ON COMPUTING VORONOI DIAGRAMS FOR SORTED POINT SETS

HRISTO N. DJIDJEV
Department of Computer Science, Rice University,
P.O. Box 1892, Houston, TX 77251, USA
and
ANDRZEJ LINGAS
Department of Computer Science, Lund University,
P.O. Box 118, 22100 Lund, Sweden

April 1995

ABSTRACT

We show that the Voronoi diagram of a finite sequence of points in the plane which gives sorted order of the points with respect to two perpendicular directions can be computed in linear time. In contrast, we observe that the problem of computing the Voronoi diagram of a finite sequence of points in the plane which gives the sorted order of the points with respect to a single direction requires $\Omega(n \log n)$ operations in the algebraic decision tree model. As a corollary from the first result, we show that the bounded Voronoi diagrams of simple n-vertex polygons which can be efficiently cut into the so called monotone histograms can be computed in $o(n \log n)$ time.

1. Introduction

Consider a set S of n points (sites), $S = s_1, ..., s_n$, in the Euclidean plane. For $i = 1, ..., n$, let $R(s_i)$ be the region of all points p in the plane which are closer to s_i than to any other site in S. The region $R(s_i)$ can be seen as the common intersection of the $n-1$ half-planes induced by the perpendicular bisectors of the segment with endpoints s_i and s_j, $j = 1, ..., n, j \neq i$, that contain s_i. For this reason, the regions $R(s_i)$ are convex. They form a partition of the plane called the *Voronoi diagram* of S ($Vor(S)$ for short).[4,10] The maximal straight-line segments or half-lines on the boundaries of the regions in $Vor(S)$ are called edges of $Vor(S)$. The endpoints of the edges are called vertices of $Vor(S)$. The straight-line dual of $Vor(S)$ is called the Delaunay triangulation of S.

The problem of computing Voronoi diagrams and Delaunay triangulations and their diverse generalizations is central in computational geometry because of the variety of applications. There are few known algorithms for computing $Vor(S)$ which run in time $O(n \log n)$.[5,10] They are time-optimal since for instance in the algebraic decision tree model the problem of sorting n real numbers easily reduces to the problem of computing $Vor(S)$.[10] However, it might be possible to use special properties of a restricted class of point sets in order to obtain $o(n \log n)$-time algorithms. The problem of finding such classes seems to be very hard. The only previously known result is for a point set S, where the sites in S are given as vertices of a convex polygon in clockwise or counterclockwise order. Aggarwal, Guibas, Saxe and Shor have shown that $Vor(S)$ can be computed in linear time in this case.[1] Actually, they proved a theorem about computing the convex hull of certain polygons in the three-dimensional Euclidean space in linear time (Section 3) [1]. By the known relationship between the problem of computing $Vor(S)$ and the problem of computing the convex hull of projections of S on the paraboloid $U = \{(x, y, x^2 + y^2) \mid (x, y) \in E^2\}$[4,6], they could conclude that the problem of computing the Voronoi diagram of the vertices of a convex polygon is solvable in linear time. They also showed a bunch of other problems in the plane to be solvable in linear time, using their aforementioned theorem or its generalization (Section 4)[1].

In this paper, we derive an interesting consequence from the generalized theorem of Aggarwal et al. stating that if the sequence of sites on the plane is given in sorted order with respect to two perpendicular directions then their Voronoi diagram can be computed in linear time. Let a *histogram* (a *monotone histogram*) respectively denote a simple polygon $(h_1, ..., h_n, h_{n+1})$ such that the subsequence $(h_1, ..., h_n)$ is sorted with respect to a direction (or to two perpendicular directions, respectively). We can also rephrase our result as follows: the Voronoi diagram of the set of vertices of a monotone histogram given in clockwise or counterclockwise order along its perimeter can be computed in linear time. Note that a monotone histogram may have up to a linear number of reflex angles, and on the other hand any convex polygon can be trivially cut into four monotone histograms. Thus our result for monotone histograms is more general that the statement that the Voronoi diagram of convex polygon can be computed in linear time (as it implies the statement). In contrast, we observe that the problem of computing the Voronoi diagram of the set of vertices of a histogram requires $\Omega(n \log n)$ operations in the algebraic decision tree model.[10] See also the related results by Seidel.[12]

