Forest-Like Abstract Voronoi Diagrams in Linear Time*

Cecilia Bohler, Rolf Klein, and Chih-Hung Liu †

Abstract

Voronoi diagrams are a well-studied data structure of proximity information, and although most cases require $\Omega(n \log n)$ construction time, it is interesting and useful to develop linear-time algorithms for certain Voronoi diagrams. For example, the Voronoi diagram of points in convex position, and the medial axis and constrained Voronoi diagram of a simple polygon are a tree or forest structure and can be computed in linear time. In order to provide a more general approach, we study abstract Voronoi diagrams in a domain where each site has a unique face touching the boundary of the domain, implying that the diagram is a forest-like structure, and develop a linear-time algorithm. Since abstract Voronoi diagrams are a category of Voronoi diagrams, our algorithm works for many concrete Voronoi diagrams.

1 Introduction

Voronoi diagrams [2, 3, 9] are a well-studied data structure of proximity information, used in many different engineering and science applications [4, 19]. In principle, given a set S of sites on the plane, the Voronoi diagram is a planar subdivision such that all points in a region share the same nearest site in S. Sites can be points, line segments, circles, polygons, and so on, and the distance measure can be the Euclidean distance, L_p norms, convex distance function, geodesic distance, and so on. There have been $O(n \log n)$-time construction algorithms for many concrete cases by randomized incremental construction, divide-and-conquer paradigm, or plane-sweep method.

In order to provide a unifying construction method, Klein [10] introduced *Abstract Voronoi Diagrams* (AVDs, for short). Here, no sites, circles, or distance measures are given. Instead, for each pair of sites p and q from S one takes an unbounded curve $J(p, q) = J(q, p)$ as primary object, together with the open domains $D(p, q)$ and $D(q, p)$ it separates. Abstract Voronoi regions are defined by

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

and the abstract Voronoi diagram by

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

The following axioms were required to hold for each subset S' of S.

(A1) *Each curve $J(p, q)$, where $p \neq q$, is unbounded. After stereographic projection to the sphere, it can be completed to a closed Jordan curve through the north pole.*

(A2) *Each nearest Voronoi region $VR(p, S')$ is nonempty and pathwise connected.*

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

The abstract Voronoi diagram can be constructed in $O(n \log n)$ steps by the divide-and-conquer paradigm [10] and the randomized incremental construction [15].

However, certain practical applications require only a specific substructure of the entire diagram or a special kind of Voronoi diagram. Although the construction time is $\Omega(n \log n)$ for many kinds of Voronoi diagrams, it is still possible to compute a specific part or a special case faster. Aggarwal et al. [1] developed a linear-time algorithm for Euclidean Voronoi diagrams of points in convex position. Their algorithm further allows to delete a site from Euclidean Voronoi diagrams in time linear to the structural changes, and also speeds up the algorithm for the k^{th}-order Voronoi diagram in [17] by a $O(\log n)$ factor.

Later, Klein and Lingas [12] generalized their idea to abstract Voronoi diagrams where a Hamiltonian path passing each bisecting curve exactly once is given called *Hamiltonian abstract Voronoi diagrams*, and proposed a linear-time algorithm. For all subsets S' of S such a Hamiltonian path runs through each Voronoi region of $V(S')$ exactly once.

Moreover, there are two kinds of Voronoi diagrams of a simple polygon, both consisting of a tree or forest structure, who have received considerable attention. First, the medial axis of a simple polygon is the Voronoi

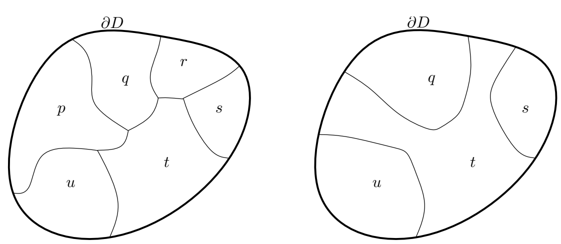
*This work was supported by the European Science Foundation (ESF) in the EUROCORES collaborative research project EuroGIGA/VORONOI. Chih-Hung Liu was supported by Alexander von Humboldt-Foundation, Germany.

†University of Bonn, Institute of Computer Science I, Germany, bohler@cs.uni-bonn.de, rolf.klein@uni-bonn.de, chliu_10@citi.sinica.edu.tw

diagram of its polygonal edges. Lee [16] first proposed an $O(n \log n)$-time algorithm for this medial axis, and Chin et al. [6] later developed a linear-time algorithm. Second, the constrained Voronoi diagram of a simple polygon is the Voronoi diagram of its polygonal vertices constrained by its polygonal edges. Lee and Lin [18] first derived an $O(n \log n)$-time algorithm for these kinds of constrained Voronoi diagrams, and then Klein and Lingas [13] proposed a linear-time algorithm in the L_1 metric. Furthermore, in the Euclidean metric, Klein and Lingas [13] later developed a randomized linear-time algorithm, and Chin and Wang [7] finally gave a deterministic linear-time algorithm.

However, the convex position is not applicable for many other geometric objects and other distance measures. Furthermore, computing a Hamiltonian path for a given AVD is NP-complete, which we will prove by Corollary 4 in Section 2. Besides, those linear-time algorithms [6,7,14] for the medial axis and the constrained Voronoi diagram of a simple polygon depend on a decomposition of a simple polygon, which prevents them from being extended to a more general setting.

