An Efficient Randomized Algorithm for Higher-Order Abstract Voronoi Diagrams*

Cecilia Bohler[1], Rolf Klein[2], and Chih-Hung Liu[3]

1 **Department of Computer Science, University of Bonn, Bonn, Germany**
bohler@cs.uni-bonn.de
2 **Department of Computer Science, University of Bonn, Bonn, Germany**
rolf.klein@uni-bonn.de
3 **Department of Computer Science, University of Bonn, Bonn, Germany**
chliu@uni-bonn.de

Abstract

Given a set of n sites in the plane, the order-k Voronoi diagram is a planar subdivision such that all points in a region share the same k nearest sites. The order-k Voronoi diagram arises for the k-nearest-neighbor problem, and there has been a lot of work for point sites in the Euclidean metric. In this paper, we study order-k Voronoi diagrams defined by an abstract bisecting curve system that satisfies several practical axioms, and thus our study covers many concrete order-k Voronoi diagrams. We propose a randomized incremental construction algorithm that runs in $O(k(n-k)\log^2 n + n\log^3 n)$ steps, where $O(k(n-k))$ is the number of faces in the worst case. Due to those axioms, this result applies to disjoint line segments in the L_p norm, convex polygons of constant size, points in the Karlsruhe metric, and so on. In fact, this kind of run time with a polylog factor to the number of faces was only achieved for point sites in the L_1 or Euclidean metric before.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Order-k Voronoi Diagrams, Abstract Voronoi Diagrams, Randomized Geometric Algorithms

Digital Object Identifier 10.4230/LIPIcs.SoCG.2016.21

1 Introduction

Given a set S of n sites in the plane, the order-k abstract Voronoi diagram $V_k(S)$ of S partitions the plane into Voronoi regions such that all points within a region $\mathrm{VR}_k(H,S)$ share the same set H of k nearest sites in S where the underlying proximity is defined by an abstract bisecting curve system. Order-k Voronoi diagrams solve the k nearest neighbor problem. However, the distance measure in the real world is in general not the Euclidean but depends on the type of geometric sites and the underlying environment. To deal with diverse proximities, Klein [16] introduced the notion of *abstract Voronoi diagrams* to model the proximity by an abstract bisecting curve system $\mathcal{J}$. For two sites $p, q \in S$, he considered a simple curve $J(p,q)$ as a bisector that splits the plane into two domains $D(p,q)$ and $D(q,p)$. $D(p,q)$ represents the set of points closer to p than to q. Under these circumstances, the order-k Voronoi region $\mathrm{VR}_k(H,S)$ is defined as:

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

* This work was supported by Deutsche Forschungsgemeinschaft (DFG Kl 655/19) in a DACH project.

If a bisecting curve system satisfies certain axioms, combinatorial properties and algorithms are directly applicable to the resulting Voronoi diagrams [3, 5, 16, 17, 18, 21].

We consider the following six axioms: for each subset S' of S of size at least 3,

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

(A2) Each point in the plane belongs to the closure of some first-order Voronoi region.

(A3) No first-order Voronoi region in $V_1(S')$ 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. At most three bisecting curves $J(p,\cdot)$ associated with the same site p can pass through the same point.

(A6) The number of vertical tangencies of a curve $J(p,q)$ is $O(1)$, and for any two curves $J(p,q)$ and $J(s,t)$, their vertical tangencies are distinct.

Despite Axiom (A1), $\mathrm{VR}_k(H,S)$, where $k>1$, may consist of disjoint *faces*, i.e., connected components. The six axioms cover many concrete Voronoi diagrams, so the abstract Voronoi diagram is a prototype of many concrete ones. It is known that axioms (A1) and (A2) need only be verified for subsets S' of size 3, and (A3), for size 4 subsets [17, 3]. The properties postulated in (A5) are to avoid technical difficulties; for order-1 abstract Voronoi diagrams it has been shown in [16, 17] how to proceed without these assumptions, and we are confident that the results in this paper can be generalized, too.

Lee [19] first proved that the order-k Voronoi diagram has $O(k(n-k))$ faces for point sites in the Euclidean metric. Only recently, Papadopoulou and Zavershynskyi [23] showed that the number of faces for disjoint line segments remains $O(k(n-k))$, and this bound remains valid for intersecting segments if $k \geq n/2$. Soon later, Bohler et al. [3] proved that the number of faces in the order-k abstract Voronoi diagram is at most $2k(n-k)$, which is a tight bound in the abstract setting. Gemsa et al. [14] and Liu and Lee [20] studied point sites in the city and geodesic metrics, respectively, neither of which lies under the envelope of the above axioms. Construction algorithms have been well-studied for point sites in the Euclidean metric. Deterministic algorithms by Lee [19] and by Chazelle and Edelsbrunner [10] achieve $O(k^2 n \log n)$ and $O(n^2 + k(n-k)\log^2 n)$ time, respectively. Clarkson [12] proposed a randomized divide-and-conquer algorithm with $O(kn^{1+\epsilon})$ time. Moreover, Aurenhammer and Schwarzkopf [2] and Boissonnat et al. [6] studied on-line algorithms. Most efficient algorithms rely on a geometric transformation that maps each point site to a hyperplane in three dimensions which is tangent to a unit paraboloid $z = x^2 + y^2$ at the vertical projection of the point site. In this situation, computing the order-k Voronoi diagram is reduced to computing the so called k-level of the arrangement formed by the transformed hyperplanes.

In recent years, some algorithms for settings different from point sites in the Euclidean metric were invented. For point sites in the L_1 metric, Liu et al. [20] derived an output-sensitive algorithm with $O(m \log n)$ time, where $m = O(\min\{k(n-k), (n-k)^2\})$; for line segments, Papadopoulou and Zavershynskyi [23] obtained $O(k^2 n \log n)$ time. Bohler et al. [5] developed a randomized divide-and-conquer algorithm for the abstract version, and obtained $O(kn^{1+\epsilon})$ time, which works for many concrete cases including disjoint line segments. Their algorithm interprets Clarkson's general idea [12] and replaces geometric operations with combinatorial ones.

Agarwal et al. [1] proposed a randomized incremental algorithm to compute the k-level which intermediately maintains cells that possibly intersect the final k-level, and their algorithm yields $O(k(n-k)\log n)$ construction time for order-k Voronoi diagrams. Chan [7] proposed a framework to compute the k-level, and adopted Agarwal et al's algorithm as a

black-box to obtain $O(nk \log k + n \log n)$ time. Ramos [24] later improved the construction time to $O(n \log n + nk^{O(\log^* k)})$, and his algorithm is a mixture of divide-and-conquer and incremental scheme with Chan's framework on the top. Very recently, Chan and Tsakalidis [8] derandomized Chan's framework and achieved $O(nk \log k)$ (or $O(nk \log k \log \log k)$) deterministic time.

It is an interesting question if a construction time with a polylogarithmic factor to the output size can be achieved for settings different from point sites in L_1 or L_2. In other words, is it possible to extend classical k-level algorithms [1, 7, 24] to abstract Voronoi diagrams? Such an extension would be quite different from Bohler et al.'s [5] extension of Clarkson [12], because Clarkson's algorithm principally adopts two planar sub-algorithms although he explained the general idea in three dimensions. In contrast, those k-level algorithms fully perform in three dimensions, exploiting the geometry of planes tangent to the paraboloid, and it seems quite challenging to convert them to non-point sites and non-Euclidean metrics, or even to the abstract version, based only on combinatorial properties of curves.

In this paper we give a first positive answer, proposing an $O(k(n-k) \log^2 n + n \log^3 n)$-time randomized incremental algorithm for order-k abstract Voronoi diagrams. Due to the six axioms, this result applies to a wide range of concrete order-k Voronoi diagrams including point sites in any algebraic convex distance metric or the Karlsruhe metric, disjoint line segments and disjoint convex polygons of constant size in the L_p norms, or under the Hausdorff metric (assuming for now sites to be in general position; see the comment on axiom (A5) from above). Such near-optimal run time is achieved for the first time for examples different from point sites in the Euclidean and L_1 metrics.