In [1], the authors also mention the open problem of computing the so called *generalized Delaunay triangulation* of a simple polygon in time $o(n \log n)$. The generalized Delaunay triangulation of a simple polygon P is a triangulation of P such that no circumcircle of a triangular face within the polygon contains a vertex of P in its interior. It includes the straight-line dual of the so called *bounded Voronoi diagram* of P.[9,11] In analogy to the concept of the standard Voronoi diagram, the latter is defined as follows. The bounded Voronoi diagram of P, $Vorb(P)$ for short, is the partition of the interior of P into regions $BR(v)$, where v ranges over vertices of P, such that a point inside P belongs to $BR(v)$ if and only if (p, v) is the shortest straight-line segment that connects p with a vertex of P without crossing any edge of P. The edges of $Vorb(P)$ are maximal straight-line segments on the boundaries of its regions that do not overlap with the perimeter of P, and the vertices of $Vorb(P)$ are the endpoints of the edges. It is not difficult to see that if P is a convex polygon or a monotone histogram then $Vorb(P)$ is identical to the intersection of the standard Voronoi diagram of the vertices of P with the inside of P. This is not necessarily true for more complicated simple polygons like histograms, monotone polygons or star-shaped polygons.[10] It is an open problem whether $Vorb(P)$ could be build in substantially less than $n \log n$ time for such polygons P. In this paper, we show that any simple polygon that can be efficiently cut into a small number of monotone histograms admits such a time-efficient construction of the bounded Voronoi diagram.

The paper is organized as follows. In Section 2 we formulate two previously known facts that we use to obtain our results. In Section 3 we develop a linear time algorithm for constructing Voronoi diagrams of monotone histograms. In Section

4 we provide an $\Omega(n \log n)$ lower bound for the problem of constructing Voronoi diagrams of general (non- monotone) histograms thus showing that our results can not be generalized to arbitrary histograms. Finally, in Section 5, we consider an application of our algorithm from Section 3.

2. Preliminaries

Guibas and Stolfi defined the so called *lifting mapping* μ of the Oxy plane into E^3 by $\mu(x, y) = (x, y, x^2 + y^2)$.[6] The paraboloid which is the image of the whole plane under the mapping μ will be denoted by U.

Fact 1 [4] *Let S be a finite set of points in the plane. There is a one-to-one correspondence between the edges of the lower part of the convex hull of $\mu(S)$ and the edges of the Delaunay triangulation of S. Given the convex hull of $\mu(S)$, we can compute the Delaunay triangulation of S in linear time.*

Aggarwal et al. used the above fact to derive their linear upper bound on the time needed to construct the Voronoi diagram of a convex polygon. They proved that the convex hull of a polygon in E^3 which is the image of a convex polygon under the mapping μ can be computed in linear time. The convexity implies that the Voronoi diagram of the polygon in the plane is a tree. Hence, by Fact 2.1, the straight-line dual to the lower hull of the polygon in E^3 is also a tree. Analogously, by the correspondance between the edges of the upper hull of $\mu(S)$ and the so-called furthest-site Delaunay triangulation of S,[4] the straight-line dual to the upper hull of $\mu(S)$ is also a tree. This is essential in the proof in [1]. In (Section 4)[1] Aggarwal et al. also presented the following generalization of their theorem on convex hull construction.

Fact 2 [1] *Let P be a polygon $(p_1, ..., p_n)$ in E^3. Suppose that for each edge of any subpolygon P' of P given by a subsequence of $(p_1, ..., p_n)$ there exists a plane that contains the edge and leaves all other vertices of P' in the same half-space. Then the convex hull of the vertices of P can be constructed in time $O(n)$.*

The upper or lower convex hull of the above polygon P doesn't have to be a tree in general. However, the closed edge path on the convex hull given by $(p_1, ..., p_n)$, not necessarily overlapping with the horizon, yields a forest structure on either side. Therefore, the algorithm due to Aggarwal et al. works here all the same with the "halfspheres" defined by $(p_1, ..., p_n)$.

