Linear-Time Algorithms for the Farthest-Segment Voronoi Diagram and Related Tree Structures⋆

Elena Khramtcova and Evanthia Papadopoulou

Faculty of Informatics, Università della Svizzera italiana (USI), Lugano, Switzerland
{elena.khramtcova, evanthia.papadopoulou}@usi.ch

Abstract. We present linear-time algorithms to construct tree-like Voronoi diagrams with disconnected regions after the sequence of their faces along an enclosing boundary (or at infinity) is known. We focus on the farthest-segment Voronoi diagram, however, our techniques are also applicable to constructing the order-$(k+1)$ subdivision within an order-k Voronoi region of segments and updating a nearest-neighbor Voronoi diagram of segments after deletion of one site. Although tree-structured, these diagrams illustrate properties surprisingly different from their counterparts for points. The sequence of their faces along the relevant boundary forms a Davenport-Schinzel sequence of order ≥ 2. Once this sequence is known, we show how to compute the corresponding Voronoi diagram in linear time, expected or deterministic, augmenting the existing linear-time frameworks for points in convex position with the ability to handle non-point sites and multiple Voronoi faces.

1 Introduction

It is well known that the Voronoi diagram of points in convex position can be computed in linear time, given the order of their convex hull [1]. Linear-time constructions also exist for a class of related diagrams such as the farthest-point Voronoi diagram, computing the medial axis of a convex polygon, and deleting a point from the nearest-neighbor Voronoi diagram. In an abstract setting, a *Hamiltonian abstract Voronoi diagram* can be computed in linear time [9], given the order of Voronoi regions along an unbounded simple curve, which visits each region exactly once and can intersect each bisector only once. This construction has been extended recently to include forest structures [5] under similar conditions where no region can have multiple faces within the domain enclosed by the curve. The medial axis of a simple polygon can also be computed in linear time [8]. It is therefore natural to ask what other types of Voronoi diagrams can be constructed in linear time.

Classical variants of Voronoi diagrams such as higher-order Voronoi diagrams for sites other than points, had surprisingly been ignored in the literature of computational geometry until recently [4, 13]. Given a set S of n simple geometric

---

objects in the plane, called sites, the *order-k Voronoi diagram* of S is a partitioning of the plane into regions such that every point within a region has the same k nearest sites. For $k = 1$, this is the *nearest-neighbor Voronoi diagram* and for $k = n - 1$ it is the *farthest-site Voronoi diagram* of S. Despite similarities, these diagrams for non-point sites, e.g., line segments, illustrate fundamental structural differences from their counterparts for points, such as the presence of disconnected regions (see also [2, 6, 10]). This had been a gap in the computational geometry literature, until recently, as segment Voronoi diagrams are fundamental to problems involving proximity among polygonal objects. This paper contributes further in closing this gap. For more information on Voronoi diagrams see the book of Aurenhammer et al. [3]. For application examples of higher order segment Voronoi diagrams see, e.g., [11] and references therein.

In this paper we give linear-time algorithms (expected and deterministic) for constructing tree-like Voronoi diagrams with disconnected regions, after the sequence of their faces within an enclosing boundary (or at infinity) is known. We focus on the farthest-segment Voronoi diagram, however, the same techniques are applicable to constructing the order-$(k+1)$ subdivision within a given order-k segment Voronoi region, and updating in linear time the nearest-neighbor segment Voronoi diagram after the deletion of one site. Interestingly, the latter two problems require computing initially two different tree-like diagrams. A major difference from the respective problems for points is that the sequence of faces along the relevant enclosing boundary forms a Davenport-Schinzel sequence of order at least two,[1] in contrast to the case of points, where no repetition can exist. Repetition introduces several complications, including the fact that the sequence of Voronoi faces along the relevant boundary for a subset of the original segments, $S' \subset S$, is not a subsequence of the respective sequence for S. In addition, such a subsequence may not even correspond to a Voronoi diagram. Thus, the intermediate diagrams computed by our algorithms are interesting on their own right. They have the structural properties of the relevant segment Voronoi diagram, however, they do not correspond to such a diagram nor are they instances of abstract Voronoi diagrams.

The purpose of this paper is to extend the paradigm of the existing linear constructions for tree-structured diagrams beyond the case of points in convex position [1]. Our goal is to generalize fundamental techniques known for points to more general objects so that the computation of their basic diagrams can be unified, despite their structural differences. As a byproduct we also improve the time complexity of the basic iterative approach to construct the order-k segment Voronoi diagram to $O(k^2n + n \log n)$ from the standard $O(k^2n \log n)$ [13], and also updating a nearest neighbor diagram after deletion of one site in time proportional to the number of updates performed in the diagram.

---

[1] Order-3 for the farthest-segment Voronoi diagram [2, 12], order-4 for the order-k segment Voronoi diagram (easy to derive from [13]), order-2 for disjoint segments or the corresponding abstract Voronoi diagrams [13, 10].