Our algorithm is strongly inspired by the k-level algorithm of Agarwal et al. [1]. We also proceed incrementally and maintain all faces of the higher-order Voronoi regions that are "active". This notion can be translated to abstract Voronoi diagrams in the following way. For a subset R of S and for each face F of $\mathrm{VR}_j(Q, R)$ we maintain its intersections with the farthest Voronoi diagram $\mathrm{FV}(Q)$, and with the nearest Voronoi diagram $V(R \setminus Q)$.

Each resulting sub-face $F \cap \mathrm{FVR}(q, Q)$ is decomposed into trapezoids. Such a trapezoid $\triangle$ is called *active* if, for some representative point $y \in \triangle$, the level of q at y does not exceed k plus the number of conflicts of $\triangle$. Here, the *level* of q equals the number of $s \in S$ such that $y \in D(s, q)$ holds, plus 1. The *conflicts* of $\triangle$ are sites $s \in S$ whose bisector $J(s, q)$ intersects $\triangle$ (or one of its defining edges, which may exceed the trapezoid).

Similarly, a trapezoid in a sub-face $F \cap \mathrm{VR}(p, R \setminus Q)$ is called active if the level of p, at some point, is larger than k minus the number of conflicts. Face F is called active if both intersections contain an active trapezoid.

The analysis in [1] relies on two simple facts: a hyperplane that crosses a segment must separate its two endpoints, and a new hyperplane separates a cell into two adjacent cells. The former implies that both the conflict size of a simplex (the number of hyperplanes intersecting it) and the level difference between two points in a cell (the number of hyperplanes separating them) are upper bounded by a constant times the diameter of the cell (the maximal number of hyperplanes intersecting a line segment in it); the latter implies that the diameter of one generated cell is at least half that of the original cell. However, this geometric analysis could not be directly applicable to the abstract setting because a bisector that intersects a vertical segment or a Voronoi edge does not necessarily separate its two endpoints, and a new site can separate a face into a non-constant number of new faces and they are even not necessarily pairwise adjacent. The second reason also illustrates an important phenomenon in the abstract setting that an order-k region may be disconnected for $k > 1$.

Therefore, we are using a different approach that may be interesting in its own right.

What one would really like to maintain, in the incremental construction, are those faces $F \subseteq \mathrm{VR}_j(Q,R)$ that contain a non-empty face of some region $\mathrm{VR}_k(H,S)$, such that $Q = H \cap R$ holds. We call these faces F *essential*. It is easy to verify that all essential faces are active. We were able to show some sort of converse: with high probability, all faces our algorithm constructs, including all active faces and those who are found to be inactive and discarded, are essential with respect to some order k' in a certain interval around k.

In [1], a new hyperplane partitions a cell of the arrangement into two cells. The geometry of the lowest-vertex triangulation can be used to decide the activity of the two new cells in time proportional to the total number of new or destroyed simplices. In the abstract setting, a face may be split into many faces, and in each of them old trapezoids may survive. Since we cannot afford to re-check them, in order to decide the activity of the new faces, we employ, for each active face, a nested data structure storing sub-faces and their edges. This approach only works because the intersections $F \cap \mathrm{FV}(Q)$ and $F \cap V(R \setminus Q)$ can be shown to be trees; for the second structure, this fact was not previously known. The tree structure allows the sub-faces of F in these intersections to be stored in cyclic order, which behave well under insertion of a new site. However, using the data structure approach incurs an extra $\log n$ factor in our run time bound.

When inserting a new site s, for each face F of $\mathrm{VR}_j(Q,R)$, we compute the sets

$$\begin{aligned} F \cap \mathrm{FVR}(s, Q \cup \{s\}) &\subseteq \mathrm{VR}_j(Q, R \cup \{s\}) \\ F \cap \mathrm{VR}(s, R \cup \{s\} \setminus Q) &\subseteq \mathrm{VR}_{j+1}(Q \cup \{s\}, R \cup \{s\}). \end{aligned}$$

Each of these intersections can consist of several connected components, giving rise to several faces of the new Voronoi regions on the right hand side. But the way F may split up is controlled by an unexpected structural property, as we will see below. This property also shows how disconnected Voronoi regions emerge.

To conclude, our algorithm can be seen as a combinatorial interpretation of Agarwal et al's algorithm [1], with a different analysis and with geometric properties replaced by combinatorial facts about bisecting curves. The algorithm applies to a wide range of order-k Voronoi diagrams, at the cost of an extra $O(\log n)$ factor. To some extent, our work decodes the powerful geometric transformation between points in the plane and hyperplanes in three dimensions, and indicates that the happy marriage between arrangements and random sampling techniques can be extended to more general proximities than point sites in the Euclidean metric.

2 Preliminaries

The common boundary between two faces in $V_k(S)$ is called a *Voronoi edge*, and the common vertex among more than two faces in $V_k(S)$ is called a *Voronoi vertex*. The *order-k Voronoi diagram* $V_k(S)$ equals the union of the boundaries of all order-k regions or, equivalently, the union of the intersections of the closures of any two order-k Voronoi regions.

Due to Axiom (5), a Voronoi vertex v among $\mathrm{VR}_k(H_1,S)$, $\mathrm{VR}_k(H_2,S)$, and $\mathrm{VR}_k(H_3,S)$ can be categorised into two types: v is called *new* if $|H_1 \cap H_2 \cap H_3| = k-1$, and v is called *old* if $|H_1 \cap H_2 \cap H_3| = k-2$ [3].

Due to Axioms (A1) and (A5), any two bisecting curves $J(p,q)$ and $J(p,r)$ intersect at most twice [16]. Moreover, the following lemma holds [18].

▶ **Lemma 1.** *For any three sites $p,q,r \in S$ one has $D(p,q) \cap D(q,r) \subseteq D(p,r)$.*

This transitivity property ensures that, for each point x not situated on any bisecting curve, the relation $p <_x q \iff x \in D(p,q)$ is a total ordering on any subset of S. For $R \subseteq S$ we define

$$l_p(x,R) := 1 + |\{t \in R \setminus \{p\} \mid x \in D(t,p)\}|$$

to be the *level of* p at point x with respect to R, abbreviated $l_p(x)$ if $R = S$. For every Voronoi vertex v of $V_k(S)$, if v is the intersection between $J(p,q)$, $J(p,t)$, and $J(q,t)$, we have $l_q(v) = l_p(v) = l_t(v) = k-2$ or $k-1$ depending on whether v is old or new.

We define Γ to be a large closed Jordan curve such that no pair of bisectors cross on or outside Γ, and each bisector crosses Γ exactly twice and these intersections are transversal. If we add Γ to $V_k(S)$ and cut off all parts contained in the outer domain, we obtain a connected graph without unbounded edges, so we can view all faces in $V_k(S)$ to be bounded.

In our algorithm, the following basic operations are assumed to take $O(1)$ time:

1. For an arbitrary point x, determine if x belongs to $D(p,q)$, $J(p,q)$ or $D(q,p)$.
2. For a point x on $J(p,q)$, along one direction of $J(p,q)$, determine the next intersection with $J(s,t)$ or a straight line, or determine the next point where the curve reverses direction.
3. For two points x, y on $J(p,q)$, determine which point comes first in a given direction.

Notations. For simplicity, the first order region of a site p in R will be denoted by $\mathrm{VR}(p,R)$, and the first order diagram by $V(R)$. Similarly, if R is of size r, the Voronoi diagram of order $r-1$ is the farthest diagram, $\mathrm{FV}(R)$, and the farthest region of a site q is $\mathrm{FVR}(q,R)$.

