triples of sites have two sites in common. However, in the abstract setting, three sites may define one or two vertices and the adjacency property does not hold. Therefore, we cannot solve this problem by just using radix sort as it was done for point-sites [30].

Here, in the abstract setting, we use radix sort to extract for each bisector all Voronoi vertices that lie on it, in total $O(|V|)$ operations, where V is the set

of vertices in $\mathcal{V}_k(S)$. We also assume the existence of a sufficiently large closed curve Γ such that no two bisectors intersect outside Γ. Γ can be viewed as $J(\infty, p)$, for any $p \in S$.

Consider a set of $m_J > 0$ Voronoi vertices that belong to bisector J (including the artificial Voronoi vertices formed by the intersection between $\mathcal{V}_k(S)$ and Γ). m_J must be even; otherwise, at least one Voronoi vertex has no Voronoi edge. We can sort the m_J Voronoi vertices along one direction of J as $v_1, v_2, \ldots, v_{m_J}$ in $O(m_J \log m_J)$ operations, and then link v_{2i-1} with v_{2i} for $1 \leq i \leq m_J/2$ as Voronoi edges in $O(m_J)$ operations. Therefore, we can compute all the Voronoi edges on J in $O(m_J \log m_J)$ operations. Since $|V|$ is $O(k(n-k))$, the total number of operations is

$$O(|V|) + \sum_{J \in \mathcal{J}, m_J > 0} O(m_J \log m_J) = O\left(|V| \log |V|\right) = O\left(k(n-k) \log n\right).$$

□

Theorem 6.2.8. *$\mathcal{V}_k(S)$ can be computed in expected $O(kn^{1+\varepsilon})$ operations, where $\varepsilon > 0$, and the constant factor of the asymptotic bound depends on ε.*

Proof. The proof follows [30, Lemma 6.4]. Recall that r is a sufficiently large constant, $\alpha = O(\log r / r)$ and $\beta = O(\log r / \log \log r)$. There are two cases:

1. If $|S|/(r-5) \leq k$, then we use the algorithm from Section 6.4 to compute the vertices of the order-k Voronoi diagram in expected $O(n^2 2^{\alpha(n)} \log n)$ operations. Since $n = |S| \leq k(r-5)$, it is less than $O(r^2 k^2 \log^2 r \log^2 k)$;

2. If $|S|/(r-5) > k$ then the algorithm proceeds as follows:

   (a) Choose a random sample that satisfies the conditions of Lemma 6.2.5. Do the check by constructing $\mathcal{V}_\beta(R)$ and computing $\mathbb{V}^{\triangle}_{\beta}(R)$ from $\mathcal{V}_\beta(R)$. The construction of $\mathcal{V}_\beta(R)$ takes expected $O(r\beta^2 \log r)$ operations (see Section 6.3), and computing $\mathbb{V}^{\triangle}_{\beta}(R)$ takes additional expected $O(\beta(r-\beta)\log r)$ operations.

   The number of the trapezoids in $\mathbb{V}^{\triangle}_{\beta}(R)$ is $O(r\beta)$, and the number of operations required to check the sample is $O(nr\beta) \subset O(nr \log r)$. This dominates both $O(r\beta^2 \log r)$ and $O(\beta(r-\beta)\log r)$.

   (b) For each trapezoid in $\mathbb{V}^{\triangle}_{\beta}(R)$ compute the order-k vertices using recursion. The number of recursive calls is the number of trapezoids in $\mathbb{V}^{\triangle}_{\beta}(R)$ which is $O(r\beta) \subset O(r \log r)$. Lemma 6.2.5 implies that each recursive call inputs $O(\alpha n)$ sites. The structural complexity

bound of the higher-order Voronoi diagrams implies that each recursive call outputs $O(\alpha nk)$ vertices. Therefore the total number of the vertices received from the output of the recursion is $O(\alpha nkr \log r)$ which is $O(nk \log^2 r)$. Thus the time required to validate the vertices is $O(nk \log^2 r)$.

Therefore, the expected number $t(n)$ of operations for computing the Vononoi vertices of $\mathcal{V}_k(S)$ is

$$t(n) \leq O\left(rk \log r \log k\right)^2, \qquad n \leq k(r-5)$$
$$t(n) \leq O\left(nr \log r\right) + O\left(nk \log^2 r\right) + O(r \log r)t\left(O(n \log r/r)\right), \qquad n > k(r-5),$$

and the depth of the recursion is $D = O(\log(n/k)/\log(r/\log r))$. Where the asymptotic bounds are given as $r \to \infty$. Since $O(nr \log r) + O(nk \log^2 r) = O(nkr \log r)$ we can derive the following upper bound for $t(n)$:

$$t(n) = (k \log k)^2 (r \log r)^2 (r \log r)^D + nkO(r \log r)(\log r)^{2(D+1)}.$$

If we take $\varepsilon = O(\log \log r / \log r)$ we receive

$$t(n) = O\left(n^{1+\varepsilon} k^{1-\varepsilon} \log^2 k + n^{1+\varepsilon} k^{1-\varepsilon}\right) = O\left(kn^{1+\varepsilon}\right),$$

as $n \to \infty$, where the constant factor depends on ε, see [30, Lemma 6.4]. Since $\mathcal{V}_k(S)$ can be constructed from the Voronoi vertices of $\mathcal{V}_k(S)$ in expected $O(k(n-k)\log n)$ operations (Lemma 6.2.3), $\mathcal{V}_k(S)$ can be constructed in expected $O(kn^{1+\varepsilon})$ operations. □

6.3 Iterative Construction

The order-k abstract Voronoi diagram can be computed iteratively similarly to point sites in the Euclidean metric [59]. First we construct the nearest neighbor Voronoi diagram $\mathcal{V}_1(S)$. Then we iteratively construct $\mathcal{V}_{j+1}(S)$ from $\mathcal{V}_j(S)$ until we receive the order-k Voronoi diagram.