Therefore, we consider the abstract Voronoi diagram in a domain where its structure is a forest and each of its sites has exactly one face, see Figure 1. Let $D \subseteq \mathbb{R}^2$ be a bounded domain, e. g. a domain bounded by Γ, where Γ is a simple closed curve intersecting each bisector exactly twice such that no two bisectors intersect in a connected component entirely enclosed by the outer domain of Γ. In the following, without explicit indication, $V(S')$ means $V(S') \cap D$ and $\mathrm{VR}(p, S')$ means $\mathrm{VR}(p, S') \cap D$.



(a) $V(S)$ (b) $V(S')$

Figure 1: (a) Abstract Voronoi diagram $V(S)$ in a domain D, the ordering of the regions along ∂D is p, q, r, s, t, u.
(b) For a subset $S' \subset S$, $V(S')$ may be a forest, the ordering of the regions along ∂D is here t, q, t, s, t, u.

We require the additional axiom:

(A4) *$V(S)$ is a tree. For all $S' \subseteq S$, $V(S')$ is a forest and each Voronoi region has exactly one face.*

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

Theorem 1 *Given a domain D together with the ordering of the Voronoi regions along ∂D we can compute $V(S)$ in time $O(n)$.*

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

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

That is why in this paper we chose a different definition than in [12]. Compared to this algorithm for Hamiltonian abstract Voronoi diagrams our algorithm has two major differences in the coloring (Section 4.1) and selection (Section 4.2), and we prove the corresponding theoretical properties for the correctness. For the coloring, our algorithm needs to consider two more subcases, and two consecutive sites in the sequence can be both colored red, while no consecutive sites in [12] are colored red. For the selection, our algorithm needs to modify $V(S')$ into a tree for applying Aggarwal's selecting lemma [1].

A preliminary version of this work appeared in [5].

2 NP-completeness

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

Let $V(S)$ denote an arbitrary AVD. In [10], Theorem 2.7.3, it has been shown that $V(S)$ together with the large curve Γ around the diagram is a biconnected planar graph with vertex-degree ≥ 3, the vertices on Γ are of degree 3. Also the opposite is true, namely each graph fulfilling these properties represents an AVD. In this book a slightly different definition of AVD's was used, but it is easy to see that the Theorem is still true for our setting based on axioms (A1) to (A3). This means that the dual of $V(S)$, the dual of the graph structure of $V(S)$ inside Γ, is a biconnected planar graph and vice versa. Thus a Hamiltonian curve with respect to $V(S)$ is equivalent to a Hamiltonian path, with its endpoints on the outer face, in a biconnected planar graph.

To show that it is NP-hard to even decide the existence of such a path, we reduce the problem of deciding whether a Hamiltonian cycle exits in a Delaunay triangulation. In [8] it has been shown that this problem is NP-complete.

Lemma 2 *Let G be a biconnected planar graph. The problem to determine whether a Hamiltonian path P with endpoints on the outer face of G exists is NP-complete.*

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

From this lemma we get our theorem.

Theorem 3 *It is NP-complete to determine if for a given abstract Voronoi diagram there exists an unbounded simple curve visiting each Voronoi region exactly once.*

Corollary 4 *For a given system of bisecting curves it is NP-comlete to determine if there exists an unbounded simple curve crossing each bisecting curve exactly once.*

Proof. An unbounded simple curve crossing each bisecting curve exactly once visits each Voronoi region in $V(S)$ exactly once, see Lemma 3 in [12]. □

3 Normalizing a Bisector System

Let $\{J(p,q) \mid p,q \in S\}$ be a system of bisecting curves in general position fulfilling axioms (A1) to (A3) and a relaxed version of (A4):

(A4') $V(S')$ is a forest for all $S' \subseteq S$.

Axiom (A4') implies that any two related bisectors cross *at most* once. Would all pairs of related bisectors cross *exactly* once then $V(S')$ would be a tree for all subsets S' of S. Let Γ be a closed curve encircling all intersection points of bisecting curves, so that each bisecting curve consists of two unbounded segments outside of Γ and a bounded segment inside.

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

Theorem 5 *By introducing crossings of neighboring bisecting curves outside of Γ (and afterwards enlarging Γ to include all new intersection points) we can transform $\{J(p,q) \mid p,q \in S\}$ into a* normal *system $\{J'(p,q) \mid p,q \in S\}$ of bisecting curves such that*

(i) axioms (A1) to (A3) and (A4') are fulfilled,

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

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

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

Proof. If the bisector system is not normal there exist three different sites p,q,r in S where $J(q,p), J(q,r), J(p,r)$ fail to cross but instead run as shown in Figure 2. By axiom (A4'), the Voronoi region of p in $V(S)$ extends to infinity, w.l.o.g. through the northern part of the strip (the southern part may be blocked by other sites in $V(S)$). Let m denote the number of unbounded southern bisector segments between $q|p$ and $p|r$. Clearly, $m \geq 1$ because $q|r$ is situated between $q|p$ and $p|r$. We call $q|p$ and $p|r$ a "strip of width m".

The theorem follows by applying the following lemma repeatedly.

