

Weighted Voronoi Diagrams in the Maximum Norm

Günther Eder* and Martin Held†

*FB Computerwissenschaften, Universität Salzburg
5020 Salzburg, Austria
*geder@cs.sbg.ac.at
†held@cs.sbg.ac.at*

Received 1 February 2019

Accepted 22 May 2019

Published 25 October 2019

Communicated by J. S. B. Mitchell

We consider multiplicatively weighted points, axis-aligned rectangular boxes and axis-aligned straight-line segments in the plane as input sites and study Voronoi diagrams of these sites in the maximum norm. For n weighted input sites we establish a tight $\Theta(n^2)$ worst-case bound on the combinatorial complexity of their Voronoi diagram and introduce an incremental algorithm that allows its computation in $\mathcal{O}(n^2 \log n)$ time. Our approach also yields a truly simple $\mathcal{O}(n \log n)$ algorithm for solving the one-dimensional version of this problem, where all weighted sites lie on a line.

Keywords: Weighted voronoi diagrams; maximum norm; line segments; rectangles; incremental construction.

1. Introduction

1.1. Related work

In 1984, Aurenhammer and Edelsbrunner⁴ introduced an $\mathcal{O}(n^2)$ time algorithm to compute the multiplicatively weighted Voronoi diagram of n weighted point sites in the standard Euclidean metric. Their algorithm is optimal in the worst case since this diagram may consist of $\Theta(n^2)$ faces, edges and nodes. Both their analysis of the combinatorial complexity as well as their algorithm rely on the fact that the bisector between a pair of weighted points is given by a circle. Aurenhammer and Edelsbrunner define spheres on these bisector circles and convert them into half-planes using a spherical inversion. Later Aurenhammer uses divide&conquer

This is an Open Access article published by World Scientific Publishing Company. It is distributed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 (CC BY-NC-ND) License which permits use, distribution and reproduction, provided that the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

†Corresponding author.

to obtain an $\mathcal{O}(n \log n)$ time and $\mathcal{O}(n)$ space algorithm for the one-dimensional weighted Voronoi diagram, where all weighted input points lie on a line.³

Papadopoulou and Lee⁶ formally study Voronoi diagrams of line segments in the maximum norm (L_∞ -metric). They apply a sweep-line algorithm to compute such a diagram in $\mathcal{O}(n \log n)$ time for n line segments. Vyatkina and Barequet⁹ show that multiplicatively weighted Voronoi diagrams (in the Euclidean metric) of n lines have a combinatorial complexity of $\mathcal{O}(n^2)$ and can be computed by means of wavefront propagation in $\mathcal{O}(n^2 \log n)$ time. Lower-envelope computations can be used to compute (weighted) Voronoi diagrams in $\mathcal{O}(n^{2+\varepsilon})$ time, where $\varepsilon > 0$.^{1,8} Papadopoulou and Xu⁷ show how to compute the Voronoi diagram of point sites under the L_∞ Hausdorff distance. Their two-pass plane sweep approach takes $\mathcal{O}(n \log n)$ time and linear space if a specific crossing number is at most linear. In prior work Papadopoulou⁵ shows that this diagram is equivalent to the additively weighted Voronoi diagram in the maximum norm.

1.2. Our contribution

We pick up the lead and investigate multiplicatively weighted Voronoi diagrams in the maximum norm. As input sites we consider weighted points and weighted axis-aligned rectangular boxes and straight-line segments. Similar to the Euclidean setting, the multiplicatively weighted Voronoi diagram subdivides the plane into regions such that every point in the interior of a region is closest to exactly one site. In Figs. 1(a) and 1(b), we illustrate the diagrams in both the L_2 - and L_∞ -norm over the same set of points and weights.

In Sec. 3, we establish a tight $\Theta(n^2)$ bound on the combinatorial complexity of the multiplicatively weighted Voronoi diagram of a set of n sites in the maximum norm. This worst-case bound is the same in the Euclidean metric and is shown

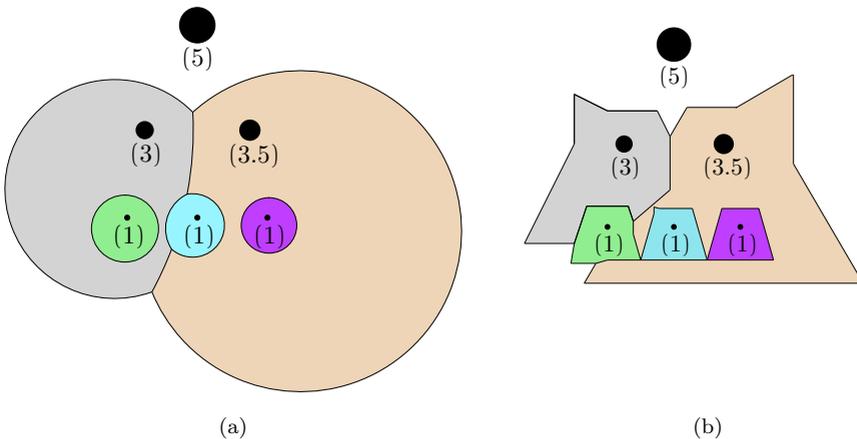


Fig. 1. Multiplicatively weighted Voronoi diagram of six point sites, associated weight in brackets; (a) in L_2 -norm; (b) in L_∞ -norm.