3. The main result

Recall from Introduction that a *histogram* is a simple polygon $H = (h_1, ..., h_{n+1})$ $(n > 1)$ such that there is a direction in which the subsequence $(h_1, ..., h_n)$ is sorted. If there are two perpendicular directions in which the subsequence $(h_1, ..., h_n)$ is sorted then H is a monotone histogram. See Fig. 1. (Note that our definition of a histogram and a monotone histogram is slightly more general than the usual one where one would require the distinguished directions to be perpendicular to the edge (h_1, h_{n+1}) or to the edges (h_1, h_{n+1}) and (h_n, h_{n+1}) respectively). To simplify the exposition we assume throughout the paper that no three points from H are colinear. We also assume that the sequence of vertices of monotone histograms form *increasing* sequences with respect to the two perpendicular directions.

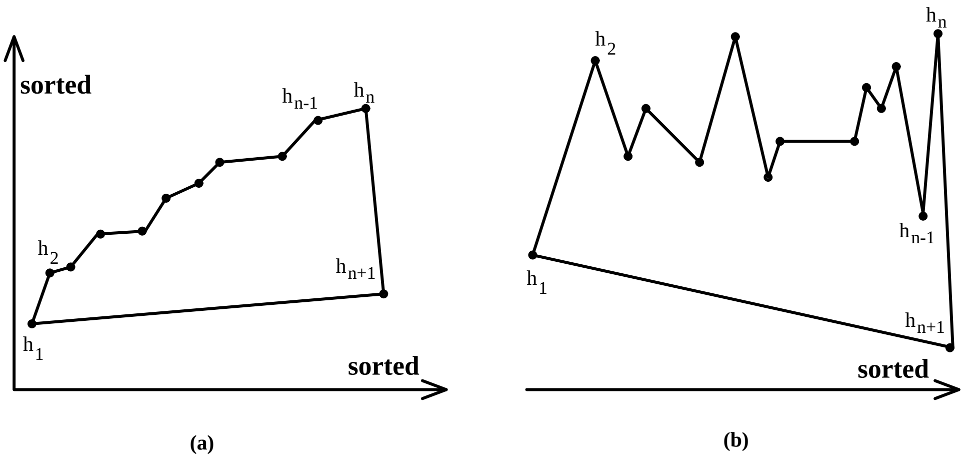

Fig. 1. (a) a monotone histogram (b) a non-monotone histogram

Theorem 1 *Let H be a monotone histogram. The Voronoi diagram of any k-vertex subsequence H' of H can be computed in time $O(k)$.*

Proof. Let $H = (h_1, ..., h_n, h_{n+1})$ and $H' = (h'_1, ..., h'_k)$. We may assume without loss of generality that h_{n+1} is not in H' since we can always update in linear time the Voronoi diagram after adding a single site.[10] Consider the projection $\mu(H') = (\mu(h'_1), ..., \mu(h'_k))$ of H on the paraboloid U. By Facts 1 and 2, it is sufficient to prove that for each edge of $\mu(H')$ there is a plane that includes the edge and leaves all other vertices of H' in the same half-space.

The above property is easily seen to hold for the edges $(\mu(h'_i), \mu(h'_{i+1}))$, where $i = 1, ..., k-1$, by the relationship between the convex hull of $\mu(H')$ and the Delaunay triangulation of the vertices of H' following from Fact 1. To prove this consider a point p on (h'_i, h'_{i+1}). Since H' is a subsequence of a monotone histogram then all vertices of H' below h'_i have a larger distance to p than h'_i has. Similarly, all vertices of H' above h'_{i+1} have a larger distance to p than h'_{i+1} has. It follows that the edge (h'_i, h'_{i+1}) has to intersect an edge of the Voronoi diagram of the vertices of H' separating the region of h'_i from the region of $h'_i + 1$. Consequently, $(h'_i, h'_i + 1)$ is an edge of the Delaunay triangulation of the vertices of H' which implies that $(\mu(h'_i), \mu(h'_{i+1}))$ is an edge of the convex hull of $\mu(H')$ by Fact 1. This implies the existence of a plane satisfying the required condition.