2   Preliminaries and Definitions

Let S be a set of arbitrary line segments in $\mathbb{R}^2$; segments in S may intersect or touch at a single point. The distance between a point q and a line segment s_i is $d(q, s_i) = \min\{d(q, y) \mid y \in s_i\}$, where $d(q, y)$ denotes the ordinary distance between two points q, y in the L_2 (or the L_p) metric. The bisector of two segments $s_i, s_j \in S$ is $b(s_i, s_j) = \{x \in \mathbb{R}^2 \mid d(x, s_i) = d(x, s_j)\}$. For disjoint segments, $b(s_i, s_j)$ is an unbounded curve that consists of a constant number of pieces, where each piece is a portion of an elementary bisector between the endpoints and open portions of s_i, s_j. If two segments intersect at point p, their bisector consists of two such curves intersecting at p.

The farthest Voronoi region of a segment s_i is $freg(s_i) = \{x \in \mathbb{R}^2 \mid d(x, s_i) > d(x, s_j), 1 \leq j \leq n, j \neq i\}$. For disjoint line segments or line segments that intersect but do not touch at endpoints, the order-k Voronoi region of a set H, where $H \subset S$, $|H| = k$, and $1 \leq k \leq n - 1$, is $k\text{-}reg(H) = \{x \mid \forall s \in H, \forall t \in S \setminus H \ d(x, s) < d(x, t)\}$. For an extension of this definition to line segments forming a *planar straight-line graph*, see [13]. Note, for $k = n - 1$, $freg(s_i) = k\text{-}reg(S \setminus \{s_i\})$. The (non-empty) farthest (resp., order-k) Voronoi regions of the segments in S, together with their bounding edges and vertices, define a partition of the plane, called the *farthest-segment Voronoi diagram*, denoted $\text{FVD}(S)$, see Fig. 1(a) (resp., *order-k Voronoi diagram*). Any maximally connected subset of a Voronoi region is called a *face*.
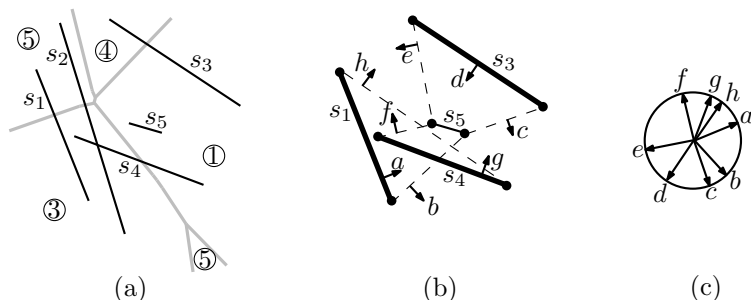

Fig. 1. [12] (a) FVD(S), $S = \{s_1, \ldots, s_5\}$; (b) its farthest hull; (c) Gmap(S)

A farthest Voronoi region $freg(s_i)$ is non-empty and unbounded in direction ϕ if and only if there exists an open halfplane, normal to ϕ, which intersects all segments in S but s_i [2]. The line ℓ, normal to ϕ, bounding such a halfplane, is called a *supporting line*. The direction ϕ (normal to ℓ) is referred to as the *hull direction* of ℓ and it is denoted by $\nu(\ell)$. An unbounded Voronoi edge separating $freg(s_i)$ and $freg(s_j)$ is a portion of $b(p, q)$, where p, q are endpoints of s_i and s_j, such that the line through $\overline{pq}$ induces an open halfplane that intersects all segments in S, except s_i, s_j (and possibly except additional segments incident to p, q). Segment $\overline{pq}$ is called a *supporting segment*; the direction normal to it

pointing to the inside of this halfplane is denoted by $\nu(\overline{pq})$ and is called the *hull direction* of $\overline{pq}$. A segment $s_i \in S$ such that the line ℓ through s_i is supporting, is called a *hull segment*; its *hull direction* is $\nu(s_i) = \nu(\ell)$, normal to ℓ. The closed polygonal line obtained by following the supporting and hull segments in the angular order of their hull directions is called the *farthest hull*. Figs. 1(a) and 1(b) illustrate a farthest-segment Voronoi diagram and its hull respectively. In Fig. 1(b), supporting segments are shown in dashed lines, and hull segments are shown in bold. Arrows indicate the hull directions of all supporting and hull segments. For more information see [12].