by Aurenhammer and Edelsbrunner.⁴ We explain how to apply a simple incremental construction approach that allows to compute the actual Voronoi diagram in $\mathcal{O}(n^2 \log n)$ time and $\mathcal{O}(n^2)$ space in Sec. 4.

As a by-product of our investigation we obtain an $\mathcal{O}(n \log n)$ algorithm for computing weighted Voronoi diagrams in one dimensions, where all input points lie on a line; cf. Sec. 6. While we cannot beat this worst-case optimal time bound, our algorithm is considerably simpler than Aurenhammer's divide&conquer solution.³

2. Basics and Characterization

We start with formulating the multiplicatively weighted Voronoi diagram in the L_∞ norm. Let $S := \{s_1, s_2, \dots, s_n\}$ denote a finite set of n distinct points, axis-aligned rectangular boxes and axis-aligned straight-line segments, so-called *sites*, in \mathbb{R}^2 . No two sites of S are allowed to share a common point. Since a box is regarded as a (compact) two-dimensional subset of the plane this assumption also rules out one site of S being contained inside another box of S . A weight function $w : S \rightarrow \mathbb{R}^+$ assigns a strictly positive weight $w(s)$ to every site $s \in S$. For reasons that will soon become apparent we start with assuming that all weights are distinct. (We will waive this assumption in Sec. 5.)

The weighted L_∞ -distance $d_w(p, s)$ from an arbitrary point p in \mathbb{R}^2 to a site $s \in S$ is given by the standard L_∞ -distance $d(p, s)$ from p to s divided by the weight of s . If p belongs to s , where s is a line segment or a box, then $d_w(p, s) := 0$. Otherwise,

$$d_w(p, s) := \frac{1}{w(s)} \cdot \min_{q \in s} d(p, q). \tag{1}$$

Ofcourse, if s is a point site then $d_w(p, s)$ results in the weighted point-to-point L_∞ -distance.

The multiplicatively weighted Voronoi Diagram $\mathcal{V}^\infty(S)$ of S is a subdivision of the plane formed by regions, edges and nodes. For s_i in S , the (*weighted*) *Voronoi region* $\mathcal{R}(s_i)$ of s_i is the set of all points of the plane that are not closer to any other site s_j in S , that is

$$\mathcal{R}(s_i) := \{p \in \mathbb{R}^2 : d_w(p, s_i) \leq d_w(p, s_j) \text{ for all } s_j \in S \text{ with } i \neq j\}.$$

As usual, the multiplicatively weighted Voronoi diagram $\mathcal{V}^\infty(S)$ of S is defined as

$$\mathcal{V}^\infty(S) := \bigcup_{s \in S} \partial \mathcal{R}(s),$$

where $\partial \mathcal{R}(s)$ denotes the boundary of $\mathcal{R}(s)$. A *face* of $\mathcal{V}^\infty(S)$ is one connected component of such a region. Note that a weighted Voronoi region may consist of many connected components; see Sec. 3. Thus, several faces may belong to one Voronoi region.

The *bisector* $b(s_i, s_j)$ for the two distinct sites s_i, s_j of S models the set of points of the plane that are at the same weighted distance from s_i and s_j . Hence, a

non-empty intersection of two Voronoi regions is a subset of the bisector of the two defining sites. Following common terminology, a connected component of such an intersection is called a (Voronoi) edge of $\mathcal{V}^\infty(S)$. An endpoint of an edge is called a (Voronoi) node.

The set of points in the plane that are at weighted distance t from a site s in S is given by the boundary of the Minkowski sum of s with an axis-aligned square with side length $2t \cdot w(s)$ that is centered at the origin. For $i \in \{1, 2, \dots, n\}$ and $s_i \in S$, we denote this axis-aligned rectangle by $\square_i(t)$. (We are only interested in the boundary of the rectangle rather than in the region bounded by it.) Ofcourse, the rectangle $\square_i(t)$ is an axis-aligned square whose center coincides with s_i if s_i is a point site. Let $\mathcal{U}(t) := \{\square_1(t), \dots, \square_n(t)\}$ denote the set of these n rectangles for $t > 0$. We find it convenient to regard $\mathcal{U}(t)$ as a function of either time or distance since at time t every point p on $\square_i(t)$ is at L_∞ -distance $t \cdot w(s_i)$ from s_i , i.e., at weighted distance t .