The hard part of the proof is to show that there exists a plane that includes $(\mu(h'_k), \mu(h'_k))$ and leaves all other vertices of H' in the same half-plane. This follows from the lemma below. □

Lemma 1 *Let $h'_1, h'_2, ..., h'_k$ be a sequence of points in the plane which is sorted in increasing order with respect to two perpendicular directions. Then there exists a plane Π containing the segment with endpoints $\mu(h'_1)$ and $\mu(h'_k)$ such that all points $\mu(h'_i), i = 2, ..., k-1$, belong to the same halfspace determined by Π.*

Proof. We will prove the lemma by explicitly constructing Π. Choose a coordinate system in E^3 such that the points of H' are in the Oxy plane and the two perpendicular directions are axes Ox and Oy. For any $J : E^3 \longrightarrow \{true, false\}$

by $Q\{J(x,y,z)\}$ we shall denote the set of points of E^3 for which $J(x,y,z) = true$. For instance, $Q\{x > x*\}$ and $Q\{x < x*\}$ will denote the two halfspaces determined by the plane $x = x*$. Let $h'i = (x_i, y_i)$, $i = 1, ..., k$, and let D denote the part of the paraboloid $z = x^2 + y^2$ that is contained in the region

$$Q\{(x_1 < x < x_k) \wedge (y_1 < y < y_k)\} = Q\{x > x_1\} \cap Q\{x < x_k\} \cap Q\{y > y_1\} \cap Q\{y < y_k\}.$$

It is obvious that $\mu(h'_i) \in D, i = 2, ..., k-1$, and therefore it will be enough to define Π so that all points of D belong to the same half space determined by Π.

The intersection of D with any plane Π_σ is perpendicular to Oxy and incident to the point $\mu(h'_1) = (x_1, y_1, x_1^2 + y_1^2)$, where $\sigma \in [0, \Pi/2]$ is the angle between Ox and Π_σ, is a portion of a parabola in Π_σ is with endpoints $(x_1, y_1, x_1^2+y_1^2)$ and some point on $Q\{x = x_k\}$ or $Q\{y = y_k\}$ depending on σ. The endpoints of these parabolas are $\mu(h'_k)$ and $a = (x_k, y_1, x_k^2+y_1^2)$, in $Q\{x = x_k\}$, and $\mu(h'_k)$ and $b = (x_1, y_k, x_1^2+y_k^2)$, in $Q\{y = y_k\}$. From the convexity of function $f(x) = x^2$ in E^2 all points of $D \cap Q\{x = x_k\}$ are below the segment joining a and $\mu(h'_k)$ and all points of $D \cap Q\{y = y_k\}$ are below the segment joining b and $\mu(h'_k)$. Then also in Π_σ any point of $D \cap \Pi$ lies below the segment ss joining $\mu(h'_1)$ with the intersection point between Π_σ and $(a, \mu(h'_k))$ for $\sigma \in [0, \arctan(y_k/x_k)]$, or $(b, \mu(h'_k))$ for $\sigma \in [\arctan(y_k/x_k), \pi/2]$. We will complete the proof by showing that points $\mu(h'_1), \mu(h'_k), a$, and b are coplanar and all segments s_σ belong to the plane, Π, defined by $\mu(h'_1), \mu(h'_k), a$, and b (notice that $(\mu(h'_1), \mu(h'_k)) = s_\sigma$ if $\sigma = \arctan(y_k/x_k)$). As

$\mu(h'_1) = (x_1, y_1, x_1^2 + y_1^2)$,
$\mu(h'_k) = (x_k, y_k, x_k^2 + y_k^2)$,
$a = (x_k, y_1, x_k^2 + y_1^2)$,
$b = (x_1, y_k, x_1^2 + y_k^2)$,

then the segment $(\mu(h'_1), \mu(h'_k))$ and the segment (a, b) intersect at the point $(\frac{x_1+x_k}{2}, \frac{y_1+y_k}{2}, \frac{x_1^2+y_1^2+x_k^2+y_k^2}{2})$, whence the claim follows. □