For a subset R of S, we define $\mathrm{VF}(R)$ as the collection of all faces in $V_j(R)$ for $1 \le j \le |R|-1$. For an open set $A \subseteq \mathbb{R}^2$, we use ∂A and $\mathrm{cl}A$ to denote its boundary and closure, respectively.

3 Main concepts

Let $(s_1, s_2, \ldots, s_n)$ be a random sequence of S, and let R_i be the first i sites in the sequence, i.e., $R_i = \{s_1, s_2, \ldots, s_i\}$. We incrementally insert a site in the sequence and finally obtain $V_k(S)$. However, since $E[\sum_{i=k+1}^{n} |V_k(R_i)|] = \Omega(nk^2)$ [2] but $|V_k(S)| = O(k(n-k))$ [3], it is too expensive to compute all $V_k(R_i)$ for $k+1 \le i \le n$.

We now describe an alternate approach inspired by [1].

3.1 Dominance

To compute the faces of the non-empty regions of $V_k(S)$ we are maintaining certain faces of lower-order diagrams.

▶ **Definition 2.** A face F_1 of $\mathrm{VR}_{j_1}(Q_1,R_1)$ *dominates* a face F_2 of $\mathrm{VR}_{j_2}(Q_2,R_2)$, where $R_1 \subseteq R_2$ and $j_1 \le j_2$, if $F_1 \supseteq F_2$ and $Q_1 = R_1 \cap Q_2$. A face is called *essential* if it dominates a face of $V_k(S)$.

We observe that $Q_1 = R_1 \cap Q_2$ implies $(R_1 \setminus Q_1) \cap Q_2 = \emptyset$. Moreover, $Q_1 \subseteq Q_2$ and $R_1 \subseteq R_2$ imply $\mathrm{VR}_{j_1}(Q_1,R_1) \supseteq \mathrm{VR}_{j_2}(Q_2,R_2)$, so that $F_1 \supseteq F_2$ already follows from $F_1 \cap F_2 \neq \emptyset$ since faces are path-connected. Clearly, the dominance relation is transitive.

Essential faces can be characterized in the following way.

▶ **Lemma 3.** *A face F of $VR_j(Q,R)$ dominates a face of $V_k(S)$ if and only if there exists a point $x \in F$ where $x \in FVR(q,Q)$ and $x \in VR(p, R \setminus Q)$ such that $l_q(x) \le k < l_p(x)$.*

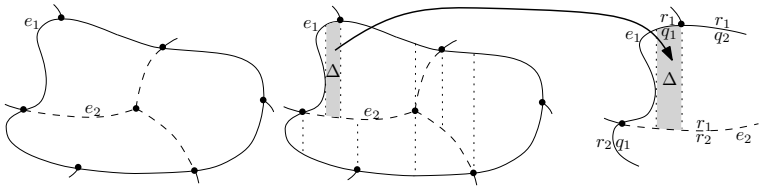

Figure 1 Left: $\check{F}= (V(R\setminus Q)\cap F)\cup\partial F$. Middle: $\check{F}^{\nabla}$. Right: $d(\triangle) = r_1$ and $D(\triangle)=\{r_1, r_2, q_1, q_2\}$.

Proof. *Necessity:* Let C be the face of $V_k(S)$ dominated by F and let C belong to $\mathrm{VR}_k(H, S)$. Consider a point $x \in C$ that does not lie on any bisector. Since H are the k nearest sites of x and $Q \subseteq H$, q is at most the k^{th} nearest neighbor of x, and since $(R \setminus Q) \cap H = \emptyset$, p is at least the $(k+1)^{\text{st}}$ nearest neighbor of x, implying $l_q(x) \leq k < l_p(x)$.

Sufficiency: Let C be the face of $V_k(S)$ that contains x and let C belong to $\mathrm{VR}_k(H, S)$. Since $l_q(x) \leq k$ and q is the farthest neighbor of x in Q, $Q \subseteq H$, and since $l_p(x) > k$ and p is the nearest neighbor of x in $R \setminus Q$, we have $(R \setminus Q) \cap H = \emptyset$, implying $Q = R \cap H$, so that F dominates C. ◀

Lemma 3 suggests to partition a face F of $\mathrm{VR}_j(Q, R)$ by $\mathrm{FV}(Q)$ and $V(R\setminus Q)$, respectively, and we use $\hat{F}$ and $\check{F}$ to denote these two subdivisions. Faces in $\hat{F}$ and $\check{F}$ are called *sub-faces*. The following lemma determines the structures of $\hat{F}$ and $\check{F}$, and implies that each sub-face of $\hat{F}$ or $\check{F}$ shares exactly one edge with the boundary of F, which is called an *outer* edge, while the edges of the refined Voronoi diagrams are called *inner* edges. (See Fig. 1)

▶ **Lemma 4.** *For a face F of $VR_j(Q, R)$, if $j > 1$, $FV(Q) \cap F$ is a tree, and if $j < |R| - 1$, $V(R \setminus Q) \cap F$ is a tree. (If $j = 1$ (resp. $j = |R| - 1$), $\hat{F}$ (resp. $\check{F}$) has exactly one sub-face.)*

3.2 Trapezoidal Decompositions

As before, let F be a face of $\mathrm{VR}_j(Q, R)$. For efficient computations, we partition (the sub-faces in) $\hat{F}$ and $\check{F}$ into vertical trapezoids, and denote the result by $\hat{F}^{\nabla}$ and $\check{F}^{\nabla}$, respectively. Abusing the notation slightly, we also use $\hat{F}$ and $\check{F}$ to denote the corresponding decompositions. F^{∇} is the set of trapezoids in $\hat{F}^{\nabla}$ and $\check{F}^{\nabla}$. For example, the middle drawing of Fig. 1 illustrates $\check{F}^{\nabla}$.

We assume that each trapezoid is adjacent to at most two trapezoids on either side. This assumption can be attained by inserting zero-width trapezoids whenever necessary [9].

▶ **Lemma 5.** *F^{∇} has $O(|F|)$ trapezoids, where $|F|$ is the number of edges bounding F.*

Now, we describe what information on single trapezoids we are interested in. For all trapezoids $\triangle$ in $\hat{F}^{\nabla}$ of a sub-face $F \cap \mathrm{FVR}(q, Q)$, we call q the *owner* of $\triangle$, denoted by $d(\triangle)$. Similarly, for all trapezoids $\triangle$ in $\check{F}^{\nabla}$ that decompose a sub-face $F \cap VR(p, R \setminus Q)$, we call $d(\triangle) := p$ the owner of $\triangle$.

By $D(\triangle)$ we denote the set of sites *defining* $\triangle$. Precisely, $\triangle$ is defined by at most two edges (top and bottom) and two vertical segments (left and right). Note that an edge here means an edge of $\hat{F}$ or $\check{F}$, and it could exceed the boundary of $\triangle$. For example, as shown in the right of Fig. 1, the top and bottom edges of $\triangle$ are e_1 and e_2. All the edges and the segments are associated with $d(\triangle)$-bisectors, and $|D(\triangle)| \leq 9$. See Fig. 1 for an illustration.

Moreover, for a trapezoid $\triangle$, we say a site $t \in S \setminus D(\triangle)$ is in *conflict with* $\triangle$ if $J(t, d(\triangle))$ intersects $\triangle$ or one of the two edges that define $\triangle$. In other words, $\triangle$ does not exist in $\mathrm{VF}^{\nabla}(R \cup \{t\})$, or one of its defining edges has been changed. It is clear that no site

$t \in R \setminus D(\triangle)$ can have a conflict with $\triangle$. We let $J(\triangle)$ be the set of sites t in conflict with $\triangle$, and $w(\triangle)$ be $|J(\triangle)|$. We call $J(\triangle)$ and $w(\triangle)$ the *conflict list* and *conflict size* of $\triangle$, respectively. In addition, we say a site $t \in S \setminus R$ *conflicts with* $F \in \mathrm{VF}(R)$ if t conflicts with a trapezoid of F^∇.