Theorem 6.3.1. *$\mathcal{V}_k(S)$ can be computed in expected $O(k^2 n \log n)$ operations.*

Proof. Since the algorithm in [55] can compute $\mathcal{V}_1(S)$ in expected $O(n \log n)$ operations, by Lemma 6.2.1, the expected number of operations is bounded by

$$O(n \log n) + \sum_{j=1}^{k-1} O(j(n-j)\log n) = O(k^2 n \log n)$$

□

6.4 Random Walk Method

We construct $\mathcal{V}_k(S)$ by computing $\partial VN_k(p,S)$ for every $p \in S$, i.e., all the Voronoi edges of $\mathcal{V}_k(S)$ belonging to $J(p,q)$, see Eq. 6.2. Chazelle and Edelsbrunner [28] computed $\partial VN_k(p,S)$ based on dynamic convex hulls and the fact that $VN_k(p,S)$ is simply connected. However, dynamic convex hulls are not applicable in the abstract setting. Since $VN_k(p,S)$ is simply connected, we can adopt Har-Peled's [48] random walk algorithm to compute $\partial VN_k(S)$.

Given an arrangement of n x-monotone curves, each pair of which intersects at most t times, and a curve γ, Har-Peled's [48] random walk algorithm computes the zone of γ, i.e. the faces of the arrangement intersected by γ. Consider the vertical decomposition of the arrangement. The algorithm traverses the curve γ while maintaining the trapezoids of the faces that were already encountered.

$\partial VN_k(p,S)$ is a substructure of the arrangement of $n-1$ bisectors $\mathcal{J}(p) = \{J(p,q) \mid q \in S \setminus \{p\}\}$, where the bisectors in $\mathcal{J}(p)$ are not x-monotone, but they have constant number of vertical tangency points. Therefore, the structural complexities of the arrangement and its vertical decomposition are of the same asymptotic magnitude. We view $\partial VN_k(p,S)$ as γ and compute $\partial VN_k(p,S)$ using the same technique. We construct $\partial VN_k(p,S)$ in the following way:

1. For each connected component of $\partial VN_k(p,S)$ compute a starting point.
2. For each starting point, traverse the corresponding part of $\partial VN_k(p,S)$.

As we walk we can determine the next direction in $O(1)$ time (see Lemma 6.4.2).

Lemma 6.4.1. *The starting points of $\partial VN_k(p,S)$ for each of its connected components can be computed in total $O(n \log n)$ expected time.*

Proof. Consider the function $\sigma_p(x)$, $x \in \mathbb{R}^2$:

$$\sigma_p(x) = \left|\{s \in S \mid s \neq p, x \in D(s,p)\}\right|.$$

For a continuous walk in the plane, $\sigma_p(x)$ changes by 1 when the walk hits or leaves the curve from $\mathcal{J}(p)$. For each edge e of $\partial VN_k(p,S)$ and each point x in e, p is the k-nearest site of x in S such that $\sigma_p(x) = k-1$. Moreover, by Definition 9, for each vertex v of $\partial VN_k(p)$, $\sigma_v(p)$ is $k-1$ or $k-2$, depending on whether v is a new or an old order-k Voronoi vertex.

Recall that we assume the existence of a sufficiently large closed curve Γ such that no two bisectors intersect outside Γ. We first collect starting points of

unbounded edges of $\partial VN_k(p,S)$, and if there does not exist one, $VN_k(p,S)$ is bounded.

Suppose $VN_k(p,S)$ is unbounded. There are $2n-2$ intersections between the extremely large closed curve Γ and $\mathcal{J}(p)$, and it takes $O(n\log n)$ operations to sort them along Γ. Let $v_1, v_2, \ldots, v_{2n-2}$ be the intersections of Γ and $\mathcal{J}(p)$ sorted along the Γ. We can compute $\sigma_p(v_1)$ in $O(n)$ operations by checking whether $v_1 \in D(p,q)$ or $v_1 \in D(q,p)$ for each $q \in S \setminus \{p\}$. Then we traverse Γ and every time we encounter the new intersection point we either increase or decrease σ_p by 1, depending on the orientation test. The traversal takes additional $O(n)$ operations and it computes $\sigma_p(v_i)$ for $2 \leq i \leq 2n-2$. The starting points are those v_i that have $\sigma_p(v_i) = k-1$. If during the traversal we did not encounter any of the starting points then $VN_k(p,S)$ is bounded and we proceed with the following.