We start with investigating the bisector of two distinct sites s_i and s_j in S . Consider a distance $t > 0$ that is small enough such that neither $\square_i(t)$ intersects $\square_j(t)$ nor one of them is contained inside the other. Now let t increase. A non-empty intersection of the two rectangles $\square_i(t)$ and $\square_j(t)$ comprises all points that are at weighted distance t from both s_i and s_j . The first intersection of the two expanding rectangles corresponds to a single point or a line segment. When expanding the rectangles further the bisector between s_i and s_j is traced out along the intersection $\square_i(t) \cap \square_j(t)$. Hence, by taking the union of all points of intersection between $\square_i(t)$ and $\square_j(t)$, for all $t > 0$, we obtain the bisector of s_i and s_j . In Fig. 2, we show such an expansion scenario for the squares of two point sites, and Fig. 3(a) shows the bisector between two box sites. We note that $\square_i(t)$ will be fully contained in $\square_j(t)$ for all values of t that are large enough if $w(s_i) < w(s_j)$. Hence, the bisector between s_i and s_j is a closed curve if $w(s_i) \neq w(s_j)$.

Recall that in the standard L_∞ setting a bisector of two unweighted point sites need not be a curve but may be a two-dimensional subset of the plane: Consider two point sites s_i and s_j that have the same y -coordinate and the same weight. At time t their respective squares $\square_i(t)$ and $\square_j(t)$ meet for the first time, i.e., two vertical

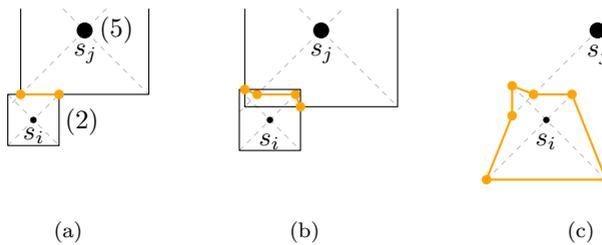


Fig. 2. (Color online) Sample expanding squares for two weighted point sites s_i, s_j ; the radii of the disks that mark the sites are chosen in relation to their weights (in parentheses). Their bisector is shown in orange.

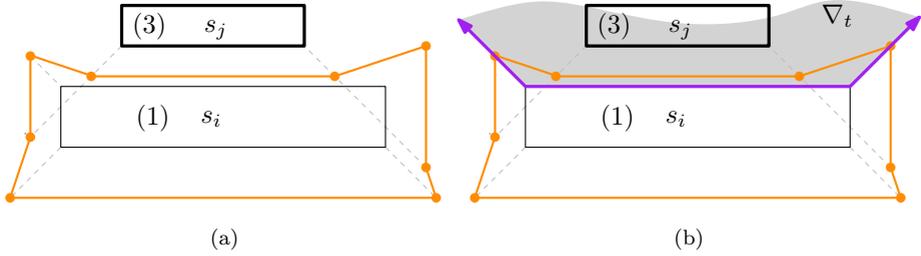


Fig. 3. (Color online) (a) Bisector (orange) between two boxes whose site weights are noted in parentheses. (b) The top wedge ∇_t traced out by the top side of the expanding rectangle of s_i .

sides (left and right, resp.) overlap. (Ofcourse, that time t equals half the distance of the x -coordinates of s_i and s_j .) Due to equal weight and equal y -coordinate the top sides of the rectangles are collinear, as are the bottom sides. Therefore, at time $t' \geq t$ these sides overlap partially, and this overlap expands as t' increases. Hence, the bisector of these two sites resembles an hourglass, cf. Fig. 5(a).

We now argue that this cannot happen if all weights are distinct. Assume that $w(s_i) < w(s_j)$. The intersection of $\square_i(t)$ and $\square_j(t)$ can form a line segment if and only if two sides of the two rectangles lie on the same supporting line. The simple but crucial insight is that this can happen at most once for every pair of parallel sides: Even if two sides of \square_i and \square_j propagate in the same direction then the distinct weights result in distinct speeds and ensure that the moving copies of the sides are contained in the same line at most once.

Hence, except for a constant number of values of t , the intersection of $\square_i(t)$ and $\square_j(t)$ is formed by at most two discrete points. Standard analysis shows that all points of intersection of a vertical and a horizontal line that propagate at constant speeds lie on a line. Thus, if two non-parallel sides of $\square_i(t)$ and $\square_j(t)$ intersect then their point of intersection traces out a line segment.

Let us now analyze the bisector between two box sites s_i and s_j in more detail. Again assume that $w(s_i) < w(s_j)$. Every side of the expanding rectangle $\square_i(t)$ moves within precisely one face of the exterior straight skeleton of the box. If s_i is a point site (rather than a box) then every such face is formed by a right-angled wedge whose two bounding rays originate at the site. For convenience we refer to such a face as wedge also in the case of rectangular sites. Let ∇_t, ∇_b and ∇_l, ∇_r be the top (bottom, resp.) and left (right, resp.) wedge; see Fig. 3(b). These four wedges cover all of the exterior of s_i .