In the sequel, we will select an arbitrary point $y_\triangle \in \triangle$ for each trapezoid $\triangle \in F^\nabla$, and compute the level of its owner, $l_{d(\triangle)}(y_\triangle)$.

3.3 Active faces

Since it is hard to determine the existence of x in Lemma 3, instead of processing the essential faces, we define "active" faces in the following way.

▶ **Definition 6.** Consider a face F of $\mathrm{VR}_j(Q, R)$. For a trapezoid $\triangle$ of $\hat{F}^\nabla$, $\triangle$ is called *active* if $l_{d(\triangle)}(y_\triangle) - w(\triangle) \le k$; for a trapezoid $\triangle$ of $\check{F}^\nabla$, $\triangle$ is called *active* if $l_{d(\triangle)}(y_\triangle) + w(\triangle) > k$. Finally, F is called *active* if both $\hat{F}^\nabla$ and $\check{F}^\nabla$ contain an active trapezoid.

▶ **Lemma 7.** *An essential face must be active. Active faces in $VF(S)$ are faces in $V_k(S)$.*

4 Algorithm

Let $\mathrm{AF}(R)$ be the set of active faces in $\mathrm{VF}(R)$, the set of all faces of all higher order regions over site set R, and let AF_i be $\mathrm{AF}(R_i)$, for short. We first compute AF_{10} directly, and then from $i = 10$ to $i = n - 1$, we iteratively compute AF_{i+1} from AF_i, leading to the faces of $V_k(S)$ (Lemma 7). Finally, since adjacencies between faces are not recorded in AF_i, we gather all vertices of $V_k(S)$ from faces of $V_k(S)$, and link all those vertices in $O(k(n-k)\log n)$ time [5] to obtain $V_k(S)$.

During the incremental construction we maintain the following structures:

- The decompositions F^∇ for all faces $F \in \mathrm{AF}(R)$; for every trapezoid in F^∇, its at most four adjacent trapezoids, and its at most two defining edges; for every edge of $\hat{F}$ and $\check{F}$, all adjacent trapezoids on its two sides.
- The conflict lists: for every trapezoid $\triangle \in \mathrm{AF}(R)^\nabla$, the conflict list $J(\triangle)$, and for every site $s \in S \setminus R$, the set of all trapezoids of $\triangle \in \mathrm{AF}(R)^\nabla$ in conflict with s.
- Levels: for every trapezoid $\triangle \in \mathrm{AF}(R)^\nabla$, a chosen point $y_\triangle \in \triangle$ and $l_{d(\triangle)}(y_\triangle)$.
- Decompositions of nearest- and farthest-site Voronoi diagrams: $V(R)^\nabla$ and $\mathrm{FV}(R)^\nabla$.
- Data structures: a data structure for every face $F \in \mathrm{AF}(R)$, allowing the localization of trapezoids in F and a quick test of the active status of F; these data structures will be discussed in Section 4.4.

Once a new site $s \in S \setminus R$ has been randomly chosen for insertion, we find the trapezoids in conflict with s and, by means of the data structures, the faces they belong to.

4.1 Influence of a new site on an old face

Throughout the rest of this section, let F denote a face of $\mathrm{VR}_j(Q, R)$ in conflict with s. We compute the sets

$$F_- := F \cap \mathrm{FVR}(s, Q \cup \{s\})$$
$$F_+ := F \cap \mathrm{VR}(s, R \cup \{s\} \setminus Q)$$

by tracing their boundaries in F, denoted by J_- and J_+, through $\hat{F}^\nabla$ and $\check{F}^\nabla$, respectively.

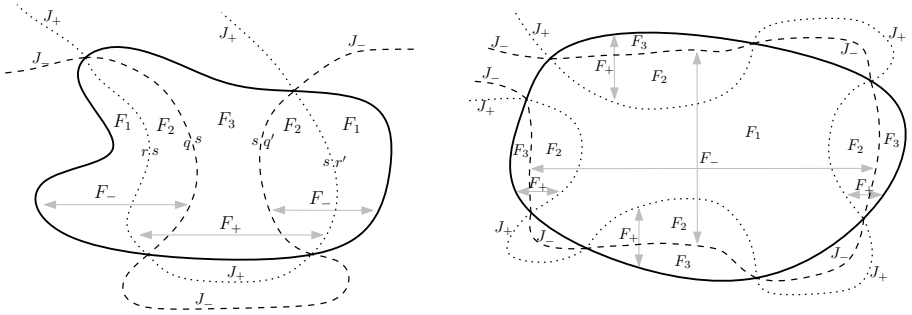

Figure 2 J_- partitions F into F_- and F_3, J_+ partitions F into F_+ and F_1, and $F_2 = F_+ \cap F_-$.

In this section we study which new faces arise from F and how to obtain their decompositions into farthest- and nearest-site Voronoi diagrams. In the subsequent sections, tracing of J_- and J_+ at the trapezoid level and updating trapezoids, including conflict lists and activity status, will be addressed.

The next lemma can be derived by evaluating local orderings.

▶ **Lemma 8.** *Each face of F_- is a face of $VR_j(Q, R \cup \{s\})$, and each face of F_+ is a face of $VR_{j+1}(Q \cup \{s\}, R \cup \{s\})$.*

Note that F_- and F_+ can be disconnected and overlap; see Fig. 2. Yet, these sets and their boundaries have surprising structural properties, as the following lemmas show.

▶ **Lemma 9.** *$F_- \cup F_+ = F$. J_- and J_+ do not intersect each other inside F.*

Proof. For the first statement, assume the contrary that there exists a point $x \in F$ such that $x \notin F_-$ and $x \notin F_+$. Since $x \notin F_-$, there exists a site $q \in Q$ such that $x \in D(s, q)$. Since $x \notin F_+$, there exists a site $p \in R \setminus Q$ such that $x \in D(p, s)$. By the transitivity (Lemma 1), $x \in D(p, q)$. However, since $x \in F$, $x \in D(q, p)$, leading to a contradiction.

For the second statement, if J_- and J_+ intersect inside F, $F_- \cup F_+ \subsetneq F$, contradicting the first statement. ◀

▶ **Lemma 10.** *Neither J_- nor J_+ contains a closed cycle in F*

Proof. If J_- contains a closed cycle, then $\mathrm{FV}(Q \cup \{s\})$ contains a bounded face, contradicting Lemma 4. Assume that J_+ contains a closed cycle in F, and let K be the region enclosed. Since J_+ is the boundary of $\mathrm{VR}(s, R \cup \{s\} \setminus Q)$, K must be $\mathrm{VR}(s, R \cup \{s\} \setminus Q)$; otherwise, there must exist a site $t \in R \setminus Q$ such that $\mathrm{VR}(t, R \cup \{s\} \setminus Q)$ is enclosed by $\mathrm{VR}(s, R \cup \{s\} \setminus Q)$, implying that $J(s, t)$ is closed, i.e., a contradiction. In this situation, J_+ is exactly a closed curve. Since $F_- \cup F_+ = F$ (Lemma 9), J_- does not contain any closed cycle, and J_+ is a closed curve, F_- is exactly F, and contains $\mathrm{VR}(s, R \cup \{s\} \setminus Q)$. By Lemma 8, F_- is a face of $\mathrm{VR}_j(Q, R \cup \{s\})$, and by Lemma 4, each face of $F_- \cap V(R \cup \{s\} \setminus Q)$ touches the boundary of F. However, since J_+ is a closed curve in F and does not intersect ∂F, $\mathrm{VR}(s, R \cup \{s\} \setminus Q)$ forms a bounded region in $F_- \cap V(R \cup \{s\} \setminus Q)$, contradicting Lemma 4. ◀

▶ **Lemma 11.** *J_- intersects ∂F at a point x if and only if J_+ intersects ∂F at x. Hence, J_- intersects F if and only if J_+ intersects F.*

Proof. For the first statement, we show necessity as follows. Let e be the Voronoi edge of F which contains x and assume that e is between $\mathrm{VR}_j(Q, R)$ and $\mathrm{VR}_j(Q \cup \{p\} \setminus \{q\}, R)$, where $q \in Q$ and $p \in R \setminus Q$, i.e., $e \subset J(p, q)$. In this situation, q is the farthest site of x in Q, and

p is the nearest site of x in $R \setminus Q$. Since $x \in \mathrm{FVR}(q, Q)$ and J_- passes through x, $J(q, s)$ passes through x. Since $J(q, s)$ and $J(p, q)$ intersect at x, $J(p, s)$ passes through x. Since $x \in \mathrm{VR}(p, R \setminus Q)$ and $J(p, s)$ passes through x, J_+ passes through x.