Lemma 6 *If there is a strip of width $m \geq 1$ we can introduce another triplet crossing point while maintaining properties (1) to (3).*

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

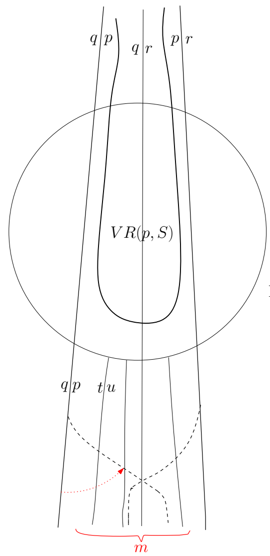


Figure 2: In principle we want to move $q|p$ and $p|r$ together such that they cross on $q|r$. This may require reordering the bisector segments in between, as we must not cause related bisectors to cross more than once.

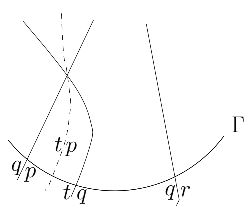


Figure 3: Illustration of the case $t|u = t|q$.

Proof. By induction on m. If $m = 1$ then $q|p, q|r, p|r$ are direct neighbors and can be "braided" to obtain a triplet crossing point, v. We enlarge Γ to include v, and have obtained the new ordering $p|r, q|r, q|p$. Axioms (A1) to (A4') are still fulfilled.

Now let $m > 1$, and let $t|u$ be the right hand side neighbor of $q|p$, as shown in Figure 2. If $t|u = q|r$ then we are done with moving $q|p$ to the right, and start to move $p|r$ to the left towards $q|r$ in a symmetric way.

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

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

$t|u = t|p$. Here $t|p$ and $p|r$ form a strip of width $m-1$, so that induction applies. $t|u = t|q$. We observe that $t \neq p$ because $q|p$ runs to the north. Similarly, we have $t \neq r$. Since the region of q is nonempty, $q|p$ and $t|q$ must intersect, as shown in Figure 3. But then $t|p$ must appear in between—a contradiction.

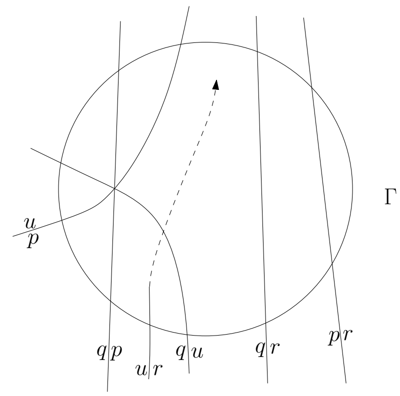


Figure 4: An impossible situation.

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

First, if $q|p$ and $q|u$ cross we have the situation shown in Figure 4. We observe that $q|u$ cannot cross $q|r$, too, because it would need to cross it twice in order to run to $q|p$. Since the region of p is unbounded to the north, $u|p$ and $p|r$ cannot cross, so $u|r$ must run between them to the north. If $u|r$ were situated, on the southern part of Γ, between $q|p$ and $q|u$, as shown in Figure 4, it could not run to the north because it could not cross $q|u$ (the crossing would have to lie on $q|r$, too, which is impossible as $q|u$ and $q|r$ are disjoint). Thus, $u|r$ appears between $q|u$ and $p|r$, and $q|u$ and $u|r$ form a strip of width $< m$. Induction applies because the region of u is unbounded to the north.

Second, let us assume that $q|u$ intersects neither $q|p$ nor $q|r$, as shown in Figure 5. In the Voronoi diagram of p, q, u, bisector $q|u$ must run between $q|p$ and $p|u$ without crossing, because the region of p runs to the north. If, on the southern boundary of Γ, bisector $p|u$ appears to the left of $p|r$ then $q|p$ and $p|u$ form a strip of width $< m$, and induction applies. Otherwise, we have the situation depicted in Figure 5. Since $q|u$ and $q|r$ are disjoint by assumption, $q|r$ and $u|r$ cannot cross. Thus, $q|u$ and $u|r$ form a strip of width $< m$, and we can apply induction since the region of u is unbounded to the north.

Third, we assume that $q|u$ intersects $q|r$ but not $q|p$; see Figure 6. As in the previous case, bisector $q|u$ must run between $q|p$ and $p|u$ without crossing, and if $p|u$ appears to the left of $p|r$ we can apply induction to the strip formed by $q|p$ and $p|u$. Let us assume that $p|u$ is situated to the right of $p|r$. If $p|r$ and $p|u$ were disjoint, either the region of r or of u would be empty in the Voronoi diagram of p, r, u, because $r|u$ would run to the left of $p|r$; see Figure 7.

Thus, there must be a crossing, as shown in Figure 6. But there is no way the two segments of $r|u$ can be connected, since multiple crossings are not allowed—a

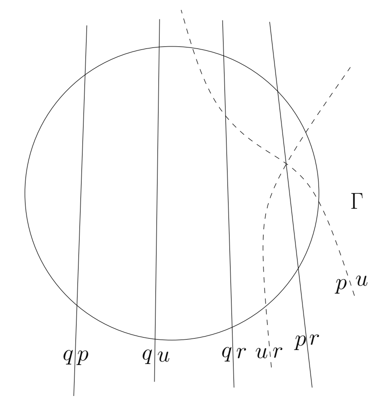


Figure 5: Bisectors $q|u$ and $u|r$ form a strip of width $< m$.