Suppose $\partial VN_k(p,S)$ is bounded, i.e., for $1 \leq i \leq 2n-2$, $\sigma_p(v_i) \neq k-1$. We compute $V_1(\{p\},S)$, which takes expected $O(n\log n)$ operations by the algorithm in [55]. We take any edge e on the boundary of $V_1(\{p\},S)$. Assume that e belongs to $J(p,q)$. It is clear that for each point x in e, $\sigma_p(x) = 0$. We compute the intersection points between $J(p,q)$ and the rest of curves in $\mathcal{J}(p)$. Since bisectors in $\mathcal{J}(p)$ intersect pairwise at most twice, the total number of intersections is at most $2n-4$. We sort the intersection points along $J(p,q)$ in $O(n\log n)$ time. We pick a point x in e, and traverse $J(p,q)$ from x along one direction. Since $\partial VN_k(p,S)$ is bounded and encloses $V_1(\{p\},S)$, the traversal will find the intersection point v such that $\sigma_p(v)$ is $k-1$ or $k-2$, which is the starting point. Therefore the starting points can be computed in total $O(n\log n)$ expected time. □

Lemma 6.4.2. *During the traversal of $\partial VN_k(p,S)$ each time the algorithm encounters a new intersection point of the bisectors, the next traversal direction can be determined in $O(1)$ time.*

Proof. Let $e = (u,v)$ be an edge of $\partial VN_k(p,S)$. Suppose e belongs to $J(p,q)$ and v is an intersection point between $J(p,q)$ and $J(p,t)$. Let $e' = (v,w)$ be the next edge to be traversed, i.e. e' belongs to $\partial VN_k(p,S)$. Since e' corresponds to the bisector $J(p,t)$ this leaves us with two directions to choose.

By Definition 10, for each point $x \in e$, p and q are both the k-nearest sites of x, and for each point $y \in e'$, p and t are both the k-nearest site of y. Therefore, if $e \in D(p,t)$ then $e' \in D(p,q)$. Otherwise, $e' \in D(q,p)$. Therefore we can test in $O(1)$ time both directions and choose the one belonging to $\partial VN_k(p,S)$. □

Following [48], the expected number of operations required to compute the boundary of the k-neighborhood by the random walk is $O(\lambda_{t+2}(n+m)\log n)$,

where t is the maximum number of intersections between two bisectors, and m is the complexity of $\partial VN_k(p,S)$. Lemma 6.1.2 implies that a pair of bisectors in $\mathcal{J}(p)$ intersects at most twice, therefore $t = 2$.

The main difference between computing the zone in the original version of the algorithm [48] and computing $\partial VN_k(p,S)$ is that the latter is additionally augmented by the vertical rays from the points of vertical tangency. However, since by the axiom (A6) each bisector allows only a constant number of points of vertical tangency, the expected number of operations increases only by a constant factor.

Theorem 6.4.3. *$\mathcal{V}_k(S)$ can be computed in $O(n^2 2^{\alpha(n)} \log n)$ expected operations.*

Proof. Consider the site s_i and its k-neighborhood $\partial VN_k(s_i,S)$. The boundary of the k-neighborhood can be computed in expected $O(\lambda_4(n+m_i)\log n)$ operations, where m_i is the complexity of the boundary. We compute $\mathcal{V}_k(S)$ by computing the boundary of the k-neighborhood for every site $s_i \in S$. Therefore, the expected overall number of operations is

$$\sum_{i=1}^{n} O(\lambda_4(n+m_i)\log n).$$

Eq. 6.2 implies that every edge of $\partial VN_k(s_i,S)$ is an edge of $\mathcal{V}_k(S)$, therefore $\sum_{i=1}^{n} m_i = O(k(n-k))$. We apply well-known bounds for $\lambda_4(\cdot)$ [84] and receive

$$\begin{aligned}\sum_{i=1}^{n} O\left(\left(n+m_i\right) 2^{\alpha(n+m_i)} \log n\right) &= \sum_{i=1}^{n} O\left(\left(n+m_i\right) 2^{\alpha(n)} \log n\right) \\ &= O\left(\left(n^2 + k(n-k)\right) 2^{\alpha(n)} \log n\right),\end{aligned}$$

which implies the claim. □

Theorems 6.3.1 and 6.4.3 bound the running time of the two algorithms that are used for the subroutines in the divide and conquer algorithm, see Theorem 6.2.8. In fact, we can use any $\widetilde{O}(n^2)$ algorithm that constructs the zone in the arrangement of Jordan curves instead of the random walk method, where in this dissertation $\widetilde{O}$-notation hides a polylogarithmic factor.

6.5 Summary

The abstract Voronoi diagram is a powerful unification of the properties of the concrete Voronoi diagrams in a single mathematical abstraction. It assumes sev-

eral axioms for the bisecting system and once a concrete Voronoi diagram satisfies the axioms, all the structural, combinatorial and algorithmic results become applicable. The higher-order abstract Voronoi diagram is an extension of the abstract Voronoi diagram to an arbitrary k, $1 \leq k < n$.

In this chapter we have discussed a unification of two abstract frameworks: Klein's abstract Voronoi diagrams [53] and the Clarkson-Shor technique [30, 31]. The unification has been achieved by defining the notion of a "conflict" for higher-order abstract Voronoi diagrams that allows application of some of the results of the Clarkson-Shor framework. This results in a randomized divide and conquer construction algorithm for higher-order abstract Voronoi diagrams, which has $O(kn^{1+\varepsilon})$ expected time complexity for any constant $\varepsilon > 0$. The algorithm uses the other two algorithms for solving small subproblems during the divide and conquer process. The randomized incremental algorithm has $O(k^2 n \log n)$ expected time complexity and is very useful for small orders. The random walk method has $O(n^2 2^{\alpha(n)} \log n)$ expected time complexity and is useful for large orders close to n. The random walk method is based on the idea of Chazelle and Edelsbrunner [28] and it uses Har-Peled's random walk technique [48].