Remark 1 Lemma 1 can also be proved by purely algebraic means by showing that the determinant of the matrix

$$C = \left\{ \begin{array}{cccc} x_1 & y_1 & x_1^2 + y_1^2 & 1 \\ x_1 & y_k & x_1^2 + y_k^2 & 1 \\ x_k & y_1 & x_k^2 + y_1^2 & 1 \\ x & y & x^2 + y^2 & 1 \end{array} \right\}$$

is negative for $x_1 < x < x_k$ and $y_1 < y < y_k$. By elementary linear algebra computations we find

$$det(C) = (x_k - x_1)(y_k - y_1)[(x - x_1)(x - x_k) + (y - y_1)(y - y_k)],$$

whence $det(C) < 0$ for $x_1 < x < x_k$ and $y_1 < y < y_k$.

Actually, an analogous determinant is examined in the proof of a known fact implying that the image of any circle under the lifting mapping μ is contained in a plane such that the two resulting halfspaces respectively contain the images of the interior and the exterior of the circle under μ (Lemma 8.1)[6]. As it was pointed by an unknown referee, we can also reduce the proof of Lemma 1 to the above fact by constructing the circle that passes through (x_1, y_1) and (x_k, y_k) and encircles all the points (x, y) where $x_1 < x < x_k$ and $y_1 < y < y_k$.

Remark 2 If a histogram $H = (h_1, ..., h_n, h_{n+1})$ is monotone, the corresponding two perpendicular directions can be easily found in $O(n)$ time (if not given) by considering the set of the angles between some fixed half-line and the half-lines determined by $(h_i, h_{i+1}), i = 1, ..., n-1$.

4. The lower bound

To derive an $\Omega(n \log n)$ lower bound on the problem of computing the Voronoi diagram of vertices of a histogram in the algebraic decision tree model, we consider the following auxiliary decision problem:

Given a sequence $x_1, .., x_n$ of n reals, decide whether the smallest difference between any two elements in the sequence is smaller than ε, where ε is any positive constant.

The above problem is called ε-closeness problem in [6], where also the following claim is stated.

Fact 3 *For any $\varepsilon > 0$ the ε-closeness problem requires $\Omega(n \log n)$ tests in the algebraic decision tree model.*

On the other hand, we have the following lemma.

Lemma 2 *The problem of 1-closeness can be reduced to the problem of computing the Voronoi diagram of vertices of a histogram using a linear number of operations.*

Proof. Let $x_1, x_2, ..., x_n$ be a sequence of n reals. We may assume without loss of generality that the reals are greater than $1/2$. Form the histogram $H = ((0,0), (1/(3n), x_1), (2/(3n), x_2), ..., ((n/(3n), x_n), (1/2, 0))$ (see Figure 2). Compute $Vor(H)$ and its intersection with the Oy axis. Test whether all intervals $R(x_i) \cap Oy, i = 1, ..., n$, are not empty and appear in a sorted order with respect to Oy (it can easily be done using a linear number of tests). If so, we can trivially find the minimum difference between any two x_i's using a linear number of tests, and compare it with 1. Otherwise, there exists $j, 1 \leq j \leq n$, such that the projection of the site $(j/(3n), x_j)$ on the Oy axis, i.e. $(0, x_j)$, belongs to the region of another site. The other site can be neither $(0,0)$ nor $(1/2, 0)$ since $x_j > 1/2$ and the site $(j/(3n), x_j)$ is in the distance at most $1/3$ from $(0, x_j)$. It follows from the construction of H that there is a site $(k/(3n), x_k)$, $1 \leq k \leq n$, $k \neq j$, which is at distance at most $1/3$ from $(0, x_j)$. This immediately implies that $(0, x_k)$ is at distance smaller than $1/3$ from $(0, x_j)$. Thus, the minimum difference is definitely smaller than 1 in this case. $\square$

Note that the problem of 1-closeness is actually reduced to the problem of computing the Voronoi diagram of vertices of the histogram H outside H in the proof of Lemma 2. By moving H slightly to the left such that the topmost vertex is placed on the Oy axis, we can obtain an analogous reduction of the problem of 1-closeness to the problem of computing the Voronoi diagram of vertices of H inside H using a linear number of operations.