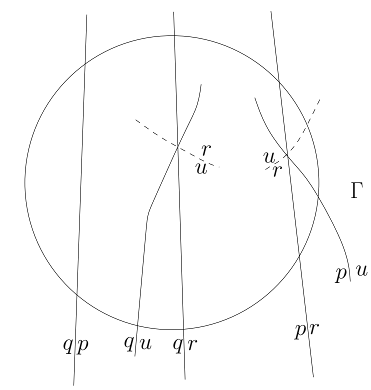


Figure 6: An impossible situation, since the two segments of $J(r, u)$ cannot be connected.

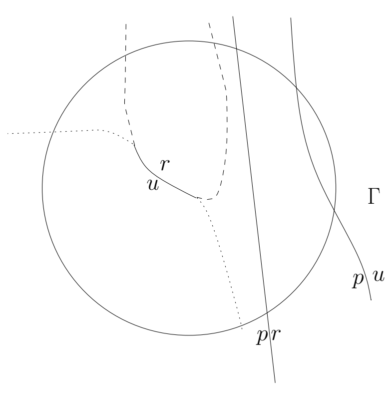


Figure 7: The Voronoi region of r or of u would be empty.

contradiction.

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

□

4 The Algorithm

Let us get back to our actual problem. Here a domain D together with the ordering of the Voronoi regions along its boundary is given.

Definition 1 *For each set of sites $S' \subseteq S$ let $\pi(S')$ be the sequence of regions of $V(S')$ along ∂D. Since $V(S)$ partitions ∂D into $|\pi(S)|$ pieces, each element of $\pi(S)$ corresponds to a unique piece. For each element p of $\pi(S)$, $d(p)$ is a point on its corresponding piece.*

Remark that $\mathrm{VR}(p, S) \subseteq \mathrm{VR}(p, S')$, thus $d(p) \in \mathrm{VR}(p, S')$ for all subsets S' of S. Actually, $\pi(S)$ depends on the starting point and the direction of a traversal along ∂D. W. l. o. g., we assume the starting point is known and the direction is **clockwise**. Axiom (A4) implies that each element in $\pi(S)$ occurs only once. For subsets S' of S we have the following observation.

Lemma 7 *$\pi(S')$ is a Davenport-Schinzel-Sequence of order 2.*

Proof. By definition no element of the sequence appears twice without another site in between. So, suppose there are $p \neq q \in S'$ such that p, q, p, q occur in this ordering in $\pi(S')$. Then either the two p's or the two q's can not be connected in D, a contradiction to axiom (A4). □

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

Definition 2 *Let $\pi'(S')$ be the subsequence of $\pi(S)$ containing all elements from S', i. e. $\pi'(S')$ is a DSS of order 1.*

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

1. Color each element of $\pi(S)$ either *blue* or *red*, i.e., π is partitioned into $\pi'(B)$ and $\pi'(R)$, and S is partitioned into B and R, such that both $|B|$ and $|R|$ are a constant fraction of $|S|$, and for each two consecutive red sites, r_1, and r_2, in π, $\mathrm{VR}(r_1, B \cup \{r_1, r_2\})$ and $\mathrm{VR}(r_2, B \cup \{r_1, r_2\})$ are not adjacent. See Section 4.1 for details.

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

3. Select a subset C from R such that $|C|$ is a constant fraction of $|R|$, and for any two sites, c_1 and c_2, $\mathrm{VR}(c_1, B \cup \{c_1, c_2\})$ and $\mathrm{VR}(c_2, B \cup \{c_1, c_2\})$ are not adjacent. See Section 4.2 for details.

   - Add artificial Voronoi edges to obtain a tree structure $V^*(B)$
   - Apply Aggarwal et al.'s selecting Lemma [1] on $V^*(B)$

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

5. Compute $V(G)$ from $\pi'(G)$ recursively, where $G = R \backslash C$ and $\pi'(G)$ is obtained from $\pi'(R)$ by removing all elements in C.

6. Merge $V(B \cup C)$ and $V(G)$.

Step 1 can be carried out in linear time according to Section 4.1, Step 3 and Step 4 can be completed in linear time according to Section 4.2 and 4.4, and Step 6 can be implemented in linear time using the general merge method described in [10]. Since $|B|$ and $|G|$ is a constant fraction of $|S|$, the above claims conclude Theorem 1.

Definition 3 *For a set S of sites, a subset S' of S, and a site p of S', a connected intersection between $VR(p, S')$ and ∂D is* redundant *if it does not contain the connected intersection between $VR(p, S)$ and ∂D. From the viewpoint of $\pi'(S')$, which is a subsequence of $\pi(S)$, a connected intersection between $VR(p, S')$ and ∂D is* redundant *if it does not contain $d(p)$ for p in $\pi'(S')$.*

Definition 4 *For all $S' \subseteq S$, a pqr-vertex of $V(S')$ is a Voronoi vertex adjacent to $VR(p, S')$, $VR(q, S')$, and $VR(r, S')$ clockwise. If $VR(p, S')$ is the only region bordering $VR(q, S')$, $VR(p, S')$ encloses $VR(q, S')$, for brevity we say p* encloses *q* in *$V(S')$.*

4.1 Red-Blue-Coloring Scheme

The coloring scheme consists of two steps, where a site is blue as long as it is not colored red. See also Figure 8.

1. For every 5 consecutive sites along $\pi(S)$, (l, m, p, q, r), p is colored *red* if one of the following conditions holds. Let T be $\{l, m, p, q, r\}$.

   (i) There is a *mpq*-vertex in $V(T)$.