The algorithms described in this chapter provide the improvement of the construction time for the following concrete Voronoi diagrams:

- Disjoint line segments in L_p metric, for $1 < p < \infty$. The previous construction algorithms provide $O(k^2 n \log n)$ construction time, see Chapter 3;

- Disjoint non-enclosed convex polygons of constant size. Previously the construction could be done by generalizing the iterative $O(k^2 n \log n)$-time construction algorithm [59, 73];

- Additively weighted points with non-enclosed circles. In Voronoi diagrams of additively weighted points the distance between a point p and a weighted point s is measured as $d_w(p,s) = d(p,s) - w(s)$, where $w(s)$ is the weight of the point s and $d(p,s)$ is the regular distance. The previous result due to Rosenberger allowed to do the deterministic construction in $O(k^2 n \log n)$ time [76];

- Power diagrams with non-enclosed circles. This restricted class of power diagrams can be constructed through the construction of the k-level of planes in 3D. Though the general planes in 3D are more complex than the power diagram with non-enclosed circles, these algorithms provide the best running so far which is $O(n \log n + nk^2)$ [23].

Chapter 7

Conclusions

Higher-order Voronoi diagrams of non-point sites are generally harder to investigate than higher-order Voronoi diagrams of point sites. A point is the simplest kind of geometric object, with zero dimensions, thus allowing the use of multiple techniques including but not limited to geometric transformations and dualities with other fundamental geometric objects. Consequently the major part of theoretical results in computational and combinatorial geometry is related to points. The higher-order Voronoi diagram had been studied only for the case of points. The most advanced algorithmic results for higher-order Voronoi diagrams of points utilize the simplicity of points and thus they are not easy to extend to other kinds of sites. Until recently, higher-order Voronoi diagrams of non-point sites had been studied only for two border cases: nearest neighbor and farthest Voronoi diagrams.

In this dissertation we have studied higher-order Voronoi diagrams of polygonal objects. The greater part of the investigation is dedicated to line segments, including intersecting line segments and line segments forming a planar straight-line graph. The results include investigation of structural properties, combinatorial complexity bounds and construction algorithms, most of which are applicable in the L_p metric, $1 \leq p \leq \infty$. The construction algorithms include the extensions of the iterative algorithm, a sweepline algorithm, and randomized algorithms for the higher-order abstract Voronoi diagrams. The sweepline algorithm allows on-the-fly computations, is simple to implement, useful for small orders and it can be used to construct all order-i Voronoi diagrams for $i \leq k$. The randomized algorithms for higher-order abstract Voronoi diagrams can be applied to any concrete Voronoi diagrams satisfying several basic assumptions: disjoint line segments, power diagrams, additively weighted points, disjoint convex polygons, etc. The randomized divide and conquer algorithm is the result

of the unification of two mathematical abstractions: abstract Voronoi diagrams and the Clarkson-Shor framework. It has the best running time among the algorithms investigated in this dissertation. The random walk algorithm is useful for values of $k = \Omega(n)$. The complete list of the contributions of this dissertation is given in Section 1.2.

The applications of higher-order Voronoi diagrams have been mostly limited to points, with the exception of [69]. Unfortunately, points are not always well-suited to represent real-life instances. We hope that the results of this dissertation will initiate the investigation of higher-order Voronoi diagrams of polygonal objects in computation geometry as well as in real-life applications.

7.1 Future Directions

The investigation of higher-order Voronoi diagrams of polygonal sites opens a number of research directions. Our results can be used as the foundation for investigations of other non-trivial cases, including generalizations of sites, spaces and distance functions. In the following section we discuss some of the possible future directions.

7.1.1 Simple Polygons

In this section we discuss possible future investigations of the higher-order Voronoi diagram of simple polygons. We also assume that the boundaries of the polygons are disjoint. Simple polygons have been studied for farthest Voronoi diagrams by Cheong et al. [29]. These results can be used in the analysis of the higher-order Voronoi diagrams of polygonal objects.

The case of convex polygons is an interesting special case of general simple polygons. Convex polygons in a general setting may: (1) Intersect; (2) Be enclosed in one another.

In the case when the convex polygons intersect, their bisectors become non-Jordan curves. This creates issues in the structural complexity analysis, which can be treated similarly to the intersection of line segments, see Section 3.3.

Closed Bisectors More interesting is the case of the convex polygons which are allowed to be enclosed. Consider a pair of convex polygons, where one is enclosed inside another. The bisector between two enclosed convex polygons is a locus of points equidistant from their boundaries, which is a closed curve.

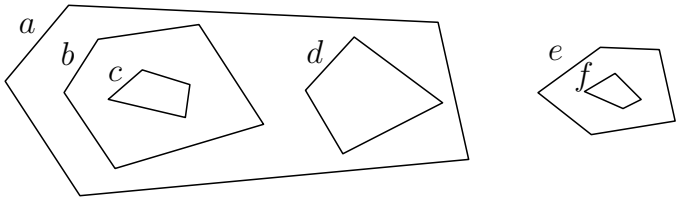

Figure 7.1. Enclosed convex polygons.