The Gaussian map of FVD(S), denoted Gmap(S), (see Fig. 1(c)) provides a correspondence between the faces of FVD(S) and a *circle of directions K* [12]. K can be assumed to be a unit circle, where each point x on K corresponds to a direction as indicated by the radius of K at x. Each Voronoi face is mapped to an arc on K, which represents the set of directions along which the face is unbounded. An arc is delimited by two consecutive hull directions of supporting segments. The Gmap(S) can be viewed as a cyclic sequence of consecutive arcs on K, where each arc corresponds to one face of FVD(S). Two neighboring arcs α, γ are separated by the hull direction $\nu(\alpha, \gamma)$ of a supporting segment $\overline{pq}$ ($\nu(\alpha, \gamma) = \nu(\overline{pq})$); $\nu(\alpha, \gamma)$ is the direction towards infinity of the relevant portion of bisector $b(p, q)$. The arc of a hull segment is called a *segment arc* and consists of two sub-arcs separated by the hull direction ν of the segment, where each sub-arc corresponds to an endpoint of the hull segment. An arc that corresponds to a single endpoint of a segment is called a *single-vertex* arc. The Gmap(S) can be computed in $O(n \log n)$ time (or output-sensitive $O(n \log h)$ time, where $h = |\text{Gmap}(S)|$) [12].

The standard point-line duality transformation T offers a correspondence between the faces of FVD(S) and envelopes of *wedges* [2]. A segment $s_i = uv$ corresponds to a *lower wedge*, defined by the lower envelope of $T(u)$ and $T(v)$ (see, e.g., Fig. 5), and to an *upper wedge* defined as the area above the upper envelope of $T(u)$, $T(v)$. Let E (resp., E') be the boundary of the union of the lower (resp., upper) wedges. The faces of FVD(S) correspond exactly to the edges of E and E' [2]. Let the upper and lower Gmap be the portion of Gmap(S) above and below the horizontal diameter of K respectively.

There is a clear correspondence between E (resp., E') and the upper (resp., lower) Gmap: the vertices of E are exactly the hull directions of supporting segments on the upper Gmap and the apexes of wedges in E are exactly the hull directions of hull segments [12]. In fact, any x-monotone path π in the arrangement of upper (resp., lower) wedges can be transformed into a sequence of arcs in the portion of K above (resp., below) its horizontal diameter. Each edge of π, portion of $T(u)$, corresponds to an arc on K for u, and each vertex of π, which is an intersection point $T(u) \cap T(v)$, corresponds to the hull direction $\nu(\overline{uv})$ of the supporting segment $\overline{uv}$.

Throughout this paper, given an arc α, let s_α denote the segment in S that induces α.

3   The Farthest Voronoi Diagram of a Sequence

Let G be a sequence of arcs on the circle of directions K, corresponding to a pair of x-monotone paths in the dual space, one in the arrangement of upper (resp. lower) wedges. No arcs in the sequence can overlap and no gaps are allowed. We call G an *arc sequence*. Consecutive arcs of the same segment in G are assumed unified into a single maximal arc.

In the following we define the farthest Voronoi diagram of such an arc sequence G, $FVD(G)$. For $G = Gmap(S)$, $FVD(G) = FVD(S)$. The diagrams of such sequences appear as intermediate diagrams in the process of computing $FVD(S)$, however, they do not correspond to any type of segment Voronoi diagram. We first define such a diagram and then present an arc deletion and arc insertion operation, which constitute the basis for our algorithms.

Given an arc $\alpha \in G$ and a point $x \in \mathbb{R}^2$, $x \not\in s_\alpha$, let $r(x, s_\alpha)$ denote the ray emanating from x in the direction $\overrightarrow{px}$, where p is the point in s_α closest to x (see Fig. 2). We say that x is *attainable* from α if the direction of $r(x, s_\alpha)$ is contained in α. A point x in the interior of s_α is attainable from α if $\nu(s_\alpha)$ is in α (i.e., if α is a segment arc). An endpoint of s_α is attainable from all its corresponding arcs (see Sec. 2).

Let $d(x, \alpha) = d(x, s_\alpha)$, if x is attainable from α, and let $d(x, \alpha) = -\infty$, otherwise. The locus of points attainable from arc α is called the *attainable region* of α, $R(\alpha)$. Fig. 2 illustrates the attainable regions of arcs α_1, α_2, and β, shaded. Intuitively, an arc α *exists* only for points within its attainable region (i.e., α is relevant exclusively within $R(\alpha)$ and it should not be considered outside).

Remark 1. For arcs $\alpha_1, \alpha_2 \in G$ of the same segment s_α, $R(\alpha_1) \cap R(\alpha_2) \setminus \{s_\alpha\} = \emptyset$.

Given two arcs α, β ($s_\alpha \neq s_\beta$) we define their *arc bisector* by $b(\alpha, \beta) = b(s_\alpha, s_\beta) \cap R(\alpha) \cap R(\beta)$. If $s_\alpha = s_\beta$ and α, β are consecutive, then $b(\alpha, \beta) = R(\alpha) \cap R(\beta)$ is called the *artificial bisector* of α, β. The farthest Voronoi region of an arc α is now defined in the ordinary way

$$freg(\alpha) = \{x \in \mathbb{R}^2 \mid d(x, \alpha) > d(x, \gamma), \forall \text{ arc } \gamma \in G, \gamma \neq \alpha\}.$$

The subdivision of the plane derived by the farthest regions of all arcs in G and their boundaries, is called the *farthest Voronoi diagram* of G, denoted $FVD(G)$. The closure of $freg(\alpha)$ is denoted by $\overline{freg}(\alpha)$.