   (ii) $\mathrm{VR}(m, T)$ encloses $\mathrm{VR}(p, T)$.

   (iii) $\mathrm{VR}(q, T)$ encloses $\mathrm{VR}(p, T)$.

2. For every 3 consecutive sites along $\pi(S)$ that are all blue, the middle one is colored *red*.

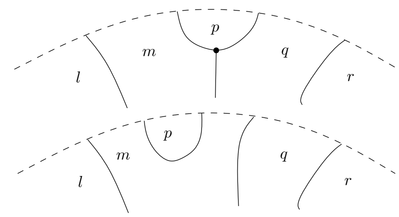


Figure 8: Two cases where p is colored red.

Let R be the set of red sites, and B be the set of blue sites. Observe that the final diagram $V(S)$ is a tree, but in the recursion $V(S)$ may be a forest, e. g. when $S = B$. Then we use the sequence $\pi'(S)$ instead of $\pi(S)$.

Lemma 8 *No 3 consecutive sites in $\pi(S)$ are all colored red.*

Proof. For the sake of a contradiction assume that three consecutive sites r_1, r_2, r_3 are all red. Let s_1 and s_2 be the two consecutive sites previous to r_1, and s_3 and s_4 the two consecutive sites after r_3. By definition r_1, r_2 and r_3 can not be colored red by step 2. Thus we need only consider step 1. There are three different cases for r_1 to be colored red.

Case 1: There is an $s_2r_1r_2$-vertex in $V(\{s_1, s_2, r_1, r_2, r_3\})$. This vertex is still an $s_2r_1r_2$-vertex in $V(\{s_2, r_1, r_2, r_3\})$ implying that there can not exist an $r_1r_2r_3$-vertex in $V(\{s_2, r_1, r_2, r_3\})$ and hence also no $r_1r_2r_3$-vertex in $V(\{s_2, r_1, r_2, r_3, s_3\})$. This means that r_2 must be colored red because r_1 or r_3 encloses it in $V(\{s_2, r_1, r_2, r_3, s_3\})$. But then r_2 can not be adjacent to a vertex in $V(\{s_2, r_1, r_2, r_3\})$, i.e., no $s_2r_1r_2$ vertex exists, a contradiction.

Case 2: r_1 is colored red because s_2 encloses r_1 in $V(\{s_1, s_2, r_1, r_2, r_3\})$. But then the regions of r_1 and r_2 are not adjacent in $V(\{s_2, r_1, r_2, r_3\})$ and there can be no $r_1r_2r_3$-vertex in $V(\{s_2, r_1, r_2, r_3, s_3\})$. Further r_1 can not enclose r_2. This means that r_2 must be colored red because r_3 encloses it in $V(\{s_2, r_1, r_2, r_3, s_3\})$. But then there can be no $r_2r_3s_3$-vertex in $V(\{r_1, r_2, r_3, s_3, s_4\})$ and r_3 can not be enclosed by r_2 or s_3 in $V(\{r_1, r_2, r_3, s_3, s_4\})$. Thus r_3 is not colored red, a contradiction.

Case 3: r_1 is enclosed by r_2 in $V(\{s_1, s_2, r_1, r_2, r_3\})$ but then because of the same reasons as in case two r_2 is not colored red. □

Corollary 9 *Let $s_1, s_2, r_1, r_2, s_3, s_4$ be 6 consecutive sites in $\pi(S)$. If r_1 and r_2 are both red, then s_2 and s_3 are both blue. Further s_2 encloses r_1 in $V(\{s_1, s_2, r_1, r_2, s_3\})$ and s_3 encloses r_2 in $V(\{s_2, r_1, r_2, s_3, s_4\})$. In particular s_2 encloses r_1 and s_3 encloses r_2 in $V(\{s_2, r_1, r_2, s_3\})$.*

Proof. Lemma 8 shows that s_2 and s_3 are both red. The case two in the proof of Lemma 8 is the only case where two consecutive sites r_1 and r_2 are both colored red. Here s_2 encloses r_1 in $V(\{s_1, s_2, r_1, r_2, s_3\})$ and r_3 encloses r_2 in $V(\{s_2, r_1, r_2, s_3, s_4\})$, implying that s_2 encloses r_1 and s_3 enclose r_2 in $V(\{s_2, r_1, r_2, s_3\})$. □

It can happen that two consecutive sites are both colored red, see Figure 9. However, we still have the following property.

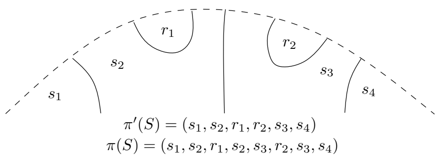


Figure 9: Two consecutive sites r_1 and r_2 in $\pi'(S)$ are both colored red, but their regions are not adjacent

Lemma 10 *Let r_1 and r_2 be two consecutive red sites. Then $VR(r_1, B \cup \{r_1, r_2\})$ and $VR(r_2, B \cup \{r_1, r_2\})$ are not adjacent.*

Proof. Let s_1 be the site previous to r_1 and s_2 the site after r_2 in π. There are three cases.

Case 1: There is no blue site between r_1 and r_2. Because of Corollary 9, s_1 and s_2 are both blue and s_1 encloses r_1 and s_2 encloses r_2 in $V(s_1, r_1, r_2, s_2)$. Thus it follows directly that the regions of r_1 and r_2 can not be adjacent in $V(B \cup \{r_1, r_2\})$.