The proof of sufficiency is symmetric to the necessity proof. The second statement directly follows from the first statement and Lemma 10. ◀

For short, we let $F_1 := F \setminus F_+$ and $F_3 := F \setminus F_-$. By the above, the set $F_2 := F_- \cap F_+$ may consist of several faces, each of which is bounded by one segment of J_- and one of J_+. Each face of F_2 touches ∂F in exactly two points where J_- and J_+ meet; see Fig. 2. The following observation helps in constructing the farthest-site and nearest-site Voronoi diagrams inside the new faces.

▶ Observation. For points $x_1 \in F_1$, $x_2 \in F_2$, and $x_3 \in F_3$, the levels of the new site s at x_1, x_2, and x_3 with respect to $R \cup \{s\}$ are at least $j+2$, exactly $j+1$ and at most j, respectively.

For each face C of F_-, we obtain $\mathrm{FV}(Q) \cap C$ by clipping $\mathrm{FV}(Q) \cap F$ with the segments of J_- bounding C. To obtain $V(R \cup \{s\} \setminus Q) \cap C$ we observe that inside F_1 site s cannot be the nearest; here we can keep $V(R \setminus Q)$, clipped by J_+. To be added is a face of F_2 bounded by J_+ and J_- which equals the nearest Voronoi region of s in $V(R \cup \{s\} \setminus Q)$ inside C.

Similarly, let C' be a face of F_+. Inside F_3, site s cannot be the farthest in $Q \cup \{s\}$; here we can still use $\mathrm{FV}(Q) \cap F$, clipped by J_-. To obtain $\mathrm{FV}(Q \cup \{s\}) \cap C'$ we add the same face of F_2 as above, which also equals the farthest region of s in $\mathrm{FV}(Q \cup \{s\})$ inside C'. Moreover, we clip $V(R \setminus Q) \cap F$ by $J_+ \cap C'$ to obtain $V(R \cup \{s\} \setminus (Q \cup \{s\}) \cap C'$; see Figures 2 and 3.

To sum up, once we have J_- and J_+, we can generate new faces from F and update their farthest-site and nearest-site Voronoi diagrams. Since this process will not generate $\mathrm{VR}_1(\{s\}, R \cup \{s\})$ and $\mathrm{VR}_r(R, R \cup \{s\}) = \mathrm{FVR}(s, R \cup \{s\})$, we maintain $V(R)$ and $\mathrm{FV}(R)$ separately during the incremental construction, as well as their trapezoidal decompositions, conflict lists, and active statuses. The total expected time for this extra task is in $O(n \log n)$; see [21, 18, 4].

▶ Remark. Generating F_- from F is equivalent to removing F_3 from F. Since each face C of F_- is a face of $\mathrm{VR}_j(Q, R \cup \{s\})$, one may wonder which faces are adjacent to C in $V_j(R \cup \{s\})$. These faces belong to F'_+, for some faces F' in $V_{j-1}(R)$, and are obtained in the same way F_+ is obtained from F.

4.2 Tracing bisecting curves

The boundary J_- of F_- in F ($J_- = \partial F_- \cap F$) can be traced through $\hat{F}^\nabla$ using local information stored at trapezoids, starting from a trapezoid in conflict with s one of whose edges is external, i. e., belongs to ∂F. Namely, if site q is the owner $d(\triangle)$ of a trapezoid $\triangle$ then $J_- = J(q, s)$ holds inside $\triangle$. This stays true as J_- moves into a neighboring trapezoid through a vertical edge of $\triangle$. If J_- leaves $\triangle$ through its top or bottom edge e, two cases are possible. If $e \subset J(q, q')$ is an inner edge, J_- now becomes part of $J(q', s)$. In case of an outer edge on the boundary of F, tracing this segment of J_- halts.

Since two s-bisectors intersect at most twice, J_- intersects the top and the bottom edges of $\triangle$ at most four times, and since a bisector has a constant number of vertical tangencies, J_- intersects the left and right segments of $\triangle$ at most $O(1)$ times. Therefore, the size of $J_- \cap \triangle$ is $O(1)$, and only $O(1)$ time will be spent on computing $J_- \cap \triangle$. Thus, the time to trace J_- is upper bounded by the number of trapezoids in $\hat{F}^\nabla$ that are in conflict with s, times a constant.

The same holds for tracing J_+ through $\check{F}^\nabla$.

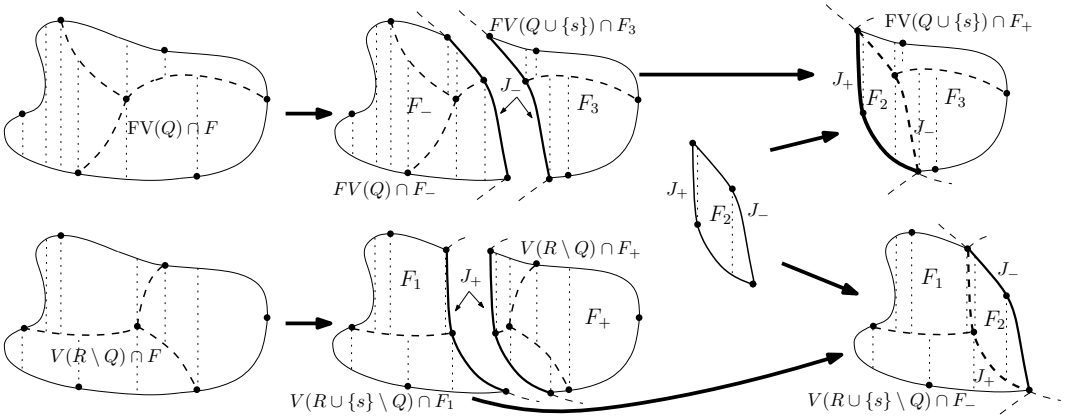

Figure 3 Computation of $F_-^{\triangledown}$ and $F_+^{\triangledown}$ from $F^{\triangledown}$. Two copies of F_2 are needed.

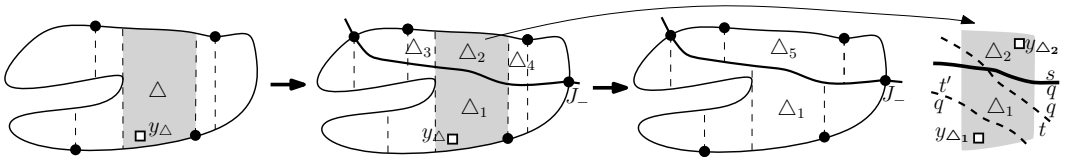

Figure 4 Separating a trapezoid and merging trapezoids.

4.3 Updating Trapezoidal Decompositions

As Fig. 3 illustrates, in F_- and F_3 existing trapezoids will be altered by J_- and J_+, whereas in F_2 a new trapezoid decomposition has to be built from scratch. While the geometric changes to the trapezoids are quite straightforward to implement, we also have to update conflict lists and activity statuses.