Definition 1. *Let $\mathcal{T}(G) = \mathbb{R}^2 \setminus \cup_{\alpha \in G} freg(\alpha)$. If all edges of $\mathcal{T}(G)$ are portions of arc bisectors, then G, $\mathcal{T}(G)$, and $FVD(G)$ are all called* proper.

For a proper sequence, $\mathcal{T}(G)$ is simply the graph structure of $FVD(G)$. The diagrams and sequences produced by our algorithms are always proper. Note, however, that for an arbitrary arc sequence, $\mathcal{T}(G)$ may contain boundaries of attainable regions and even two-dimensional regions. Fig. 3(a) illustrates $FVD(G)$ for a proper arc sequence G, which consists of three maximal arcs of segments s_1, s_4, and s_5 and is derived from $Gmap(S)$ of Fig. 1. Ray r indicates an *artificial bisector* between two consecutive arcs of s_5 (which have been unified into a single maximal arc for s_5). Fig. 3(b) illustrates $FVD(G'')$, where G'' contains an additional arc β of segment s_3 ($G'' = G \oplus \beta$).
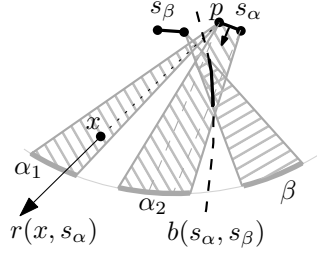
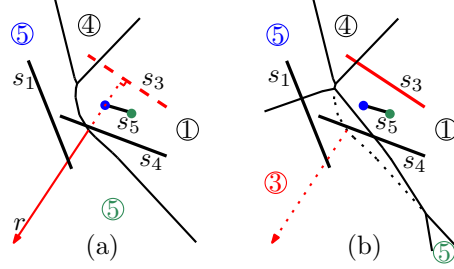**Fig. 2.** Attainable regions $R(\alpha_1), R(\alpha_2), R(\beta)$

Fig. 3. FVD(G) for an arc sequence of Fig. 1; (a) FVD(G); (b) FVD(G''), $G'' = G \oplus \beta$.

Lemma 1. *For a proper arc sequence G, $\mathcal{T}(G)$ is a tree.*

Proof (Sketch). Since G is proper, all the edges of $\mathcal{T}(G)$ are portions of arc bisectors. Let x be a point on $\mathcal{T}(G)$ along arc bisector $b(\alpha, \beta)$. We first prove that the entire ray $r(x, s_\alpha)$ must be enclosed in $\overline{freg}(\alpha)$, i.e., regions are unbounded. This is because no arc bisector involving α can bound $r(x, s_\alpha)$ as we walk on it starting at x, unless an arc δ suddenly becomes attainable because $r(x, s_\alpha)$ intersects $R(\delta)$ at point z and $d(z, \delta) > d(z, \alpha)$; but then $z \in \mathcal{T}(G)$ without being on an arc bisector, a contradiction. It remains to show that $\mathcal{T}(G)$ is connected. If $\mathcal{T}(G)$ contained two different components, there would be a face of a segment s_α inducing two non consecutive arcs in G, α_1 and α_2. But then $freg(\alpha_1)$ and $freg(\alpha_2)$ would be neighboring, contradicting Remark 1. □

An arc sequence G is called a *subsequence* of Gmap(S) if every arc of G entirely contains a corresponding arc of Gmap(S) induced by the same segment. The arcs in G are simply expanded versions of the arcs in Gmap(S). The arcs in Gmap(S) as well as their expanded versions in G are called *original arcs*. A sequence G' is called an *augmented subsequence* of Gmap(S) if G' contains at least one arc of Gmap(S) for every segment with an arc in G'. An augmented subsequence consists of *original* arcs, which are expanded versions of the arcs in Gmap(S), and *new* arcs, which do not correspond to arcs of Gmap(S). An augmented subsequence G', which has the same original arcs as G, is said to be *corresponding to G*. Note that in the dual space, G and G' no longer correspond to envelopes of wedges, but to x-monotone paths that contain portions of these envelopes. The intermediate sequences of diagrams produced by our algorithms are always augmented subsequences of Gmap(S).