Case 2: There is exactly one blue site b between r_1 and r_2. For the sake of a contradiction suppose $\text{VR}(r_1, B \cup \{r_1, r_2\})$ and $\text{VR}(r_2, B \cup \{r_1, r_2\})$ are adjacent. Then the regions of r_1 and r_2 are the only regions that may be adjacent to the region of b in $V(B \cup \{r_1, r_2\})$. If they both are adjacent to the region of b, then there is a $r_1 b r_2$-vertex in $V(B \cup \{r_1, r_2\})$. If only the region of r_1 is adjacent to the region of b, then r_1 encloses b in $V(B \cup \{r_1, r_2\})$ and if only the region of r_2 is adjacent to the region of b, then r_2 encloses b in $V(B \cup \{r_1, r_2\})$.

Now if s_1 and s_2 are both blue, then $\{s_1, r_1, b, r_2, s_2\} \subseteq B \cup \{r_1, r_2\}$ and b would have been colored red, a contradiction to the assumption that b is blue.

Now assume s_1 is red and let s_0 be the predecessor of s_1. Corollary 9 tells us that b encloses r_1 in $V(\{s_0, s_1, r_1, b, r_2\})$, but then the regions of r_1 and r_2 can not be adjacent in $V(B \cup \{r_1, r_2\})$. The case that s_2 is red is symmetric.

Case 3: There are exactly two blue sites b_1 and b_2 between r_1 and r_2. As in case 2, for the sake of a contradiction suppose $\text{VR}(r_1, B \cup \{r_1, r_2\})$ and $\text{VR}(r_2, B \cup \{r_1, r_2\})$ are adjacent. Then the regions of r_1, r_2, b_1 and b_2 are the only regions that may be adjacent to the regions of b_1 and b_2 in $V(B \cup \{r_1, r_2\})$.

Now there are two subcases:

Case 3.1. The region of b_1 or b_2 is not adjacent to any of the regions of r_1 or r_2 in $V(B \cup \{r_1, r_2\})$. Then b_1 encloses b_2 or b_2 encloses b_1 in $V(B \cup \{r_1, r_2\})$. W.l.o.g. let b_1 enclose b_2, the other case is symmetric. If s_2 is blue, then b_1 also encloses b_2 in $V(\{r_1, b_1, b_2, r_2, s_2\})$, and b_2 must be colored red. But if s_2 is red, then by Corollary 9 b_2 has to enclose r_2 in $V(\{b_1, b_2, r_2, s_2)\})$, both a contradiction.

Case 3.2. Both the regions of b_1 and b_2 are adjacent to the region of r_1 or r_2. Because all Voronoi regions are connected the region of b_1 or b_2 is adjacent to the region of r_1 but not r_2 or vice versa. Let b_1 be adjacent to the region of only r_1 and b_2, the other 3 constellations are symmetric. Then there is a $r_1 b_1 b_2$-vertex v in $V(B \cup \{b_1, b_2\})$. If s_1 is blue, then v is also a $r_1 b_1 b_2$-vertex in $V(\{s_1, r_1, b_1, b_2, r_2\})$ and thus b_1 would be red, a contradiction.

If s_1 is red, then r_1 would be enclosed by b_1 in $V(\{s_0, s_1, r_1, b_1, b_2\})$, a contradiction as in Case 2. □

4.2 Choosing Crimson Sites

We want to apply the following combinatorial lemma from [1] to obtain an independent set of crimson sites.

Lemma 11 *Let T be a binary tree embedded in the plane and for each leaf l a subtree T_l rooted at l is defined. Further, the subtrees of two consecutive leaves in the topological ordering around T are disjoint. Then one can in linear time find a fixed fraction of leaves whose subtrees are pairwise disjoint.*

To use this lemma we modify the forest $V(B)$, generated by the blue sites, by adding some edges and leaves to obtain a tree $V^*(B)$ fulfilling the claimed properties. We start with the following observation.

Lemma 12 *We can detect all redundant intersections of $V(B)$ in time $O(n)$.*

Proof. First compute $\pi(B)$ by deleting all sites from R from $\pi(S)$. This takes time $O(n)$.

Recall that $\pi(B)$ is the sequence of sites along ∂D in $V(B)$. This is a Davenport-Schinzel-Sequence of order 2, whereas $\pi'(B)$ is a Davenport-Schinzel-Sequence of order 1. Let $|B| = m \leq n$, $\pi'(B) = (p_1, \dots, p_m)$ and $\pi(B) = (q_1, \dots, q_l)$, where $q_1 = p_1$ refers to a non redundant intersection. Further $l \leq 2m-1$, because $\pi(B)$ is a DSS of order 2.

Let $q_{i_j} = p_j$ refer to a non redundant intersection and let $q_{i_{j+1}}$ be the first p_{j+1} after q_{i_j}. We claim that $q_{i_{j+1}} = p_{j+1}$ refers to a non redundant intersection implying that all q between q_{i_j} and $q_{i_{j+1}}$ refer to redundant intersections.