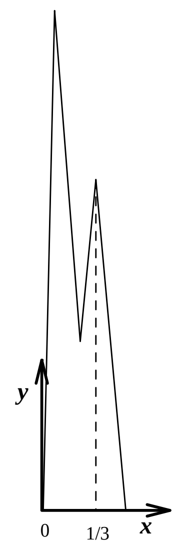

Fig. 2. The histogram H from the proof of Lemma 2

Combining Fact 3 with Lemma 2, we obtain our lower bound.

Theorem 2 *The problem of computing the Voronoi diagram or Delaunay triangulation of the vertices of an n-vertex histogram requires $\Omega(n \log n)$ operations in the algebraic decision tree model.*

Remember that the above lower bound does not give any evidence that the bounded Voronoi diagrams of histograms or even of more general simple polygons cannot be computed in time $o(n \log n)$. By slightly modifying the proof of Theorem 2, we can also derive the following theorem.

Theorem 3 *The problem of computing the Voronoi diagram or Delaunay triangulation of the vertices of a star-shaped n-vertex polygon requires $\Omega(n \log n)$ operations in the algebraic decision tree model.*

Proof. We may assume without loss of generality that the reals $x_1, x_2, ..., x_n$ in the proof of Lemma 2 are in the range $1/2$ through m. Augment the histogram H in the proof of Lemma 2 by the triangle induced by $(0,0), (0,1/2), (1/4, -2nm)$. For any non- base vertex v of H, the triangle resulting from extending the two adjacent edges of H down to the horizontal line passing through $(1/4, -2nm)$ would touch $(1/4, -2nm)$ by Thales' theorem. It follows that the augmented histogram is a star-shaped poygon. Since the Voronoi region of $(1/4, -2nm)$ in the Voronoi diagram of the augmented histogram always lies below the X-axis the argumentation from the proof of Lemma 2 also works here. □

.

5. Computing bounded Voronoi diagrams for certain polygons

By a chord of a simple polygon P, we shall mean a straight-line segment s such that the endpoints of s lie on the perimeter of P and the inside of s lies inside P. Note that the endpoints of a chord of P are not necessarily vertices of P. In order to compute the bounded Voronoi diagram of a simple polygon ($Vorb(P)$, see

Introduction), we shall recursively split it into monotone histograms along chords. Then, we shall compute the bounded Voronoi diagrams of the monotone histograms and recursively merge them employing the following lemma.

Lemma 3 *Let P be a simple polygon, and let s be a chord of P. Let P_1 and P_2 be the two polygons resulting from splitting P along s. Given the bounded Voronoi diagrams of P_1 and P_2, the bounded Voronoi diagram of P can be computed in linear time.*