Now assume that the top side of s_j lies above the top side of s_i . Ofcourse, both sides move upwards as t increases. However, our assumption on the weights implies that the top side of $\square_i(t)$ cannot catch up with the top side of $\square_j(t)$. A similar argument holds if the right side of $\square_j(t)$ lies to the right of $\square_i(t)$, etc. We call this fact the *no-piercing* property: The expanding rectangle of a lower-weighted site cannot pierce through an expanding rectangle of a higher-weighted site.

Lemma 1. *Let s_i, s_j be two sites in S and let $w(s_i) < w(s_j)$. Then at least one side of $\square_j(t)$ is never intersected by any side of $\square_i(t)$, for all $t > 0$.*

This no-piercing property implies that every side of $\square_i(t)$ can be intersected by at most three sides of $\square_j(t)$.

Lemma 2. *Let s_i, s_j be two sites in S and let $w(s_i) < w(s_j)$. Then the bisector between s_i and s_j restricted to ∇_t is formed by up to three straight-line segments. If there are three line segments then the middle segment is parallel to the top side of s_i .*

Similar statements hold for $\nabla_b, \nabla_l, \nabla_r$. This lemma implies that there are at most four times at which the intersection of $\square_i(t)$ and $\square_j(t)$ forms a line segment. A more refined analysis of the expansion of \square_i and \square_j shows that at most three of all pairs of moving parallel sides can actually overlap. E.g., in Fig. 2(c) we see that the bisector contains three axis-aligned segments. Clearly an overlap between the top side of $\square_i(t)$ and the bottom side of $\square_j(t)$ and an overlap between the right side of $\square_i(t)$ and the left side of $\square_j(t)$ is not possible unless they occur at the same time and degenerate to a corner-corner contact. Furthermore, a bisector polygon encloses the site with smaller weight. We call the vertices of this bisector polygon *Voronoi joints*. It is easy to see that all interior angles of a bisector polygon are less than $3\pi/4$. This implies that the bisector polygon between s_i and s_j restricted to ∇_t is a polygonal chain that is monotone with respect to the x -axis.

Lemma 3. *Let s_i, s_j be two sites in S and let $w(s_i) < w(s_j)$. Then the bisector between s_i and s_j is a polygon with a constant number of edges and joints. It consists of two polygonal chains which are strictly monotone with respect to the x -axis and two polygonal chains which are strictly monotone with respect to the y -axis. Each polygonal chain lies within exactly one of the four wedges $\nabla_t, \nabla_b, \nabla_l, \nabla_r$ and consists of at most three edges. The two chains in ∇_t, ∇_b each contain at most one horizontal edge, while the chains in ∇_l, ∇_r each contain at most one vertical edge.*

Since $\mathcal{V}^\infty(S)$ is formed by portions of bisectors, $\mathcal{V}^\infty(S)$ is a planar straight-line graph. It contains Voronoi joints as vertices of degree two, and Voronoi nodes as vertices of degree three or higher. We note that distinct weights prevent $\mathcal{V}^\infty(S)$ from containing rays, i.e., unbounded edges: Let s_j be the site of S with maximum weight. Then there exists a time t_j such that $\square_j(t)$ will contain the scaled rectangles of all other sites for all $t > t_j$. Thus, the Voronoi region of s_j is the only unbounded region.

From now on we assume that the sites s_1, \dots, s_n of S are indexed according to decreasing weight. That is, $w(s_i) > w(s_{i+1})$, for $1 \leq i < n$. Let $S_i := \{s_1, \dots, s_i\}$ denote the i highest-weighted sites of S , for $1 \leq i \leq n$. Lemma 3 implies that the bisector between s_i and any other s_j in S_i is a simple polygon that consists of four monotone polygonal chains, with each chain within precisely one of the

four wedges. And, of course, each bisector polygon contains s_i within its interior. Since the Voronoi region $\mathcal{R}(s_i)$ of s_i relative to the other sites of S_i is given by the intersection of these $i - 1$ polygonal areas we get a fairly simple way to obtain $\mathcal{R}(s_i)$.