Suppose $q_{i_{j+1}} = p_{j+1}$ is redundant. Then there must be a $q' = p_{j+1}$ after $q_{i_{j+1}}$ in $\pi(B)$ referring to the non redundant intersection of p_{j+1}. This means that all q

between $q_{i_{j+1}}$ and q' are redundant and thus for all such $q \neq p_{j+1}$ there is another q before $q_{i_{j+1}}$ or after q' in $\pi(B)$. Because all faces of $V(B) \cap D$ are connected in D, the intersection of $q_{i_{j+1}}$ can be connected to the intersection of q' by path in $D \cap \mathrm{VR}(p_{j+1}, B)$. But then no $q \neq p_{j+1}$, there must be at least one between $q_{i_{j+1}}$ and q', can be connected to any intersection of q before $q_{i_{j+1}}$ or after q' by a path in $D \cap \mathrm{VR}(q, B)$, a contradiction. □

Now we construct $V^*(B)$ out of $V(B)$ by the following operations, compare Figure 10.

(i) For all redundant intersections on ∂D link the two leaves bounding it along ∂D.

If the redundant intersection borders another redundant intersection on its right end, then let the leaf between them now be a vertex in $V^*(B)$. Observe that this is a vertex of degree 3. Otherwise connect the right end of the link to $V(B)$ without creating a vertex. The same is done on the left side of the redundant intersection.

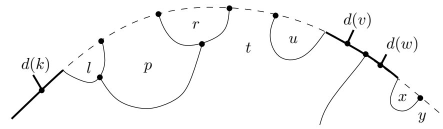
Next we attach some leaves to $V^*(B)$ outside of D such that between each pair of consecutive blue sites b_i and b_{i+1} having one (or two) red site(s) in between, there is exactly one (or between one and two) leaves in $V^*(B)$. If there is no red site between b_i and b_{i+1} there is also no leaf.

(ii) If there are one or two red sites r_1 and r_2 between two consecutive blue sites b_i and b_{i+1} but no leaf between them, then there is a connected set of redundant intersections between b_i and b_{i+1}. If $d(r_j)$, $j = 1, 2$ lies within this sequence we attach a leaf to $V^*(B)$ at $d(r_j)$, otherwise if $d(r_j)$ lies to the left (right) of the sequence we attach a leaf at the leftmost (rightmost) point of the sequence. If both $d(r_j)$, $j = 1, 2$ are to the left (right) of the redundant intersection sequence, only one leaf is attached at the leftmost (rightmost) point.

Between two consecutive blue sites there are at most two red site. Thus for every connected sequence of redundant intersections at most two leaves are attached. Further, between each pair of consecutive blue sites separated by one or two red sites there is now at least one leaf.

(iii) If there is a leaf in $V(B)$ between two consecutive blue sites b_i and b_{i+1}, which are not separated by a red site, it is pruned like in [12].

Lemma 13 *$V^*(B)$ is a binary tree and can be constructed in time $O(n)$.*



$$\pi(S) = (k, l, m, p, q, r, s, t, u, v, w, x, y \ldots)$$
$$\pi'(B) = (l, p, r, t, u, x, y \ldots)$$
$$\pi(B) = (t, l, p, r, t, u, t, y, x, y, \ldots)$$

Figure 10: $V^*(B)$, fat edges indicate redundant intersections and new leaves.

Proof. It is clear that $V^*(B)$ is a forest. So assume it is disconnected. Then there is a site $b \in B$ whose Voronoi region in $V(B)$ intersects ∂D in more than one component. By (i) all these components are non redundant. But then b has to appear several times in $\pi(S)$, a contradiction. Definitions (i) to (iii) imply that all internal nodes of $V^*(B)$ are of degree 3.

By lemma 12 we can detect all redundant intersections in time $O(n)$. In the same time operation (i) can be accomplished. For operation (ii) and (iii) we have to walk once around ∂D and look at consecutive blue sites.

For each pair of consecutive blue sites b_i and b_{i+1} we test if there are zero, one or two red sites in between. If there is no red site between b_i and b_{i+1} but a leaf, we prune the leaf in constant time. If there are one or two red sites r_1 and r_2 but no leaf between b_i and b_{i+1} we test if $d(r_1)$ and $d(r_2)$ lie to the left, within or to the right of the redundant intersection sequence between the two blue sites and attach one or two leaves like described in (ii). For each redundant intersection this takes constant time and there are $O(n)$ redundant intersections altogether. □

4.3 Coloring Crimson

If two blue sites b_i and b_{i+1} are separated by a red site r in $\pi(S)$ but the leaf between them is not contained in $\mathrm{VR}(r, B \cup \{r\})$, then r is enclosed by the region of b_i (or b_{i+1}) in $V(B \cup \{r\})$. In this case color r crimson *with respect to* b_i (b_{i+1}) and if the leftmost (rightmost) leaf between b_i and b_{i+1} is not contained in the region of a consecutive site, associate with r the subtree containing only this leaf. If two red sites are between b_i and b_{i+1} and both are colored crimson because of b_i (b_{i+1}) associate only one of them with the leftmost (rightmost) leaf.

Up to now we may already have colored some red sites crimson. To make sure we receive a fixed fraction of crimson sites we apply lemma 11 in the following way. For each leaf l of $V^*(B)$ contained in a red region $\mathrm{VR}(r, B \cup \{r\})$ define T_l by the subtree spanned by all vertices of $V^*(B)$ contained in $\mathrm{VR}(r, B \cup \{r\})$. The next lemma shows that this is possible.