3.1 Deletion and insertion of arcs

Throughout our algorithms we use a deletion and a re-insertion operation for original arcs in sequences derived from Gmap(S). The deletion operation produces subsequences of Gmap(S) that are not necessarily proper. As a result, the insertion operation introduces new arcs, creating augmented subsequences, which are always proper. Let $G \ominus \beta$ (resp., $G \oplus \beta$) denote the arc sequence derived from G after deleting from it (resp., inserting to it) arc β.
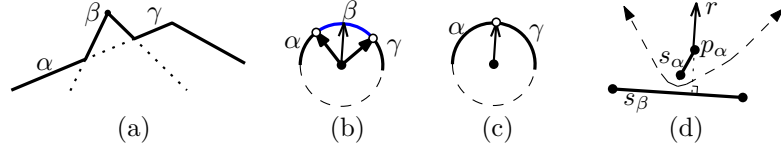
Fig. 4. Sequence $\alpha\beta\gamma$, $s_\alpha = s_\gamma$. (a) The dual wedges; (b) G; (c) $G \ominus \{\beta\}$; (d) The artificial bisector $b(\alpha, \gamma) = r$; the dashed curve indicates $b(s_\alpha, s_\beta)$.

Arc deletion. A subsequence G is derived from $\mathrm{Gmap}(S)$ by deleting arcs. When an arc β is deleted from G, the neighboring arcs α and γ *expand* over β (see Fig. 4(a)-(c)). Either both α and γ expand (see Fig. 4 and Fig. 5(a) illustrating segments in the dual space) or one expands while the other shrinks (see Fig. 5(b)). During the expansion, α and γ may change from being a single-vertex arc to a segment arc. Since α and γ are original, they both remain present in $G \ominus \{\beta\}$, and their common endpoint becomes $\nu(\alpha, \gamma)$. Assuming $s_\alpha \neq s_\gamma$, $\nu(\alpha, \gamma)$ corresponds to bisector $b(\alpha, \gamma)$ as obtained from $b(s_\alpha, s_\gamma)$. If $s_\alpha = s_\gamma$, we let α and γ expand until they reach $\nu(s_\beta)$, i.e., $\nu(\alpha, \gamma) = \nu(s_\beta)$. If $s_\alpha = s_\beta$ then α expands to cover the entire β and $\nu(\alpha, \gamma) = \nu(\beta, \gamma)$.

Remark 2. The artificial bisector $b(\alpha, \gamma)$ (for $s_\alpha = s_\gamma$) is (or contains) a ray perpendicular to s_β, emanating from the relevant endpoint of s_α and extending away from s_β (see Fig. 4(d)).
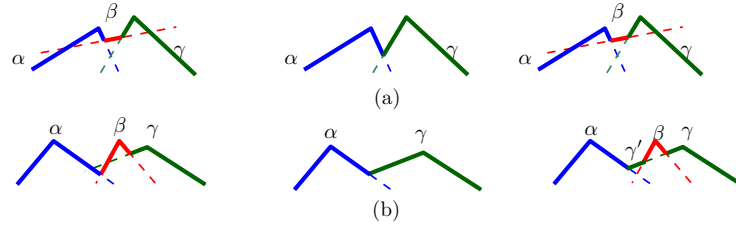

Fig. 5. Deleting and re-inserting β in sequence $\alpha\beta\gamma$. (a) α and γ enlarge; (b) γ enlarges, α shrinks. From left to right: the initial sequence; after deleting β; after re-inserting β.

Arc insertion. Let G' be a proper augmented subsequence of $\mathrm{Gmap}(S)$ and let β be an original arc, $\beta \notin G'$. Let α, γ be two consecutive original arcs in G', such that β is between α, γ in $\mathrm{Gmap}(S)$. A number of new arcs may lie between α, γ in G'. To insert β in G' there are several cases to consider. The insertion of arc β in G' corresponds to inserting $freg(\beta)$ in $\mathrm{FVD}(G')$. Fig. 5 illustrates in dual space the deletion and re-insertion of an arc β in a sequence $\alpha\beta\gamma$.

Basic cases are as follows (assuming for simplicity that α, γ are consecutive in G'): (1) s_α, s_β, and s_γ are all distinct, and $\nu(\alpha, \gamma)$ is in β. This is the standard case, resulting in $\alpha\beta\gamma$, see Fig. 5(a). (2) $s_\alpha = s_\gamma$. Then β is inserted over $\nu(s_\beta) = \nu(\alpha, \gamma)$, resulting in $\alpha\beta\gamma$, and $freg(\beta)$ is inserted over the artificial bisector $b(\alpha, \gamma)$

in FVD(G'), see Fig. 3. (3) Arc γ (equiv. α), as it appears expanded in G', entirely contains β, see Fig. 5(b) (note that α had shrunk during the deletion of β). Then the insertion of β splits γ in two arcs resulting in $\alpha\gamma'\beta\gamma$, where γ' is a new arc. In FVD(G'), $freg(\beta)$ splits $freg(\gamma)$ into $freg(\gamma)$ and $freg(\gamma')$. (4) $s_\alpha = s_\beta$ (equiv. $s_\beta = s_\gamma$). Then α is split by $\nu(\alpha,\beta)$ and one part becomes β. Note that $\nu(\alpha,\beta)$ is determined when α and β became consecutive in a deletion operation, and that α,β cannot be neighbors in Gmap(S).