Lemma 4. *Let $1 \leq i \leq n$. For $\nabla \in \{\nabla_t, \nabla_b, \nabla_l, \nabla_r\}$, the boundary of the Voronoi region $\mathcal{R}(s_i)$ of s_i relative to S_i restricted to ∇ can be obtained by a lower-envelope-like computation that involves $i - 1$ polygonal chains which all are strictly monotone relative to the same coordinate axis and which all have their end-points on the rays bounding ∇ .*

3. Combinatorial Complexity

Aurenhammer and Edelsbrunner⁴ show that a multiplicatively weighted Voronoi diagram of points in the Euclidean metric can have $\Omega(n^2)$ faces, edges, and nodes. The set-up of their example can be adapted easily to our setting, see Fig. 4: We place $\lfloor n/2 \rfloor$ point sites with large (identical) weight at close range on a line with a slope of one. The bisectors of these sites are straight lines. Hence, these point sites define $\lfloor n/2 \rfloor$ faces which form (narrow) slabs. The remaining $\lfloor n/2 \rfloor$ point sites are placed along the middle bisector, and their (relatively) small weights are chosen large enough such that their Voronoi regions overlap with all slabs. Hence, every slab is partitioned into roughly $n/2$ faces, and we get a total of $\Omega(n^2)$ faces for $\mathcal{V}^\infty(S)$.

Ofcourse, we could modify the large weights of the first $\lfloor n/2 \rfloor$ point sites such that all weights are very similar but distinct. It is easy to see that the key combinatorial properties of our set-up will remain unchanged providing that these sites are chosen close enough together and that their weights are chosen large enough relative to the weights of the other sites. Similarly, every point site could be replaced by a small box or straight-line segment.

We now establish that $\mathcal{O}(n^2)$ is also a tight upper bound on the combinatorial complexity of $\mathcal{V}^\infty(S)$. Lemma 4 tells us that the boundary of $\mathcal{R}(s_i)$ consists of four monotone polygonal chains which can be obtained by running a lower-envelope computation within each of the four wedges. In particular, Lemma 2 tells us that the bisector between s_i and s_j , for $1 \leq j < i$, restricted to ∇_t consists of at most one horizontal and at most two non-horizontal edges. We interpret a non-horizontal bisector edge that has an end-point on one of the rays of ∇_t as unbounded

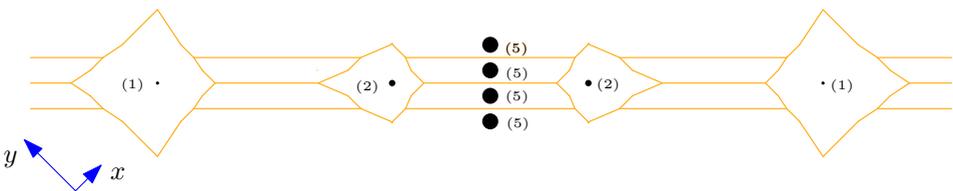


Fig. 4. (Color online) $\mathcal{V}^\infty(S)$ of eight point sites, weights denoted by the numbers in brackets. Note that the figure is rotated to save space; see the coordinate system (blue).

and replace it by a ray that extends to the exterior of ∇_t . (Replacing every non-horizontal edge by a ray will allow us a simplified analysis of the combinatorial complexity.) Hence, the portion of the bisector between s_i and s_j restricted to ∇_t is a subset of at most two rays and at most one horizontal edge. Summing over all $i - 1$ sites of S_i gives at most $2(i - 1)$ many rays and at most $i - 1$ many horizontal edges whose lower envelope \mathcal{L} contains the bisector between s_i and s_j restricted to ∇_t .

Let \mathcal{L}_r be the lower envelope of all $2(i - 1)$ many rays, and let \mathcal{L}_h be the lower envelope of all horizontal edges. Then \mathcal{L} is given by the lower envelope of \mathcal{L}_r and \mathcal{L}_h . Ofcourse, both \mathcal{L} and \mathcal{L}_r are x -monotone polygonal chains. A straightforward charging argument used in a left-to-right scan of the chains allows to establish that the lower envelope of two (infinite) x -monotone polygonal chains with m_1 and m_2 vertices has at most $2(m_1 + m_2)$ vertices.

It is also easy to see that the lower envelope of all at most $i - 1$ rightwards rays has at most $2(i - 1)$ many (finite) vertices. Hence, \mathcal{L}_r contains at most $8(i - 1)$ vertices. (This bound also follows from a more general result by Alevizos *et al.*² on the complexity of the boundary of the external region in an arrangement of rays.) Another straightforward charging argument yields that \mathcal{L}_h has at most $2i$ vertices. We conclude that \mathcal{L} has at most $2 \cdot (8 + 2) \cdot i$ vertices, thus establishing the following lemma. (We do not suggest that the constants involved are smallest possible, though.)

Lemma 5. *The Voronoi region $\mathcal{R}(s_i)$ of the site s_i relative to S_i consists of at most $80i$ edges.*

The combinatorial complexity $|\mathcal{V}^\infty(S_i)|$ of $\mathcal{V}^\infty(S_i)$ depends on $|\mathcal{V}^\infty(S_{i-1})|$ plus the combinatorial complexity of $\mathcal{R}(s_i)$. After all, the forming of $\mathcal{V}^\infty(S_i)$ from $\mathcal{V}^\infty(S_{i-1})$ by inserting $\mathcal{R}(s_i)$ causes the generation of at most one new Voronoi edge per joint of $\mathcal{R}(s_i)$, while some Voronoi edges/nodes of $\mathcal{V}^\infty(S_{i-1})$ in the interior of $\mathcal{R}(s_i)$ get removed completely. This observation implies that $\mathcal{V}^\infty(S_i)$ has at most $2 \cdot 80 \cdot i$ more Voronoi edges than $\mathcal{V}^\infty(S_{i-1})$. Hence,

$$|\mathcal{V}^\infty(S)| \leq \sum_{1 \leq i \leq n} 160 \cdot i = 80 \cdot n \cdot (n + 1),$$

thus settling Theorem 1.