4.3.1 Trapezoids in F_- and F_3

Reconstructing $\hat{F_-}^{\triangledown}$ and $\hat{F_3}^{\triangledown}$, together with their conflict lists, is similar to the standard incremental construction for vertical trapezoidal decompositions (see Mulmuley's book [22]). It involves separating existing trapezoids by J_- and merging trapezoids sharing the same edges, and takes time proportional to the total conflict size of destroyed trapezoids; see Fig. 4. Also, testing the new trapezoids for activity is quite straightforward. Thus we only remark on a subtle fact here: a trapezoid $\triangle'$ generated from an inactive trapezoid $\triangle$ may be active although $\triangle'$ is smaller than $\triangle$. This is because the top or bottom edge changes such that a site t that does not conflict with $\triangle$ may conflict with $\triangle'$. As shown in Fig. 5, $J(q', t)$ and $J(q', t)$ intersect the top edge e' of $\triangle'$ but not the top edge e of $\triangle$.

4.3.2 Trapezoids in F_2

We separate each edge of F_2 into x-monotone curves, and apply Chazelle's algorithm [9] to compute the vertical trapezoidal decomposition for each face of F_2. His algorithm works for x-monotone curves, and takes time proportional to the number of created trapezoids.

To construct the conflict lists, we note that each face of F_2 is bounded by a curve of J_- and a curve of J_+, due to the results in Subsection 4.1, and each trapezoid in $F_2^{\triangledown}$ is dominated by s. Consider a face C of F_2. Since there is no closed bisector, if a trapezoid in

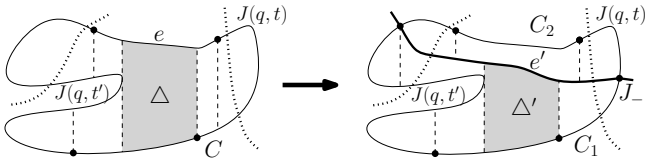

Figure 5 C is a sub-face of $F \cap \mathrm{FVR}(q, Q)$, and t and t' conflict with $\triangle'$ but not $\triangle$.

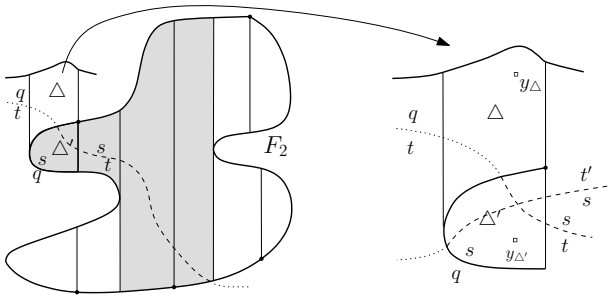

Figure 6 Left: Tracing conflicts for t where $d(\triangle) = q$ and $d(\triangle') = s$. Right: Walking from y to y' one must pass through $J(q, s)$, $J(t', s)$ and $J(s, t)$, so $l_s(y') = l_q(y) + 2 - 1$.

C^∇ is conflicted by a site $t \in S \setminus (R \cup \{s\})$, $J(s, t)$ must intersect the boundary of C, and thus t must also be in conflict with a trapezoid outside C and adjacent to the boundary of C. Therefore, we can compute conflict lists for trapezoids in C^∇ from the outside of C (see the left drawing of Fig. 6).

To check the active statuses, for each face C of F_2, we select a pair of trapezoids, $\triangle$ and $\triangle'$, outside and inside C, respectively, such that they share an edge along the boundary of C. Recall that $d(\triangle') = s$. Since $l_{d(\triangle)}(y_\triangle)$ is known, we arbitrarily choose a point $y'_\triangle \in \triangle'$, and compute $l_s(y_{\triangle'})$ by testing for each site $t \in J(\triangle) \cup J(\triangle')$ if $J(t, s)$ separates $y_\triangle$ and $y_{\triangle'}$ (see the right drawing of Fig. 6). Note that $J(s, d(\triangle))$ separates $y_\triangle$ and $y_{\triangle'}$, and $D(t, s) \cap D(s, d(\triangle)) \subseteq D(t, d(\triangle))$ holds by Lemma 1. Then, we traverse from $\triangle'$ to other trapezoids in C^∇, choose a point for each of them, and compute the corresponding level similarly. All these operations take time proportional to the total conflict size of those traversed trapezoids. Since all involved trapezoids are newly created and tested at most 4 times, the total time is proportional to the total conflict size of newly created trapezoids.

To sum up, the time to update trapezoids is proportional to the number of destroyed and newly created trapezoids and their total conflict size. Actually, the time for $\hat{F}_-^\nabla$, $\hat{F}_3^\nabla$, $\check{F}_+^\nabla$ and $\check{F}_1^\nabla$ is charged to destroyed trapezoids, and the time for F_2^∇ is charged to trapezoids newly created.

4.4 Updating Active Status of Faces

After generating new faces from a face F, we need to remove F, test new faces for being active, and remove the inactive ones, together with their trapezoids. The problem is that a new face may be active because it contains an old active trapezoid $\triangle$ that was *not* in conflict with s, so that we could not afford to visit $\triangle$.

We suggest the following solution. By Lemma 4, sub-faces of F in $\hat{F}$ or $\check{F}$ form a cyclic sequence along the boundary of F, where each sub-face appears once. Let us call a sub-face active if it contains an active trapezoid. We group consecutive inactive sub-faces into a single element and store a cyclic list of such elements and active sub-faces. Then, $\hat{F}$ contains an

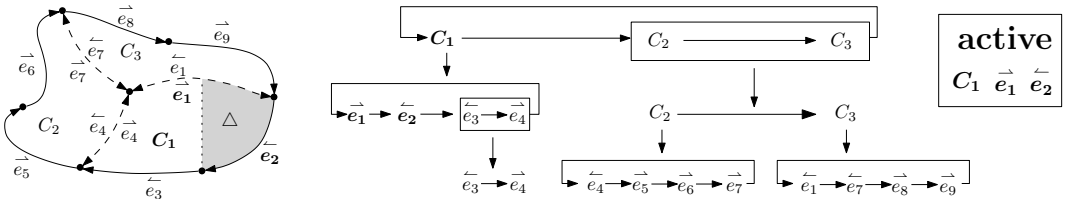

Figure 7 Left: Edges of $\hat{F}$ (dash: inner, solid: outer). Right: a 3-layer data structure for $\hat{F}$.

active sub-face if and only if this list contains more than one entry, or if its only entry is an active sub-face.

Each internal edge appears in two copies, one for the trapezoids on either side. Such a *half-edge* is called active if one of its adjacent trapezoids is active. For each sub-face C we group together inactive half-edges along ∂C, and obtain another cyclic sequence of such elements and active half-edges. Again, we can decide in $O(1)$ time if the sub-face C is active.

These lists are stored in a 3-level data structure, as shown in Fig. 7, one for each active face. If implemented as an enhanced red-black tree [25], operations insert, delete, concatenation, split, and find-set run in $O(\log m)$ amortized time (or faster), where m counts the total number of operations. When considering site set R, m can be upper bounded by $r^{O(1)}$, so that each operation takes amortized time in $O(\log r)$. This data structure allows to determine the activity of a face in constant time. How to maintain it under insertion of a new site s will be stated in a complete version later. We remark that the time to update the data structures is $\log r$ times the number of destroyed and newly created trapezoids.

5 Analysis

The time to insert a new site is proportional to the total conflict size of destroyed and newly created trapezoids plus $\log n$ times their number. However, not all the created trapezoids belong to an active face, so it is not sufficient to analyze active faces in AF_r.

While all essential faces (i. e., those dominating an order-k face) are active, the converse does not hold. But we are able to prove a slightly weaker fact.

▶ **Definition 12.** For a subset R of S of size r, let $M(R)$ denote the collection of faces in $\mathrm{VF}(R)$ each of which dominates a face of $V_{k'}(S)$, where $k - 6c\frac{n}{r}\log r \le k' \le k + 6c\frac{n}{r}\log r$.