If α and γ are not consecutive in G', a number of new arcs may be traced to find the actual entry point for β between α and γ. The insertion of β may delete a series of such consecutive new arcs. Assuming that G' is proper, it is not hard to show that $G'' = G' \oplus \beta$ is also a proper augmented subsequence of Gmap(S).
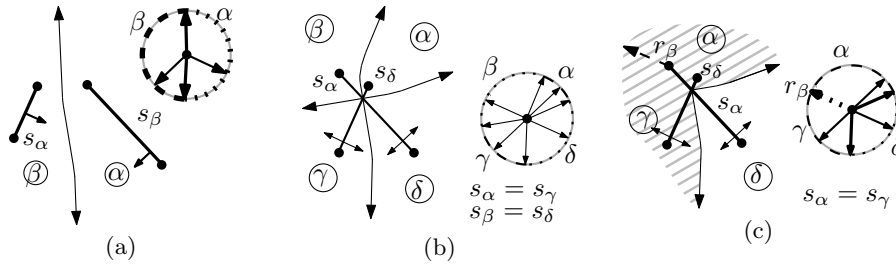

Fig. 6. FVD(G) and G. (a) $G =$ Gmap(S) for $S = \{s_\alpha, s_\beta\}$; (b) $G =$ Gmap(S) for $S = \{s_\alpha, s_\delta\}$; (c) $G = \alpha\gamma\delta =$ Gmap(S) $\ominus \beta$ where $S = \{s_\alpha, s_\delta\}$

Note that arc sequences defined by two segments are always proper. Fig. 6 illustrates such sequences and their Voronoi diagrams. Figs 6(a) and (b) show FVD(S) and Gmap(S) for two disjoint and intersecting segments respectively. Fig 6(c) illustrates FVD(G) for $G =$ Gmap(S) $\ominus \beta$, where S is the same as in Fig. 6(b). In the latter figure, arcs α and γ ($s_\alpha = s_\gamma$) become neighbors inducing one maximal arc $\alpha\gamma$; region $freg(\alpha\gamma)$ is shown shaded; it is split into $freg(\alpha)$ and $freg(\gamma)$ by the artificial bisector $b(\alpha,\gamma) = r_\beta \cup s_\alpha$.

4 A Randomized Linear Construction

We sketch an expected linear-time algorithm to compute FVD(S), given Gmap(S). It is inspired by the simple two-phase randomized approach of [7] for points in convex position and uses the concepts of Sec. 3. Let $\alpha_1, \alpha_2, \ldots, \alpha_h$ be a random permutation of arcs in Gmap(S), and let $A_i = \{\alpha_1, \alpha_2, \ldots, \alpha_i\}$, $1 \leq i \leq h$, be the set of the first i arcs in this order. Let t be the largest index such that $\alpha_1, \ldots, \alpha_t$ consists of arcs of only two segments that form exactly two maximal arcs.

The algorithm proceeds in two phases. Phase 1 computes the subsequence G_i, $t \leq i < h$, where $G_h =$ Gmap(S), and G_i is obtained from G_{i+1} by deleting arc α_{i+1} as described in Sec. 3. The two neighbors of α_{i+1} in G_{i+1} are recorded as a tentative re-entry point for α_{i+1} during phase 2. Note that both neighbors may correspond to the same segment or the segment of one neighbor may coincide with $s_{\alpha_{i+1}}$. In phase 2, the algorithm computes incrementally G_i' and FVD(G_i'),

for $t < i \leq h$, starting with $\mathrm{FVD}(G'_t)$, $G'_t = G_t$. G'_t is proper as it consists of exactly two maximal arcs. G'_{i+1} is obtained from G'_i by inserting back α_{i+1} ($G'_{i+1} = G'_i \oplus \alpha_{i+1}$). During the re-entry of α_{i+1} a *new* arc may be created, thus, $G'_{i+1} \neq G_{i+1}$. The entry point for α_{i+1} is either an unbounded bisector (regular or artificial) or an arc σ that entirely contains α_{i+1}. In the latter case, a new arc is created. At the end of phase 2 we obtain $\mathrm{FVD}(G'_h) = \mathrm{FVD}(S)$ ($G'_h = G_h$).

In the full paper we prove: (1) the complexity of G'_i is $O(i)$ despite the new arcs; (2) the expected number of new arcs traced in a step of phase 2 is constant. Then using backwards analysis we can derive the following theorem.