Theorem 1. *The multiplicatively weighted Voronoi diagram of a set of n weighted sites in the L_∞ norm consists of $\Theta(n^2)$ faces, edges, joints and nodes in the worst case.*

4. Algorithm

We are now ready to provide details of the incremental construction algorithm to compute $\mathcal{V}^\infty(S)$ which we hinted at in the previous section. Again, let the sites s_1, \dots, s_n of S be indexed according to decreasing weight, and let $S_i := \{s_1, \dots, s_i\}$

for $1 \leq i \leq n$. We proceed with showing inductively how the Voronoi region $\mathcal{R}(s_i)$ of s_i within S_i can be computed and inserted into $\mathcal{V}^\infty(S_{i-1})$ to obtain $\mathcal{V}^\infty(S_i)$.

There are several straightforward ways to compute the four lower envelopes required according to Lemma 4 in $\mathcal{O}(i \log i)$ time. We employ a divide&conquer approach: We split S_{i-1} into two subsets S'_{i-1} and S''_{i-1} of roughly equal size and recursively compute the corresponding envelopes for the bisectors between s_i and the sites of S'_{i-1} resp. S''_{i-1} . Theorem 5 tells us that each resulting polygonal chain has $\mathcal{O}(i/2) = \mathcal{O}(i)$ vertices and edges. The actual merge step is carried out similar to the merge step of merge sort and takes time linear in the number of vertices of the two chains. Hence, $\mathcal{R}(s_i)$ is computed in $\mathcal{O}(i \log i)$ time.

The second step is to embed $\mathcal{R}(s_i)$ into $\mathcal{V}^\infty(S_{i-1})$ in order to obtain $\mathcal{V}^\infty(S_i)$. A node v of $\mathcal{R}(s_i)$ is defined by exactly three sites s_i, s_n, s_m . Therefore, we have to modify the bisector between s_n and s_m . We know that the bisector between two sites consists of a constant number of edges. Unfortunately, in $\mathcal{V}^\infty(S_{i-1})$ a single bisector may contribute $\mathcal{O}(i)$ many Voronoi edges. Therefore, we set up a search structure that enables us to find a specific Voronoi edge of a bisector quickly. Suppose that an edge of the bisector $b(s_i, s_j)$, with $1 \leq j < i$, belongs to the boundary of $\mathcal{R}(s_i)$. Then we maintain such a search structure for the supporting line of that edge and store it with s_i . A standard self-balancing binary search tree, e.g., an AVL tree, is good enough for our purposes. It guarantees $\mathcal{O}(\log n)$ processing time for all operations required. Theorem 5 tell us that we store $\mathcal{O}(i)$ many search structures with s_i . The total memory required for all search structures is bounded by the number of edges of the Voronoi diagram, i.e., by $\mathcal{O}(n^2)$.

To embed $\mathcal{R}(s_i)$ we iterate over its nodes, i.e., those vertices v of $\mathcal{R}(s_i)$ which have equal weighted distance to s_i and two other sites s_m and s_n . We query our search structure for $b(s_m, s_n)$ and find the Voronoi edge e of $\mathcal{V}^\infty(S_{i-1})$ that contains v . We split e at v and update the search structure. Note that joints of $\mathcal{R}(s_i)$ can be added without further attention. Hence we can embed $\mathcal{R}(s_i)$ into $\mathcal{V}^\infty(S_{i-1})$ in $\mathcal{O}(i \log i)$ time. If $\mathcal{R}(s_i)$ does not contain nodes (but only joints) then it is formed by just one bisector $b(s_i, s_j)$ and lies completely in the interior of a face of $\mathcal{R}(s_j)$. In this case it does not intersect an edge of $\mathcal{V}^\infty(S_{i-1})$ but forms a disconnected face.

The final step is to delete the edges of $\mathcal{V}^\infty(S_{i-1})$ that are contained in the interior of $\mathcal{R}(s_i)$. As $\mathcal{R}(s_i)$ is now embedded in the diagram we apply a breath-first search originating from the nodes of $\mathcal{R}(s_i)$ towards the interior of the region. This search identifies all k_i edges inside of $\mathcal{R}(s_i)$. Removing these k_i edges takes $\mathcal{O}(k_i \log n)$ time due to the cost of maintaining the search structures. Note that a newly added region $\mathcal{R}(s_i)$ cannot contain a disconnected face of $\mathcal{V}^\infty(S_{i-1})$ completely. (Otherwise, one site would have no Voronoi region!)