We are proving that with high probability $M(R_{r+1})$ contains all faces created during the insertion of site s_{r+1}, both active ones and inactive ones that were discarded.

Since $\mathrm{VF}(R_r)^\nabla$ has $O(r^3)$ trapezoids, the time to insert the $(r+1)^{\mathrm{st}}$ site is trivially $O(nr^3)$. Therefore, if the probability for $M(R)$ to have the above property is high enough, say $1-O(r^{-5})$, the insertion time, provided that $M(R)$ fails to have it, is $O(nr^3)*O(r^{-5}) = O(\frac{n}{r^2})$. Since $\sum_{r=10}^{n} O(\frac{n}{r^2})$ is only $O(n)$, the analysis of the positive case will dominate the overall expectation.

To this end, we define a kind of ϵ-nets for $\epsilon = c\frac{1}{r}\log r$ as follows.

▶ **Definition 13.** For a subset R of S of size r and for a constant $c \ge 22$, R is an ϵ-net if

1. For each trapezoid $\triangle \in \mathrm{VF}^\nabla(R)$, $w(\triangle) \le c\frac{n}{r}\log r$,
2. and for each face $F \in \mathrm{VF}(R)$ where $F \subseteq \mathrm{VF}_j(Q,R)$ and for any two points $x, x' \in F$ where $x \in \mathrm{FVR}(q,Q) \cap \mathrm{VR}(p, R \setminus Q)$ and $x' \in \mathrm{FVR}(q',Q) \cap \mathrm{VR}(p', R \setminus Q)$,

$$\max\{|l_q(x) - l_{q'}(x')|, |l_p(x) - l_{p'}(x')|, l_p(x) - l_q(x)\} \le c\frac{n}{r}\log r.$$

▶ **Lemma 14.** *If R_r is an ϵ-net, then $M(R_{r+1})$ contains all the created faces due to the insertion of s_{r+1}.*

Proof. Consider a face F of $\mathrm{AF}(R_r)$ in conflict with s_{r+1}. It is sufficient to prove that all faces generated from F due to the insertion of s_{r+1} belong to $M(R_{r+1})$. Let F belong to $\mathrm{VR}_j(Q, R_r)$. We will prove that each face C of F_- belongs to $M(R_{r+1})$, and it is symmetric for each face of F_+. (Recall that $F_- = F \cap \mathrm{VR}_j(Q, R_{r+1})$ and $F_+ = F \cap \mathrm{VR}_{j+1}(Q \cup \{s_{r+1}\}, R_{r+1})$.) Take a point $x \in C$ where $x \in \mathrm{VR}(s_{r+1}, R_{r+1} \setminus Q)$ and $x \in \mathrm{FVR}(q, Q)$. Since $C \subseteq F$, we also let x belong to $\mathrm{VR}(p, R_r \setminus Q)$. In other words, $x \in D(s_{r+1}, p)$, so $l_q(x) < l_{s_{r+1}}(x) < l_p(x)$.

Since F is active, there exists a trapezoid $\triangle \in \hat{F}^{\nabla}$ such that there exists a point $y \in \triangle$ with $l_{d(\triangle)}(y) - w(\triangle) \le k$. Since R_r is an ϵ-net, $w(\triangle) \le c\frac{n}{r}\log r$ and $l_q(x) - l_{d(\triangle)}(y) \le c\frac{n}{r}\log r$, we have $l_q(x) \le k + 2c\frac{n}{r}\log r$. Similarly, we can derive $l_p(x) \ge k - 2c\frac{n}{r}\log r$. Since $l_p(x) - l_q(x) \le c\frac{n}{r}\log r$ (by the definiton of an ϵ-net), we have

$$k - 3c\frac{n}{r}\log r \le l_q(x) < l_p(x) \le k + 3c\frac{n}{r}\log r.$$

Finally, since $l_q(x) < l_{s_{r+1}}(x) < l_p(x)$ and $2\frac{n}{r+1}\log(r+1) \ge \frac{n}{r}\log r$, we have

$$k - 6\frac{n}{r+1}\log(r+1) \le l_q(x) < l_{s_{r+1}}(x) < l_p(x) \le k + 6\frac{n}{r+1}\log(r+1),$$

implying that C dominates a face of $V_{k'}(S)$ where $k - 6c\frac{n}{r+1}\log(r+1) \le k' \le k + 6c\frac{n}{r+1}\log(r+1)$ (see Lemma 3) and thus belongs to $M(R_{r+1})$. ◀

Using general ideas by Clarkson and Shor [13] and Haussler and Welzl [15], we analyze the probability of an ϵ-net as follows.

▶ **Lemma 15.** *With probability $1 - O(\frac{1}{r^5})$, a random sample R of S of size r is an ϵ-net.*

By Lemma 15, we can derive the expected size of $|M(R)^{\nabla}|$. The number of trapezoids in $M(R)^{\nabla}$ is proportional to the number of vertices in $M(R)$, so we analyze the latter. It is sufficient to consider the case in which R is an ϵ-net since $|M(R)|$ is trivially $O(r^3)$ and the probability that R is not an ϵ-net is only $O(r^{-5})$, leading to a product $O(r^{-2})$. We mainly prove that a vertex of $M(R)$ is a vertex of $V_m(S)$ where $k - 7c\frac{n}{r}\log r \le m \le k + 7c\frac{n}{r}\log r + 2$. If this claim holds, since $V_m(S)$ has $O(m(n-m))$ vertices [3], the total complexity of all candidates is proportional to the summation of $m(n-m)$ for $k - 7c\frac{n}{r}\log r \le m \le k + 7c\frac{n}{r}\log r + 2$, leading to $O(\frac{n}{r}k(n-k)\log r + \frac{n^3}{r^2}\log^2 r)$. Since each vertex of $\mathrm{VF}(S)$ is a vertex of $\mathrm{VF}(R)$ with probability $O(\frac{r^3}{n^3})$, this implies that the expected number of vertices in $M(R)$ is $O(\frac{r^2}{n^2}k(n-k)\log r + r\log^2 r)$.

The intuitive idea for the claim is to consider a vertex v of a face F in $M(R)$ and to let F belong to $\mathrm{VR}_j(Q, R)$ and v be an intersection between $J(t, t')$ and $J(t, t'')$. Since F is a face of $M(R)$, there exists a point $x \in F$ such that $x \in \mathrm{FVR}(q, Q)$ and $l_q(x) \le k + 6c\frac{n}{r}\log r$. Since R is an ϵ-net, $l_t(v) - l_q(x) \le c\frac{n}{r}\log r$, implying that $l_t(v) \le k + 7c\frac{n}{r}\log r$. Symmetrically, we have $l_t(v) \ge k + 7c\frac{n}{r}\log r - 2$. Since v is an order-$(l_t(v)+2)$ vertex, we prove the claim and conclude the following lemma.

▶ **Lemma 16.** $E[|M(R)^{\nabla}|] = O(\frac{r^2}{n^2}k(n-k)\log r + r\log^2 r)$.

Chazelle et al. [11] proposed an abstract framework for randomized incremental construction, and to adopt their result, we show that $M(R)$ has some monotonicity property.