Closed bisectors bring problems into the structural complexity analysis. In particular, the edges of the Voronoi diagram form no longer a connected structure. A direct consequence is that Euler's formula has to include an additional variable that represents the number of the connected components. Each order-i Voronoi diagram will have its own variable C_i representing the number of connected components in the graph structure of the order-i Voronoi diagram, where $1 \leq i < n$. The variables $C_1, \ldots, C_{n-1}$ can be related to the way the convex polygons are enclosed. Additionaly, the enclosed convex polygons influence the number of unbounded faces in the order-k Voronoi diagram. This influence may be related to the variables $C_1, \ldots, C_{n-1}$.

The inclusions of convex polygons have the following important property. Let aEb denote the relation in which polygon b is enclosed in a. Then the following property holds:

$$aEc \wedge bEc \rightarrow (aEb \vee bEa)$$

For instance, in Figure 7.1, polygon c is enclosed in polygons a and b, which means that either b is enclosed in a or a is enclosed in b. The relation E then forms a forest-like structure, which can be used to prove many properties. For instance one can probably prove that $C_1 + \ldots + C_{n-1} \leq 2n$.

General Simple Polygons General simple polygons introduce additional complications. Consider a simple polygon A and its convex hull $conv(A)$. We say that simple polygon B is in the *pocket* of the simple polygon A if B is not enclosed in A, but is enclosed in $conv(A)$. In this case, the bisector between A and B is a closed curve. The closed bisectors created by the *pocket* inclusions have the same complications as the closed bisectors created by the regular inclusions. Unlike the regular inclusions, the *pocket* inclusions are related in a much more complicated way. Let aPb denote the relation in which polygon b is enclosed in the pocket of polygon a. Then aPc and bPc does not imply aPb or bPa.

$$aPc \wedge bPc \not\rightarrow (aPb \vee bPa)$$

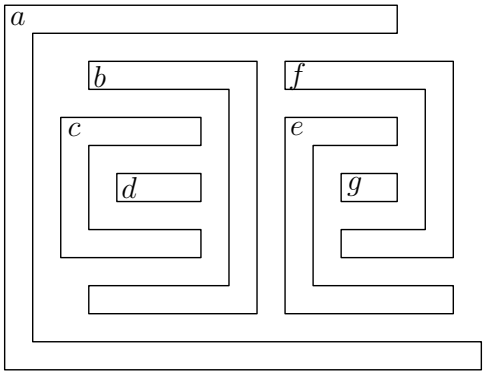

Figure 7.2. Simple polygons enclosed in the "pockets".

For instance, in Figure 7.2, the polygon d is in the pocket of c and the pocket of b. However, neither is c in the pocket of b, nor is b in the pocket of c.

7.1.2 Algorithms for Higher-Order Abstract Voronoi Diagrams

In Chapter 6 we presented three randomized construction algorithms for higher-order abstract Voronoi diagrams. These algorithms are the extensions of the construction algorithms for points to the abstract case, see Section 2.3 for an overview. A possible future direction is the extension of the algorithms for points to the abstract case. We consider the result of Chan [23] to be the best candidate. The most interesting part of the result described by Chan is the following theorem:

Given an algorithm that constructs a k-level of n planes in $\mathbb{R}^3$ in $O(f(n))$ time (where $f(n)$ is a regular function) we can construct the k-level in $O(n \log n + (n/k) f(k))$ expected time.

This result is based on the range reporting data structure described in the Section 2.3. If we could apply this data structure in the abstract case we would significantly improve the running time of the algorithms described in Chapter 6. In particular, if we could apply the result of Chan to the randomized divide and conquer algorithm we would obtain an algorithm with $O(n \log n + nk^{1+\varepsilon})$ expected time complexity. If we could apply the result of Chan to the random walk method we would obtain an algorithm with $O(n \log n + nk2^{\alpha(k)} \log k)$ expected time complexity.

The data structure of Chan operates with simplices. In the case of the abstract Voronoi diagrams we could use the trapezoidal decomposition similarly as in the Section 6.2.1.

Bibliography

[1] Agarwal, P. K., Aronov, B., Chan, T. M. and Sharir, M. [1998]. On levels in arrangements of lines, segments, planes, and triangles, *Discrete & Computational Geometry* **19**(3): 315–331.

[2] Agarwal, P. K., de Berg, M., Matousek, J. and Schwarzkopf, O. [1998]. Constructing levels in arrangements and higher-order Voronoi diagrams, *SIAM J. Comput.* **27**(3): 654–667.

[3] Agarwal, P. K., Schwarzkopf, O. and Sharir, M. [1996]. The overlay of lower envelopes and its applications, *Discrete & Computational Geometry* **15**(1): 1–13.

[4] Agarwal, P. K. and Sharir, M. [2000]. *Arrangements and Their Applications*, Elsevier Science Publishing, chapter V, pp. 49–120. SFB Report F003-092, TU Graz, Austria, 1996.

[5] Aggarwal, A., Guibas, L. J., Saxe, J. B. and Shor, P. W. [1989]. A linear-time algorithm for computing the Voronoi diagram of a convex polygon, *Discrete & Computational Geometry* **4**: 591–604.