We claim that $K := \sum_{1 < i \leq n} k_i = \mathcal{O}(n^2)$. We know that $\mathcal{R}(s_i)$ has $\mathcal{O}(i)$ edges and nodes, cf. Theorem 5. Furthermore, we know that most $\mathcal{O}(i)$ edges are added due to the insertion of $\mathcal{R}(s_i)$. Trivially, the number k_i of edges removed is bound

by the number of edges ever created. Hence, $\sum_{1 < i \leq n} k_i \leq \sum_{1 < i \leq n} \mathcal{O}(i) = \mathcal{O}(n^2)$ holds and settles our claim.

We are now ready to summarize the complexities of the steps of the algorithm: Computing the polygon $\mathcal{R}(s_i)$ and inserting it into $\mathcal{V}^\infty(S_{i-1})$ each take $\mathcal{O}(i \log n)$ time. Thus, creating and inserting all n Voronoi regions requires $\mathcal{O}(n^2 \log n)$ time. Since $K = \mathcal{O}(n^2)$ the overall time required by the removal of Voronoi edges is in $\mathcal{O}(n^2 \log n)$, thus establishing Theorem 2.

Theorem 2. *The multiplicatively-weighted Voronoi diagram $\mathcal{V}^\infty(S)$ of a set S of n weighted sites in the L_∞ norm can be computed incrementally in $\mathcal{O}(n^2 \log n)$ time and $\mathcal{O}(n^2)$ space.*

Unfortunately, there is no hope to tune this approach for achieving output sensitivity: Suppose that Fig. 4 shows the intermediate Voronoi diagram after the insertion of $n/3$ point sites of high weight and $n/3$ point sites of low weight. Then we can cover the $\mathcal{O}(n^2)$ slab-like faces by inserting the remaining $n/3$ (point) sites, with the Voronoi region of one site covering $\mathcal{O}(n)$ slab-like faces. Thus, the final Voronoi diagram might be of linear combinatorial complexity even if intermediate diagrams are of quadratic complexity.

5. Equal Weights

We know that equal weights can result in two-dimensional bisectors, as explained in Sec. 2. Papadopoulou and Lee⁶ show this in their work and decide to define the bisector in such a case as follows: They use the left lower boundary ray on the left lower end of the initial “bisector” and the right upper boundary ray on the right upper end, cf. Fig. 5. We adopt their convention in the case of point sites with identical weights and common x - or y -coordinates.

Similar “two dimensional” bisectors can occur for rectangular boxes or straight-line segments if the weights are identical. For instance, we could replace the right point site in Fig. 5(a) by a horizontal line segment and get a similar bisector. Fortunately, Papadopoulou and Lee’s convention remains applicable.

We now revisit our complexity claims. The no-piercing property, Lemma 1, holds for equal weights as well, since the equal weights imply equal speeds of the respective

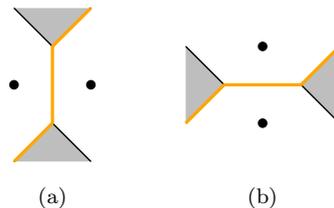


Fig. 5. (Color online) (a), (b) A “two dimensional” bisector area (gray) between two point sites, and the resulting bisector adopted by convention (orange).

expanding rectangles. Lemma 2 holds for equal weights as well, as do Lemma 3–Theorem 5 and Theorem 2. Hence the combinatorial complexity of $\mathcal{V}^\infty(S)$ stays in $\Theta(n^2)$. It is easy to realize that our algorithms remains applicable, too. Note, though, that $\mathcal{V}^\infty(S)$ has a tight linear bound on its combinatorial complexity if absolutely all weights are identical.⁶

6. One Dimensional Weighted Voronoi Diagram

Consider a set S of n point sites in \mathbb{R} where every site s has a strictly positive weight $\sigma(s) \in \mathbb{R}^+$. Aurenhammer³ shows that $\mathcal{V}(S)$ has a linear combinatorial complexity by modeling it as the lower envelope of wedges in \mathbb{R}^2 . The actual Voronoi diagram is computed in $\mathcal{O}(n \log n)$ time by means of divide&conquer and a plane sweep.

We now apply our incremental construction algorithm to compute $\mathcal{V}(S)$. Again, let $S_i := \{s_1, \dots, s_i\}$ be the subset of the i highest-weighted sites of S and let $\mathcal{V}(S_i)$ be their Voronoi diagram. We know that $\mathcal{R}(s_i)$ is a single interval because $\mathcal{R}(s_i)$ is formed by the intersection of $i - 1$ intervals which all contain s_i .