Theorem 1. *Given* $\mathrm{Gmap}(S)$*, the* $\mathrm{FVD}(S)$ *can be computed in expected* $O(h)$ *time, where h is the complexity of* $\mathrm{FVD}(S)$*.*

5   A Deterministic Linear Divide-and-Conquer Algorithm

We now augment the framework of Aggarwal et al. [1] for points in convex position with techniques from Secs. 3, 4, and derive a linear-time algorithm to compute $\mathrm{FVD}(S)$, given $\mathrm{Gmap}(S)$. Let G be a subsequence of $\mathrm{Gmap}(S)$, and let G' be a corresponding proper augmented subsequence such that the complexity of G' is $O(|G|)$, where $|G|$ denotes the number of arcs of the sequence G. Our algorithm follows the flow of [9], which in turn follows [1].

1. Unite consecutive arcs of the same segment in G into single maximal arcs.
2. Color each arc of G *red* or *blue* by applying the following two rules:
   (a) For each 5-tuple F of consecutive arcs $\alpha\beta\gamma\delta\epsilon$ in G, compute $\mathrm{FVD}(F')$ as follows: start with the sequence $\gamma\delta$, and consecutively insert the arcs β, ϵ, α (in this order) resulting in $\mathrm{FVD}(F')$. (F' is a possibly augmented version of F.) In $\mathrm{FVD}(F')$, if $freg(\gamma)$ does not neighbor any region of segments s_α and s_ϵ, color γ red; else color γ blue.
   (b) For each series of consecutive blue arcs, color red every other arc, except the last one.
3. Let B (blue) be the sequence obtained from G by deleting all the red arcs. Recursively compute $\mathrm{FVD}(B')$. (B' is a possibly augmented version of B.)
4. Partition the red arcs into *crimson* and *garnet*: Re-color as *crimson* at least a constant fraction of the red arcs, such that for any two crimson arcs, if they were inserted in $\mathrm{FVD}(B')$, their Voronoi regions would not touch.
5. Insert the crimson arcs one by one in $\mathrm{FVD}(B')$ resulting in $\mathrm{FVD}(V')$.
6. Let Gr (garnet) be the sequence obtained from G by deleting all blue and crimson arcs. Recursively compute $\mathrm{FVD}(Gr')$.
7. Merge $\mathrm{FVD}(V')$ and $\mathrm{FVD}(Gr')$ into $\mathrm{FVD}(G')$ so that $|G'|$ is $O(|G|)$.
8. For any arcs united in Step 1, subdivide their regions in $\mathrm{FVD}(G')$ into finer parts by inserting the corresponding artificial bisectors.

The recursion ends when the number of maximal arcs in G is at most five. Then $\mathrm{FVD}(G')$ can be directly computed in $O(1)$ time and also enhanced as indicated in Step 8. If all arcs in G are of the same segment, no diagram is generated but instead G is returned as a list of arcs. In this case, in Step 7, we obtain $\mathrm{FVD}(G')$ by inserting this list of arcs in $\mathrm{FVD}(V')$ one by one.

Step 2. Rules 2a and 2b guarantee that no two consecutive arcs in G are red and no three consecutive arcs in G are blue. The insertion order in Rule 2a guarantees that γ neighbors at most one new arc.

Step 4. To choose the crimson arcs we apply the *combinatorial lemma* of [1] on (a modified) $\mathcal{T}(B')$. The lemma states that for a binary tree T with n leaves embedded in $\mathbb{R}^2$, if each leaf of T is associated with a subtree of T and if for any two successive leaves these subtrees are disjoint, then in $O(n)$ time we can choose a set of leaves, whose number is at least a constant fraction of n and whose subtrees are pairwise disjoint. We associate each red arc β in G with a unique leaf of $\mathcal{T}(B')$, which would be the entry point for β in FVD(B'). If the insertion of β splits an arc of B' in two, then we also add an artificial bisector to $\mathcal{T}(B')$ to serve as an entry point for β. The leaf in $\mathcal{T}(B')$ associated with β is in turn associated with the incident subtree of $\mathcal{T}(B')$, which would be intersected by *freg*(β), if β were inserted in FVD(B'). The modified $\mathcal{T}(B')$ satisfies the requirements of the combinatorial lemma, and has complexity proportional to $|B'|$ plus the number of red arcs $|R|$.

Step 7. We obtain G' and FVD(G') by merging FVD(V') and FVD(Gr'). To keep the complexity of G' within $O(|G|)$, we merge the two diagrams while discarding parts that are guaranteed to contain no original arcs. Merging is done in two steps: (1) identify starting points for the *merge curves* between the two diagrams, and (2) trace the *merge curves*. Here, we identify starting points only for the merge curves that are related to original arcs. Skipping a merge curve has the effect of discarding the portion of one diagram that is bounded by it. This can be safely done because any portions of the diagram that are associated with only new arcs can not appear in FVD(S). G' contains all the original arcs of V' and Gr'; furthermore, $|G'|$ is $O(|V'| + |Gr'|)$. Since G' contains all the original arcs of G, it is an augmented subsequence of Gmap(S) corresponding to G. It is not hard to prove that G' is proper.