Proof. Recall that the straight-line dual of the bounded Voronoi diagram of a simple polygon is a subset of the generalized Delaunay triangulation of the polygon.[9,11] Lee and Lin[7] showed how to merge the generalized Delaunay triangulations of two polygons, resulting from splitting a larger polygon along a diagonal in linear time (this enabled them to derive an $O(n \log n)$-time divide-and-conquer algorithm for the generalized Delaunay triangulation of a simple polygon). On the other hand, Seidel[13] has introduced an extension of the bounded Voronoi diagram of a planar straight-line graph (in particular, of a simple polygon) called extended Voronoi diagram showing that its straight-line dual is the generalized Delaunay triangulation. To use the above facts, for i=1,2, we compute the extended Voronoi diagram of P_i from the bounded Voronoi diagram of P_i in linear time following the definition from [13] as follows. First, for each edge e of P_i we compute the Voronoi diagram $V(e)$ of the vertices of P_i whose regions in the bounded Voronoi diagram of P_i support the edge e, on the other side of the straight-line induced by e. Note that the order in which the regions support e corresponds to the order of the vertices definig them in the direction of e. Hence, by (Section 4)[1] and the linear size of the bounded Voronoi diagram of P_i , the constructions of the diagrams $V(e)$ for all edges e of P_i totally takes linear time. Next, we extend the regions of vertices of P_i in the bounded Voronoi diagram of P_i by their regions in $V(e)$ on the other sides of the edges e they support. It is sufficient now to compute the straight-line dual of the resulting extended diagram with possibly overlapping regions to obtain the generalized Delaunay triangulation of P_i.[13] Clearly, it can be done in linear time. Suppose first that the endpoints of s are vertices of P. We can merge the generalized Delaunay triangulations of P_1 and P_2 into the generalized Delaunay triangulation of P in linear time.[7] Next, the extended (bounded) Voronoi diagram of P can be obtained from the generalized Delanay triangulation of P which is its straight-line dual in linear time (in both cases we use the standard DCEL representation)[10]. Finally, it is enough to cut the extended Voronoi diagram along the perimeter of P to to obtain the bounded Voronoi diagram of P. Suppose that not all endpoints of s are vertices of P. Then, after merging the bounded Voronoi diagrams of P_1 and P_2 into the bounded Voronoi diagram M of P with an endpoint (or both endpoints, respectively) of s added to the set of vertices of P, we divide the region of each added vertex between the adjacent regions of vertices of P. Note that if we consider the standard Voronoi diagram of the vertices defining the adjacent regions and the added vertex (added vertices, respectively) then the boundaries of the region of any added vertex will overlap with those from M. For this reason, via Fact 1, the projections of the vertices defining the adjacent regions on the paraboloid U lie on a convex cone (whose apex is the projection of the added vertex on U). Hence, by Fact 2, we can update the diagram M after deleting the added vertex (vertices, respectively) in linear time. See also (Section 4)[1]. □

To formalize the notion of recursively cutting a simple polygon into smaller polygons along chords we need the following definition. By a *full binary tree* we mean a binary tree with any non-leaf node of degree two.

Definition 1 *Let P be a simple polygon. A full binary tree T with each node v labelled with a simple polygon $P(v)$ and each non-leaf node additionally labelled with a chord $s(v)$ of $P(v)$ is said to* cut *P into the set of polygons assigned to its leaves if it satisfies the following conditions:*

(1) if v is the root of T then $P(v) = P$;

(2) if a node v has sons w, u in T then the simple polygons $P(w)$ and $P(u)$ can be obtained by splitting $P(v)$ along $s(v)$.

Theorem 4 *Let T be a full binary tree of height d which cuts a simple n-vertex polygon P into monotone histograms each containing at least one vertex of P. The bounded Voronoi diagram of P can be computed in time $O(nd)$.*

Proof. First, we compute the Voronoi diagrams of the monotone histograms and convex polygons labelling the leaves of T. Their total size is $O(n)$. Therefore, it takes $O(n)$ time by Theorem 1. Then, for $i = 1, ..., d$, we compute the bounded Voronoi diagrams of the polygons that label the i-th level of T assuming that the bounded Voronoi diagrams of the polygons that label the $(i-1)$-th level of T are already computed and using Lemma 3. As the total size of all polygons and diagrams involved in the above computation corresponding to the i-th level of T is again $O(n)$, the computation takes also $O(n)$ time by Lemma 3. Since there are d levels in T, the total time is $O(nd)$. □

Since any convex polygon can be trivially cut into four histograms using three chords, we obtain the following generalization of the above theorem as an immediate corollary from it.

Corollary 1 *Let T be a binary tree of height d which cuts a simple polygon P into monotone histograms and convex polygons each containing at least one vertex of P. The bounded Voronoi diagram of P can be computed in time $O(nd)$.*

Corollary 2 *Let P be a simple polygon on n vertices. If P can be cut by r parallel chords into $r+1$ monotone histograms or convex polygons then its bounded Voronoi diagram can be computed in time $O(n \log r)$.*

Proof. We may assume without loss of generality that $r \leq n$. Sort the r parallel lines in the direction perpendicular to them. Form a binary tree T that cuts P into the $r+1$ monotone histograms assigning the middle chord to the root of T, and then, for each intermediate node v of T, recursively set $s(v)$ to the middle chord of the chords cutting the polygon $P(v)$ (see Definition 1). Now, by Theorem 4, the thesis follows from the fact that T has $O(\log r)$ height. □

.