We maintain the sequence of interval boundaries that correspond to the Voronoi diagram under construction in a balanced binary search tree T that supports search, insertion and deletion in logarithmic time. To compute $\mathcal{R}(s_i)$ we use T to determine the interval that contains s_i . Then we scan leftwards and rightwards from s_i . If an interval boundary encountered during this scan is closer to s_i than to the two sites of S_{i-1} which defined it then it is deleted from T . Eventually the left and right boundaries of $\mathcal{R}(s_i)$ are inserted into T . Since $\mathcal{R}(s_i)$ adds exactly one interval and splits at most one existing interval into two sub-intervals we know that $|\mathcal{V}(S_i)| = \mathcal{O}(i)$. Trivially, we can only remove intervals which were inserted previously. Hence the overall number of intervals during the entire incremental construction is bound by $\mathcal{O}(n)$, which settles the following theorem.

Theorem 3. *Maintaining a balanced binary search tree suffices to compute the multiplicatively weighted Voronoi diagram of n weighted point sites in \mathbb{R} by means of incremental construction in optimal $\mathcal{O}(n \log n)$ time and $\mathcal{O}(n)$ space.*

7. Extensions

The assumption that input line segments of S may not share common end-points can be waived without causing our approach to collapse. Hence, our approach can also compute multiplicatively weighted Voronoi diagrams of planar orthogonal straight-line graphs (where all edges are parallel to coordinate axes). Similarly, we may allow input rectangles to share corners. Note, though, that the Voronoi region of a point site will be empty if it coincides with a corner of a rectangle or end-point of a straight-line segment which has a higher weight.

It is also easy to incorporate additive weights because this merely means replacing the input sites by properly enlarged rectangles: If site s_i has additive weight a_i then we replace it by the Minkowski sum of s_i with an axis-aligned square with

side-length $2a_i$ that is centered at the origin. Our approach remains applicable as long as no pair of the resulting rectangular boxes intersect in their relative interiors.

As a matter of principle, our approach could also be generalized to rectangular boxes and straight-line segments that are not required to be axis-parallel. In this case the expanding offset structures become octagons which have four sides that are parallel to the coordinate axes. This results in several technical issues that require a careful re-phrasing of our lemmas. But even in this general case the bisector between two sites is a polygon of constant complexity, and the weighted Voronoi diagram could be constructed by our incremental algorithm. However, we have not been able to prove that the complexity of the Voronoi cell of s_i relative to S_i remains linear in i . Rather, we can only prove a worst-case complexity of $\mathcal{O}(i \cdot \alpha(i))$, where $\alpha(\cdot)$ denotes the inverse Ackerman function. As a consequence, an additional multiplicative factor of $\alpha(n)$ shows up both in our worst-case bound on the combinatorial complexity of the Voronoi diagram of n sites as well as in the time complexity of our algorithm if we allow the sites to be oriented arbitrarily. We leave it as a problem for future work to either get rid of this multiplicative factor or to establish that $\Theta(n^2 \cdot \alpha(n))$ is indeed the worst-case combinatorial complexity of the Voronoi diagram in this general setting.

Acknowledgments

This work was supported by Austrian Science Fund (FWF): Grant ORD 53-VO. We thank Peter Palfrader for interesting discussions of this problem.

References

1. P. K. Agarwal, O. Schwarzkopf and M. Sharir, The overlay of lower envelopes and its applications, *Discr. Comput. Geom.* **15**(1) (1996) 1–13.
2. P. Alevizos, J.-D. Boissonnat and F. P. Preparata, An optimal algorithm for the boundary of a cell in a union of rays, *Algorithmica* **5**(1) (1990) 573–590.
3. F. Aurenhammer, The one-dimensional weighted Voronoi diagram, *Inform. Process. Lett.* **22**(3) (1986) 119–123.
4. F. Aurenhammer and H. Edelsbrunner, An optimal algorithm for constructing the weighted Voronoi diagram in the plane, *Pattern Recogn.* **17**(2) (1984) 251–257.
5. E. Papadopoulou, Critical area computation for missing material defects in VLSI circuits, *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems* **20**(5) (2001) 583–597.
6. E. Papadopoulou and D. Lee, The L_∞ Voronoi diagram of segments and VLSI applications, *Int. J. Comput. Geom.* **11**(05) (2001) 503–528.
7. E. Papadopoulou and J. Xu, The L_∞ Hausdorff Voronoi diagram revisited, *Int. J. Comput. Geom. Appl.* **25**(02) (2015) 123–141.
8. M. Sharir, Almost tight upper bounds for lower envelopes in higher dimensions, *Discr. Comput. Geom.* **12**(3) (1994) 327–345.
9. K. Vyatkina and G. Barequet, On multiplicatively weighted Voronoi diagrams for lines in the plane, *Transactions on Computational Science XIII*, eds. M. Gavrilova and C. Tan, LNCS 6750 (Springer-Verlag, 2011), pp. 44–71.