Lemma 14 *Let r be a red site. If $VR(r, B \cup \{r\})$ intersects a leaf of $V^*(B)$, then $VR(r, B \cup \{r\}) \cap V^*(B)$ is connected. Otherwise it is empty.*

Proof. Suppose $\mathrm{VR}(r, B \cup \{r\})$ intersects a leaf of $V^*(B)$ and $\mathrm{VR}(r, B \cup \{r\}) \cap V^*(B)$ is not connected. Then $\mathrm{VR}(r, B \cup \{r\})$ would disconnect the region of a blue site in $V(B \cup \{r\})$, a contradiction.

If $\mathrm{VR}(r, B \cup \{r\})$ does not intersect a leaf of $V^*(B)$ it must be contained within a single blue region of $V(B)$, thus it can not intersect $V^*(B)$. □

Now Lemma 10 and 11 imply the requested property.

Lemma 15 *No regions of two crimson sites are adjacent in $V(B \cup C)$.*

Proof. To test whether the parent node of a leave is contained in the region of a red site it is enough to consider the diagram of the three sites adjacent to the node and the red site. Thus this test can be done in constant time. Further each leaf of F' is associated with the region of a red site. This ensures a correct application of Lemma 11 and finishes the proof. □

4.4 Insertion of Crimson Sites

We can now insert the crimson sites into $V(B)$ in order to receive $V(B \cup C)$. For each crimson site c whose region does not intersect a leaf of $V^*(B)$ we know that it is enclosed by the region of a blue site b_i or b_{i+1}. Let it be b_i, then we just have to insert the part of the bisector $J(r, b_i)$ contained in $\mathrm{VR}(b_i, B)$ as a new edge in $V(B \cup C)$. The other crimson sites can be inserted along the subtrees of $V^*(B)$ associated with them. Thus also the insertion takes time $O(n)$.

5 Discussion

A natural question is if it is possible to relax axiom (A4) and still have a linear time algorithm for computing the Voronoi diagram. There exist applications where Voronoi regions restricted to the domain D are disconnected. This is something that can happen for the farthest Voronoi diagram of line segments or when the domain D corresponds to a Voronoi region which is to be deleted from a given Voronoi diagram.

References

[1] A. Aggarwal, L. J. Guibas, J. Saxe, and P. W. Shor. A linear-time algorithm for computing the Voronoi diagram of a convex polygon. Discrete and Computational Geometry 4, pp. 591–603, 1989.

[2] F. Aurenhammer. Voronoi Diagrams: A Survey of a Fundamental Geometric Data Structure. ACM Computing Surveys 23(3), pp. 345–405, 1991.

[3] F. Aurenhammer and R. Klein. Voronoi Diagrams. In: J.R. Sack and G. Urrutia (Eds.), Handbook on Computational Geometry, Elsevier, pp. 201–290, 1999.

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

[5] C. Bohler, R. Klein, and C.-H. Liu. Forest-Like Abstract Voronoi Diagrams in Linear Time. Proc. 30th European Workshop on Computational Geometry. 2014.

[6] F. Chin, J. Snoeyink, and C. A. Wang. Finding the medial axis of a simple polygon in linear time. Discrete Computational Geometry 21, pp. 405–420, 1999.

[7] F. Chin and C. A. Wang. Finding the constrained Delaunay triangulation and constrained Voronoi diagram of a simple polygon in linear time. SIAM Journal on Computing 28(2), pp. 471–486, 1998.

[8] M. Dillencourt. Finding Hamiltonian Cycles in Delaunay triangulations is NP-complete. Discrete Applied Mathematics 64, pp. 207-217, 1996.

[9] S. Fortune. Voronoi diagrams and Delaunay triangulations. In: J.E. Goodman and J. O'Rourke (Eds.), Handbook of Discrete and Computational Geometry, Chapter 20, CRC Press LLC, pp. 377–388, 1997.

[10] R. Klein. Concrete and Abstract Voronoi Diagrams. Lecture Notes in Computer Science 400, Springer-Verlag, 1987.

[11] R. Klein, E. Langetepe, and Z. Nilforoushan. Abstract Voronoi Diagrams Revisited. Computational Geometry: Theory and Applications 42(9), pp. 885-902, 2009.

[12] R. Klein and A. Lingas. Hamiltonian abstract Voronoi diagrams in linear time. 1994 International Symposium on Algorithms and Computation (ISAAC'94), pp. 11–19, 1994.

[13] R. Klein and A. Lingas. Manhattonian proximity in a simple polygon. International Journal of Computational Geometry and Applications 5, pp. 53–74, 1995.

[14] R. Klein and A. Lingas. A linear-time randomized algorithm for the bounded Voronoi diagram of a simple polygon. International Journal of Computational Geometry and Applications 6(3), pp. 263–278, 1996.

[15] R. Klein, K. Mehlhorn, and St. Meiser. Randomized incremental construction of abstract Voronoi diagrams. Computational Geometry 3, pp. 157–184, 1993.

[16] D. T. Lee. Medial axis transformation of a planar shape. IEEE Transaction on Pattern Analysis and Machine Intelligence 4(4), pp. 363–369, 1982.

[17] D. T. Lee. On k-Nearest Neighbor Voronoi Diagrams in the Plane. IEEE Trans. Computers 31(6), pp. 478-487, 1982.

[18] D. T. Lee and A. Lin, Generalized Delaunay triangulations for planar graphs, Discrete Computational Geometry 1, pp. 201–217, 1986

[19] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu. Spatial Tessellations: Concepts and Applications of Voronoi Diagrams. Wiley Series in Probability and Statistics, 2000.