▶ **Lemma 17.** *For all trapezoids $\triangle \in M^{\nabla}(R_r)$, if $D(\triangle) \subseteq R_{r-1}$, then $\triangle \in M^{\nabla}(R_{r-1})$.*

Proof. Let F be the face in $M(R_r)$ that contains $\triangle$, i.e., $\triangle \in F^\nabla$, and let F be a face of $\mathrm{VR}_j(Q, R)$. Since $J(\triangle) \cap R_r = \emptyset$ and $D(\triangle) \subseteq R_{r-1}$, $\triangle$ must belong to some face in $\mathrm{VF}(R_{r-1})$, and we use F' to denote it. It is clear that $F' \supseteq F$. If $s_r \in Q$, F' is a face of $\mathrm{VR}_{j-1}(Q \setminus \{s_r\}, R_{r-1})$; otherwise, F' is a face of $\mathrm{VR}_j(Q, R_{r-1})$. In either case, F' dominates F. Since $F \in M^\nabla(R_r)$, F dominates a face C of $V_{k'}(S)$ where $k - 6c\frac{n}{r}\log r \le k' \le k + 6c\frac{n}{r}\log r$, and since F' dominates F, F' also dominates C. Since $\frac{n}{r-1}\log(r-1) > \frac{n}{r}\log r$ (for $r > 2$), $k - 6c\frac{n}{r-1}\log(r-1) \le k' \le k + 6c\frac{n}{r-1}\log(r-1)$, and thus F' belongs to $M(R_{r-1})$, implying that $\triangle \in M^\nabla(R_{r-1})$. ◀

Finally, we analyze the expected construction time. The whole idea is that if R_r is an ϵ-net, we charge the insertion time of s_{r+1} through $M(R_r)$; otherwise, we use $O(r^3 n)$ to bound the insertion time. The latter has been shown to be $O(n)$ over the entire construction. Let T be $\bigcup_{r=10}^{n} M^\nabla(R_r)$. Since each destroyed trapezoid must be created before and $M^\nabla(R_{r+1})$ contains all created trapezoids due to the insertion of s_{r+1} when R_r is an ϵ-net (Lemma 14), the former case during the whole incremental construction is bounded by $O(\sum_{\triangle \in T} \log n + w(\triangle))$. By Lemma 17, we can adopt Chazelle et al.'s framework [11] to prove that

$$E[|T|] = \sum_{r=10}^{n} O(1/r) E[|M(R_r)^\nabla|] \text{ and } E[\sum_{\triangle \in T} w(\triangle)] = \sum_{r=10}^{n} O(n/r^2) E[|M(R_r)^\nabla|].$$

Since $E[|M(R_r)^\nabla|] = O(\frac{r^2}{n^2} k(n-k)\log r + r\log^2 r)$ (by Lemma 16),

$$E[\sum_{\triangle \in T} \log n + w(\triangle))] = \log n \cdot E[|T|] + E[\sum_{\triangle \in T} w(\triangle)] = O(k(n-k)\log^2 n + n\log^3 n).$$

Since we can link all vertices of $V_k(S)$ in $O(k(n-k)\log n)$ time [5], we conclude the following.

▶ **Theorem 18.** *The expected time to compute $V_k(S)$ is $O(k(n-k)\log^2 n + n\log^3 n)$.*

Acknowledgement. We deeply appreciate all valuable comments from the SoCG reviewers. Due to the page limit, we could not address all the comments, but will make the remaining ones clear in the journal version.

References

1 Pankaj K. Agarwal, Mark de Berg, Jirí Matousek, and Otfried Schwarzkopf. Constructing levels in arrangements and higher order Voronoi diagrams. *SIAM J. Comput.*, 27(3):654–667, 1998.

2 Franz Aurenhammer and Otfried Schwarzkopf. A simple on-line randomized incremental algorithm for computing higher order Voronoi diagrams. In *Proceedings of the Seventh Annual Symposium on Computational Geometry (SoCG)*, pages 142–151, 1991.

3 Cecilia Bohler, Panagiotis Cheilaris, Rolf Klein, Chih-Hung Liu, Evanthia Papadopoulou, and Maksym Zavershynskyi. On the complexity of higher order abstract Voronoi diagrams. *Computational Geometry*, 48(8):539–551, 2015.

4 Cecilia Bohler and Rolf Klein. Abstract Voronoi diagrams with disconnected regions. *Int. J. Comput. Geometry Appl.*, 24(4):347–372, 2014.

5 Cecilia Bohler, Chih-Hung Liu, Evanthia Papadopoulou, and Maksym Zavershynskyi. A randomized divide and conquer algorithm for higher-order abstract Voronoi diagrams. In *Proceedings of the 25th International Symposium on Algorithms and Computation (ISAAC)*, pages 27–37, 2014.

6 Jean-Daniel Boissonnat, Olivier Devillers, and Monique Teillaud. A semidynamic construction of higher-order Voronoi diagrams and its randomized analysis. *Algorithmica*, 9(4):329–356, 1993.

7 Timothy M. Chan. Random sampling, halfspace range reporting, and construction of (<= k)-levels in three dimensions. *SIAM J. Comput.*, 30(2):561–575, 2000.

8 Timothy M. Chan and Konstantinos Tsakalidis. Optimal deterministic algorithms for 2-d and 3-d shallow cuttings. In *Proceeding of the 31st International Symposium on Computational Geometry (SoCG)*, pages 719–732, 2015.

9 Bernard Chazelle. Triangulating a simple polygon in linear time. *Discrete & Computational Geometry*, 6:485–524, 1991.

10 Bernard Chazelle and Herbert Edelsbrunner. An improved algorithm for constructing k th-order Voronoi diagrams. *IEEE Trans. Computers*, 36(11):1349–1354, 1987.

11 Bernard Chazelle, Herbert Edelsbrunner, Leonidas J. Guibas, Micha Sharir, and Jack Snoeyink. Computing a face in an arrangement of line segments and related problems. *SIAM J. Comput.*, 22(6):1286–1302, 1993.

12 Kenneth L. Clarkson. New applications of random sampling in computational geometry. *Discrete & Computational Geometry*, 2:195–222, 1987.

13 Kenneth L. Clarkson and Peter W. Shor. Application of random sampling in computational geometry, II. *Discrete & Computational Geometry*, 4:387–421, 1989.

14 Andreas Gemsa, D. T. Lee, Chih-Hung Liu, and Dorothea Wagner. Higher order city Voronoi diagrams. In *Proceedings of the 13th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT)*, pages 59–70, 2012.

15 David Haussler and Emo Welzl. epsilon-nets and simplex range queries. *Discrete & Computational Geometry*, 2:127–151, 1987.

16 Rolf Klein. *Concrete and Abstract Voronoi Diagrams*, volume 400 of *Lecture Notes in Computer Science*. Springer, 1989.

17 Rolf Klein, Elmar Langetepe, and Zahra Nilforoushan. Abstract Voronoi diagrams revisited. *Comput. Geom.*, 42(9):885–902, 2009.

18 Rolf Klein, Kurt Mehlhorn, and Stefan Meiser. Randomized incremental construction of abstract Voronoi diagrams. *Comput. Geom.*, 3:157–184, 1993.

19 D. T. Lee. On k-nearest neighbor Voronoi diagrams in the plane. *IEEE Trans. Computers*, 31(6):478–487, 1982.

20 Chih-Hung Liu and D. T. Lee. Higher-order geodesic Voronoi diagrams in a polygonal domain with holes. In *Proceedings of the Twenty-Fourth Annual Symposium on Discrete Algorithms (SODA)*, pages 1633–1645, 2013.

21 Kurt Mehlhorn, Stefan Meiser, and Ronald Rasch. Furthest site abstract Voronoi diagrams. *Int. J. Comput. Geometry Appl.*, 11(6):583–616, 2001.

22 Ketan Mulmuley. *Computational geometry – an introduction through randomized algorithms*. Prentice Hall, 1994.

23 Evanthia Papadopoulou and Marksim Zavershynskyi. On higher order Voronoi diagrams of line segments. *Algorithmica*, 2014. Published on-line.

24 Edgar A. Ramos. On range reporting, ray shooting and k-level construction. In *Proceedings of the Fifteenth Annual Symposium on Computational Geometry (SoCG)*, pages 390–399, 1999.

25 Robert Endre Tarjan and Christopher J. Van Wyk. An $O(n \log \log n)$-time algorithm for triangulating a simple polygon. *SIAM J. Comput.*, 17(1):143–178, 1988.