[6] Alon, N. and Györi, E. [1986]. The number of small semispaces of a finite set of points in the plane, *J. Comb. Theory, Ser. A* **41**(1): 154–157.

[7] Alt, H., Cheong, O. and Vigneron, A. [2005]. The voronoi diagram of curved objects, *Discrete & Computational Geometry* **34**(3): 439–453.

[8] Aurenhammer, F. [1990]. A new duality result concerning Voronoi diagrams, *Discrete & Computational Geometry* **5**: 243–254.

[9] Aurenhammer, F., Drysdale, R. L. S. and Krasser, H. [2006]. Farthest line segment Voronoi diagrams, *Inf. Process. Lett.* **100**(6): 220–225.

[10] Aurenhammer, F., Klein, R. and Lee, D.-T. [2013]. *Voronoi Diagrams and Delaunay Triangulations*, World Scientific.

[11] Aurenhammer, F. and Schwarzkopf, O. [1992]. A simple on-line randomized incremental algorithm for computing higher-order Voronoi diagrams, *Int. J. Comput. Geometry Appl.* **2**(4): 363–381.

[12] Bárány, I., Füredi, Z. and Lovász, L. [1990]. On the number of halving planes, *Combinatorica* **10**(2): 175–183.

[13] Berg, M., Cheong, O., Kreveld, M. and Overmars, M. [2008]. *Computational Geometry: Algorithms and Applications*, third edn, Springer.

[14] Blum, H. [1967]. A Transformation for Extracting New Descriptors of Shape, *in* W. Wathen-Dunn (ed.), *Models for the Perception of Speech and Visual Form*, MIT Press, Cambridge, pp. 362–380.

[15] Blum, H. [1973]. Biological shape and visual science (part I), *Journal of Theoretical Biology* **38**(2): 205–287.

[16] Blum, H. and Nagel, R. N. [1978]. Shape description using weighted symmetric axis features, *Pattern Recognition* **10**(3): 167–180.

[17] Bohler, C., Cheilaris, P., Klein, R., Liu, C.-H., Papadopoulou, E. and Zavershynskyi, M. [2013]. On the complexity of higher-order abstract Voronoi diagrams, *in* F. V. Fomin, R. Freivalds, M. Z. Kwiatkowska and D. Peleg (eds), *ICALP (1)*, Vol. 7965 of *Lecture Notes in Computer Science*, Springer, pp. 208–219.

[18] Boissonnat, J.-D., Devillers, O. and Teillaud, M. [1993]. A semidynamic construction of higher-order Voronoi diagrams and its randomized analysis, *Algorithmica* **9**(4): 329–356.

[19] Boots, B. and South, R. [1997]. Modelling retail trade areas using higher-order, multiplicatively weighted Voronoi diagrams, *Journal of Retailing* **73**(4): 519–536.

[20] Brown, K. Q. [1979]. *Geometric Transforms for Fast Geometric Algorithms*, PhD thesis, Pittsburgh, PA, USA. AAI8012772.

[21] Calabi, L. and Hartnett, W. E. [1968]. Shape recognition, prairie fires, convex deficiencies and skeletons, *The American Mathematical Monthly* **75**(4): pp. 335–342.

[22] Chan, T. M. [1999]. Remarks on k-level algorithms in the plane., *Technical report*.

[23] Chan, T. M. [2000]. Random sampling, halfspace range reporting, and construction of $\leq k$-levels in three dimensions, *SIAM J. Comput.* **30**(2): 561–575.

[24] Chan, T. M. [2003]. On levels in arrangements of curves, *Discrete & Computational Geometry* **29**(3): 375–393.

[25] Chan, T. M. [2005]. On levels in arrangements of curves, ii: A simple inequality and its consequences, *Discrete & Computational Geometry* **34**(1): 11–24.

[26] Chan, T. M. [2008]. On levels in arrangements of curves, iii: further improvements, *in* M. Teillaud (ed.), *Symposium on Computational Geometry*, ACM, pp. 85–93.

[27] Chan, T. M. [2012]. On levels in arrangements of surfaces in three dimensions, *Discrete & Computational Geometry* **48**(1): 1–18.

[28] Chazelle, B. and Edelsbrunner, H. [1987]. An improved algorithm for constructing k th-order Voronoi diagrams, *IEEE Trans. Computers* **36**(11): 1349–1354.

[29] Cheong, O., Everett, H., Glisse, M., Gudmundsson, J., Hornus, S., Lazard, S., Lee, M. and Na, H.-S. [2011]. Farthest-polygon Voronoi diagrams, *Comput. Geom.* **44**(4): 234–247.

[30] Clarkson, K. L. [1987]. New applications of random sampling in computational geometry, *Discrete & Computational Geometry* **2**: 195–222.

[31] Clarkson, K. L. and Shor, P. W. [1989]. Application of random sampling in computational geometry, ii, *Discrete & Computational Geometry* **4**: 387–421.

[32] Cuel, L., Lachaud, J.-O., Mérigot, Q. and Thibert, B. [2014]. Robust normal estimation using order-k Voronoi covariance, *30th European Workshop on Computational Geometry (EuroCG 2014)* .

[33] Dehne, F. K. H. A. and Klein, R. [1997]. "The Big Sweep": On the power of the wavefront approach to Voronoi diagrams, *Algorithmica* **17**(1): 19–32.