Corollary 3 *Let P be a simple polygon with r reflex angles. The bounded Voronoi diagram of P can be computed in time $O(n \log r)$.*

Proof. Assign weight 1 to each vertex of P at a reflex angle and weight 0 to the remaining vertices of P. Recall Chazelle's theorem on polygon cutting in its general weighted form.[2,8] Given a triangulation of P, we can find a diagonal that splits P into two sub-polygons, each of total weight not greater than two thirds of the total weight of P plus one, and also the whole family of such diagonals for sub-polygons recursively created, everything in linear time. The above family of diagonals immediately induces a binary tree T of height $O(\log r)$ that cuts P into sub-polygons with $O(1)$ reflex angles. As a triangulation of P can be constructed in linear time[3], the tree T can also be constructed in linear time. Each of the final sub-polygons with $O(1)$ reflex angles can be easily split into $O(1)$ convex parts, for instance by using at most two diagonals from the preprocessing triangulation per each reflex angle for the splitting. By augmenting T with $O(1)$ levels that corresponds to the cuts into the convex parts, we obtain a binary tree T' that cuts P into $O(r)$ convex polygons and has height $O(\log r)$. Now, the thesis follows from Corollary 1. □

6. Final remarks

Similarly as the Delaunay triangulation or the Voronoi diagram of a planar point set S corresponds to the lower part of the convex hull of $\mu(S)$ (see Fact 1), the so called furthest-site Delaunay triangulation or furthest-site Voronoi diagram of S correspond to the upper part of the convex hull of $\mu(S)$.[4] This combined with the proof of Theorem 1 via Fact 2 implies that the furthest-site Voronoi diagram of a monotone histogram can be computed in linear time.

We believe that our main result about computing the Voronoi diagram of a monotone histogram in linear time will lead to an $o(n \log n)$ time algorithm for computing the bounded Voronoi diagram of a histogram. In contrast to the case of the standard Voronoi diagram of a histogram, we do not know any non-trivial lower bound for the latter problem.

Acknowledgements

We thank unknown referees and Rolf Klein for valuable comments.

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 2 (1987), Springer Verlag.
2. B. Chazelle, A Theorem on Polygon Cutting with Applications. Proc. 23rd IEEE FOCS Symposium, 1982.
3. B. Chazelle, Triangulating a Simple Polygon in Linear Time. Proc. 31st IEEE FOCS Symposium, 1990.
4. H. Edelsbrunner, Algorithms in Combinatorial Geometry. EATCS Monographs on Theoretical Computer Science 10, Springer Verlag, Berlin (1987).
5. S. Fortune, A Sweepline Algorithm for Voronoi Diagrams, Algorithmica 2(1987), pp. 153-174.
6. L.J. Guibas and J. Stolfi, Primitives for the Manipulation of General Subdivisions and the Computation of Voronoi Diagrams. In ACM Trans. Graphics 4 (1985), 74-123.
7. D.T. Lee and A. Lin, Generalized Delaunay Triangulations for Planar Graphs. In Discrete and Computational Geometry 1 (1986), Springer Verlag, pp. 201-217.
8. A. Lingas, On Partitioning Polygons. Proc. 1st ACM Symposium on Computational Geometry, Baltimore, 1985.
9. A. Lingas, Voronoi Diagrams with Barriers and the Shortest Diagonal Problem. In Information Processing Letters 32(1989), pp. 191-198.
10. F.P. Preparata and M.I. Shamos, Computational Geometry, An Introduction. Texts

and Monographs in Theoretical Computer Science , Springer Verlag, New York (1985).

11. C. Wang and L. Schubert, An Optimal Algorithm for Constructing the Delaunay Triangulation of a Set of Line Segments. In Proc. 3rd ACM Symposium on Computational Geometry, Waterloo, pp. 223-232, 1987.
12. R. Seidel, Proving Lower Bounds for Certain Geometric Problems. In Machine Intelligence and Pattern Recognition 2, Computational Geometry, edited by G. Toussaint, North Holland 1985.
13. R. Seidel, Constrained Delaunay triangulations and Voronoi diagrams with obstacles. In Rep. 260, IIG-TU Graz, Austria, pp. 178-191.