G' is an augmented subsequence of Gmap(S) corresponding to G, and the recursive algorithm starts with $G = $ Gmap(S). Thus, at the end of the algorithm, the resulting arc sequence must be $G' = $ Gmap(S) (easy to see in dual space).

Lemma 2. $|G'|$ *is* $O(|G|)$.

Proof. Let $m = |G|$ and $S(m) = |G'|$. Since Step 4 is performed by applying the combinatorial lemma of [1], $|Gr| \leq q|R|$, where $0 < q < 1$ and $|R|$ is the number of red arcs ($|R| = |G| - |B|$). Thus, (following [1, 9]) there exist positive constants q_1 and q_2, $q_1 + q_2 < 1$, such that $|B| \leq q_1|G|$ and $|Gr| \leq q_2|G|$. At Step 4, at most one new arc is generated for every crimson arc inserted in B', thus, $|V'| = S(q_1m) + O(m)$. At Step 7, $|G'| \leq |V'| + |Gr'| + O(m)$. Thus, $|G'| \leq S(q_1m) + S(q_2m) + O(m)$. Hense, $S(m) = O(m)$. □

Since the size of the augmented subsequences is always kept bounded, the time complexity can be analyzed similarly to [1]. We conclude:

Theorem 2. *Given* Gmap(S), *the* FVD(S) *can be computed in* $O(h)$ *time, where h is the combinatorial complexity of* FVD(S).

Concluding Remarks

Theorems 1 and 2 apply also to computing the order-$(k+1)$ subdivision within an order-k Voronoi region in time proportional to the complexity of the region's boundary. It also applies to updating a nearest-neighbor segment Voronoi diagram after the deletion of one segment in time proportional to the number of updates in the diagram. In this paper we considered line segments, however, the presented techniques are not specific to them. For example, the constructions can be easily adapted for the respective farthest abstract Voronoi diagram (to be described in the full paper). Note that the farthest abstract Voronoi diagram can be constructed in expected $O(n \log n)$ time by a randomized incremental construction [10], which is not related to the randomized linear-time approach in this paper.

References

1. Aggarwal, A., Guibas, L., Saxe, J., Shor, P.: A linear-time algorithm for computing the Voronoi diagram of a convex polygon. Discrete Comput. Geom. 4, 591–604 (1989)
2. Aurenhammer, F., Drysdale, R., Krasser, H.: Farthest line segment Voronoi diagrams. Inform. Process. Lett. 100, 220–225 (2006)
3. Aurenhammer, F., Klein, R., Lee, D.T.: Voronoi Diagrams and Delaunay Triangulations. World Scientific (2013)
4. Bohler, C., Cheilaris, P., Klein, R., Liu, C.H., Papadopoulou, E., Zavershynskyi, M.: On the complexity of higher order abstract Voronoi diagrams. In: Proc. 40th ICALP. LNCS, vol. 7965, pp. 208–219 (2013)
5. Bohler, C., Klein, R., Liu, C.: Forest-like abstract Voronoi diagrams in linear time. In: Proc. 26th CCCG (2014)
6. Cheong, O., Everett, H., Glisse, M., Gudmundsson, J., Hornus, S., Lazard, S., Lee, M., Na, H.: Farthest-polygon Voronoi diagrams. Comput. Geom. 44(4), 234–247 (2011)
7. Chew, L.P.: Building Voronoi diagrams for convex polygons in linear expected time. Tech. rep., Dartmouth College, Hanover, USA (1990)
8. Chin, F., Snoeyink, J., Wang, C.A.: Finding the medial axis of a simple polygon in linear time. Discrete Comput. Geom. 21(3), 405–420 (1999)
9. Klein, R., Lingas, A.: Hamiltonian abstract Voronoi diagrams in linear time. In: 9th ISAAC. LNCS, vol. 834, pp. 11–19 (1994)
10. Mehlhorn, K., Meiser, S., Rasch, R.: Furthest site abstract Voronoi diagrams. Int. J. Comput. Geom. Ap. 11(6), 583–616 (2001)
11. Papadopoulou, E.: Net-aware critical area extraction for opens in VLSI circuits via higher-order Voronoi diagrams. IEEE T. Comput. Aid. D. 30(5), 704–716 (2011)
12. Papadopoulou, E., Dey, S.K.: On the farthest line-segment Voronoi diagram. Int. J. Comput. Geom. Ap. 23(6), 443–459 (2013)
13. Papadopoulou, E., Zavershynskyi, M.: The higher-order Voronoi diagram of line segments. Algorithmica DOI 10.1007/s00453-014-9950-0 (2014)