[34] Dey, S. K. and Papadopoulou, E. [2012]. The L_∞ (L_1) farthest line-segment Voronoi diagram, *ISVD*, IEEE, pp. 49–55.

[35] Dey, T. K. [1998]. Improved bounds for planar k-sets and related problems, *Discrete & Computational Geometry* **19**(3): 373–382.

[36] Division, I. B. M. C. R., Srinivasan, V., Nackman, L., Tang, J. and Meshkat, S. [1990]. *Automatic Mesh Generation Using the Symmetric Axis Transformation of Polygonal Domains*, Research report, IBM T.J. Watson Research Center.

[37] Edelsbrunner, H. [1986]. Edge-skeletons in arrangements with applications, *Algorithmica* **1**(1): 93–109.

[38] Edelsbrunner, H. [1987]. *Algorithms in Combinatorial Geometry*, Vol. 10 of *EATCS Monographs on Theoretical Computer Science*, Springer.

[39] Edelsbrunner, H., Guibas, L. J., Pach, J., Pollack, R., Seidel, R. and Sharir, M. [1992]. Arrangements of curves in the plane - topology, combinatorics and algorithms, *Theor. Comput. Sci.* **92**(2): 319–336.

[40] Edelsbrunner, H., Maurer, H. A., Preparata, F. P., Rosenberg, A. L., Welzl, E. and Wood, D. [1982]. Stabbing line segments, *BIT* **22**(3): 274–281.

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

[42] Erdős, P., Lovász, L., Simmons, A. and Straus, E. G. [1973]. Dissection graphs of planar point sets, *A Survey of Combinatorial Theory (Proc. Internat. Sympos., Colorado State Univ., Fort Collins, Colo., 1971). Amsterdam: North-Holland.* pp. 139–149.

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

[44] Fujii, A. [1976]. *Activity Contour Lines*, PhD thesis, Department of Architecture, University of Tokyo [in Japanese].

[45] Goodman, J. E. and Pollack, R. [1984]. On the number of k-subsets of a set of n points in the plane, *J. Comb. Theory, Ser. A* **36**(1): 101–104.

[46] Gupta, P. and Papadopoulou, E. [2008]. *Yield Analysis and Optimization*, Taylor & Francis CRC Press, chapter 7.3.

[47] Gürsoy, H. N. and Patrikalakis, N. M. [1992]. An automatic coarse and fine surface mesh generation scheme based on medial axis transform: Part ii implementation, *Engineering with Computers* **8**(4): 179–196.

[48] Har-Peled, S. [2000]. Taking a walk in a planar arrangement, *SIAM J. Comput.* **30**(4): 1341–1367.

[49] Huttenlocher, D. P., Kedem, K. and Sharir, M. [1993]. The upper envelope of Voronoi surfaces and its applications, *Discrete & Computational Geometry* **9**: 267–291.

[50] Keeney, R. [1972]. A method for districting among facilities, *Operations Research* **20**: 613–618.

[51] Khramtcova, E., Dey, S. K. and Papadopoulou, E. [2014]. Linear-time algorithms for the farthest-segment voronoi diagram and related tree structures, *CoRR* **abs/1411.2816**.

[52] Kirkpatrick, D. G. [1979]. Efficient computation of continuous skeletons, *FOCS*, IEEE Computer Society, pp. 18–27.

[53] Klein, R. [1989]. *Concrete and Abstract Voronoi Diagrams*, Vol. 400 of *Lecture Notes in Computer Science*, Springer.

[54] Klein, R., Langetepe, E. and Nilforoushan, Z. [2009]. Abstract Voronoi diagrams revisited, *Comput. Geom.* **42**(9): 885–902.

[55] Klein, R., Mehlhorn, K. and Meiser, S. [1993]. Randomized incremental construction of abstract Voronoi diagrams, *Comput. Geom.* **3**: 157–184.

[56] Lantuéjoul, C. and Maisonneuve, F. [1984]. Geodesic methods in quantitative image analysis, *Pattern Recognition* **17**(2): 177–187.

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

[58] Lee, D. T. [1982a]. Medial axis transformation of a planar shape, *IEEE Trans. Pattern Analysis & Machine Intelligence* **4**(4): 363–369.

[59] Lee, D.-T. [1982b]. On k-nearest neighbor Voronoi diagrams in the plane, *IEEE Trans. Computers* **31**(6): 478–487.

[60] Liu, C.-H., Papadopoulou, E. and Lee, D. T. [2011]. An output-sensitive approach for the L_1/L_∞ k-nearest-neighbor Voronoi diagram, *in* C. Demetrescu and M. M. Halldórsson (eds), *ESA*, Vol. 6942 of *Lecture Notes in Computer Science*, Springer, pp. 70–81.

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

[62] Mehlhorn, K., Meiser, S. and Rasch, R. [2001]. Furthest site abstract Voronoi diagrams, *Int. J. Comput. Geometry Appl.* **11**(6): 583–616.

[63] Mérigot, Q., Ovsjanikov, M. and Guibas, L. J. [2009]. Robust Voronoi-based curvature and feature estimation, *in* W. F. Bronsvoort, D. Gonsor, W. C. Regli, T. A. Grandine, J. H. Vandenbrande, J. Gravesen and J. Keyser (eds), *Symposium on Solid and Physical Modeling*, ACM, pp. 1–12.

[64] Mizutani, N., Watanabe, T., Yoshida, Y. and Okabe, N. [1993]. Extraction of contour lines by identification of neighbor relationships on a Voronoi line graph, *Systems and Computers in Japan* **24**(1): 57–68.

[65] Mount, D. M. and Arya, S. [2005]. ANN: A Library for Approximate Nearest Neighbor Searching http://www.cs.umd.edu/~mount/ANN/.

[66] Mulmuley, K. [1993]. Output sensitive and dynamic constructions of higher-order Voronoi diagrams and levels in arrangements, *J. Comput. Syst. Sci.* **47**(3): 437–458.

[67] Okabe, A., Boots, B., Sugihara, K. and Chiu, S. [2000]. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, second edn, Wiley Series in Probability and Statistics.

[68] Papadopoulou, E. [2001]. Critical area computation for missing material defects in VLSI circuits, *IEEE Trans. on CAD of Integrated Circuits and Systems* **20**(5): 583–597.

[69] Papadopoulou, E. [2011]. Net-aware critical area extraction for opens in VLSI circuits via higher-order Voronoi diagrams, *IEEE Trans. on CAD of Integrated Circuits and Systems* **30**(5): 704–717.

[70] Papadopoulou, E. and Dey, S. K. [2012]. On the farthest line-segment Voronoi diagram, *in* K.-M. Chao, T. sheng Hsu and D.-T. Lee (eds), *ISAAC*, Vol. 7676 of *Lecture Notes in Computer Science*, Springer, pp. 187–196.

[71] Papadopoulou, E. and Lee, D. T. [1999]. Critical area computation via Voronoi diagrams, *IEEE Trans. on CAD of Integrated Circuits and Systems* **18**(4): 463–474.

[72] Papadopoulou, E. and Lee, D. T. [2001]. The L_∞-Voronoi diagram of segments and VLSI applications, *Int. J. Comput. Geometry Appl.* **11**(5): 503–528.

[73] Papadopoulou, E. and Zavershynskyi, M. [2014]. The higher-order Voronoi diagram of line segments, *CoRR* **abs/1405.3806**.

[74] Ramos, E. A. [1999]. On range reporting, ray shooting and k-level construction, *Symposium on Computational Geometry*, pp. 390–399.

[75] Rasch, R. [1994]. *Abstrakte inverse Voronoidiagramme*, PhD thesis, Saarbrücken, Germany.

[76] Rosenberger, H. [1991]. Order-k Voronoi diagrams of sites with additive weights in the plane, *Algorithmica* **6**(4): 490–521.

[77] Samet, H. [2005]. *Foundations of Multidimensional and Metric Data Structures (The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling)*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[78] Seidel, R. [1988]. Constrained Delaunay triangulations and Voronoi diagrams, *Report 260 IIG-TU, Graz, Austria* pp. 178–191.

[79] Seidel, R. [1998]. The nature and meaning of perturbations in geometric computing, *Discrete & Computational Geometry* **19**(1): 1–17.

[80] Setter, O., Sharir, M. and Halperin, D. [2010]. Constructing two-dimensional Voronoi diagrams via divide-and-conquer of envelopes in space, *Transactions on Computational Science* **9**: 1–27.

[81] Shamos, M. and Hoey, D. J. [1975]. Closest-point problems, *Proceedings of the Sixteenth IEEE Symposium on Foundations of Computer Science* pp. 151–162.

[82] Sharir, M. [2003]. The Clarkson-Shor technique revisited and extended, *Combinatorics, Probability & Computing* **12**(2): 191–201.

[83] Sharir, M. [2011]. An improved bound for k-sets in four dimensions, *Combinatorics, Probability & Computing* **20**(1): 119–129.

[84] Sharir, M. and Agarwal, P. K. [1995]. *Davenport-Schinzel sequences and their geometric applications*, Cambridge University Press.

[85] Sharir, M. and Smorodinsky, S. [2003]. Extremal configurations and levels in pseudoline arrangements, *in* F. K. H. A. Dehne, J.-R. Sack and M. H. M. Smid (eds), *WADS*, Vol. 2748 of *Lecture Notes in Computer Science*, Springer, pp. 127–139.

[86] Sharir, M., Smorodinsky, S. and Tardos, G. [2001]. An improved bound for k-sets in three dimensions, *Discrete & Computational Geometry* **26**(2): 195–204.

[87] Sibson, R. [1980]. A vector identity for the Dirichlet tessellation, *Mathematical Proceedings of the Cambridge Philosophical Society* **87**: 151–155.

[88] Smith, R. W. [1987]. Computer processing of line images: A survey, *Pattern Recognition* **20**(1): 7–15.

[89] Tóth, G. [2001]. Point sets with many k-sets, *Discrete & Computational Geometry* **26**(2): 187–194.

[90] Wagner, U. [2008]. k-sets and k-facets, *Discrete and Computational Geometry - 20 Years Later (Eli Goodman, János Pach, and Ricky Pollack, eds.), Contemporary Mathematics 453, American Mathematical Society* pp. 443–514.

[91] Yap, C.-K. [1987]. An $O(n \log n)$ algorithm for the Voronoi diagram of a set of simple curve segments, *Discrete & Computational Geometry* **2**: 365–393.