Farthest Line Segment Voronoi Diagrams*

F. Aurenhammer† R. L. S. Drysdale‡ H. Krasser§

Abstract

The farthest line segment Voronoi diagram shows properties different from both the closest-segment Voronoi diagram and the farthest-point Voronoi diagram. Surprisingly, this structure did not receive attention in the computational geometry literature. We analyze its combinatorial and topological properties and outline an $O(n \log n)$ time construction algorithm that is easy to implement. No restrictions are placed upon the n input line segments; they are allowed to touch or cross.

Keywords: Computational geometry; Voronoi diagram; farthest line segment; optimal algorithm.

1 Introduction

Consider a set S of n simple geometric objects (called sites) in the plane, for example points, line segments, or circular arcs. The closest-site Voronoi diagram of S subdivides the plane into regions, each region being associated with some site $s_i \in S$, and containing all points of the plane for which s_i is closest among all the sites in S. Voronoi diagrams and their numerous variants have proven extremely useful in the algorithmic and combinatorial analysis of geometric problems, and are a well-established tool in computational geometry [2, 4, 17].

It is commonly agreed that many geometric scenarios can be modeled with sufficient accuracy by polygonal objects. In this sense, the Voronoi diagram for line segment sites is of particular importance. Indeed, this type is among the first generalizations of the standard point-site Voronoi diagram to have been considered [12, 14, 9, 19], and several practical and efficient construction algorithms have been developed; see [10, 11] and references therein.

Along with the study of closest-site Voronoi diagrams go their farthest-site counterparts. In that model, each site $s_i \in S$ gets allotted the region of all points in the plane for which s_i is the farthest site (rather than the closest) in S. While geometric properties mainly stay unaffected by this modification, the combinatorial size of the diagram may change drastically in the worst case. For instance, this happens for Voronoi diagrams under the Manhattan metric [15] (from $\Theta(n)$ to $O(1)$), and for multiplicatively weighted Voronoi diagrams [3, 13] (from $\Theta(n^2)$ to $\Theta(n)$).

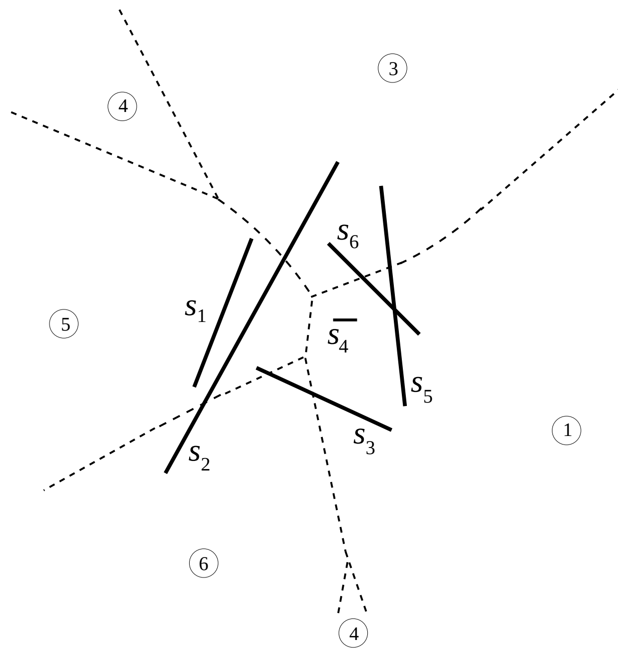

Figure 1: Farthest-segment Voronoi diagram

Interestingly, and surprising to the authors, the farthest line segment Voronoi diagram, which is the topic of the present note, has been treated as a stepchild in the vast Voronoi diagram literature. As part of a study of 2-site Voronoi diagrams [5], this type has been considered for all $\binom{n}{2}$ line segments determined by n points in the plane. Very recently, a divide-and-conquer algorithm for the case where the input line segments form a convex polygon has been given in [7]. However, nothing has been published about the farthest line segment Voronoi diagram in its general setting.

Still, the properties of this diagram deviate from the obvious. For example, regions may be disconnected, and region emptiness cannot be characterized by convex hull properties. Moreover, and unlike the closest line segment case, the number of edges and vertices of the diagram remains $\Theta(n)$ in the worst case, regardless of the crossing properties of the input segments. (It is well known that each crossing constitutes a vertex in the closest-segment diagram.) We give a structural analysis of the farthest line segment Voronoi diagram, and outline an $O(n \log n)$ con-

*Research supported by the FWF Joint Research Project 'Industrial Geometry', S9205-N12

†Institute for Theoretical Computer Science, University of Technology, Graz, Austria. `auren@igi.tugraz.at`

‡Department of Mathematics and Computer Science, Dartmouth Colleger, Hanover, U.S.A. Work done while this author visited the Institute for Theoretical Computer Science, University of Technology, Graz, Austria. `scot.drysdale@dartmouth.edu`

§Institute for Theoretical Computer Science, University of Technology, Graz, Austria. `hkrasser@igi.tugraz.at`

struction algorithm that is implementable using basic data structures.

Among its potential applications are computing the smallest disk that contacts all the given segments, and finding the largest gap to be bridged between any two segments. This information can be derived from the diagram without asymptotic increase of runtime. As another example, we may wish to preprocess a set S of n points so that, given a query point q, one can quickly report a line segment spanned by S and being farthest from q. The Voronoi diagram data structure in [6], together with the approach in [16], implies $O(n \log n)$ preprocessing time and $O(\log n)$ query time.

2 Structural properties

Let $S = \{s_1, \ldots, s_n\}$ be an arbitrary set of line segments in the Euclidean plane $\mathbb{R}^2$. Line segments in S are allowed to cross or may touch at single points. The distance $d(x, s_i)$ of a point $x \in \mathbb{R}^2$ to a line segment $s_i \in S$ is measured to the closest point on s_i. That is, $d(x, s_i) = \min\{\delta(x, p) \mid p \in s_i\}$, where δ denotes the Euclidean distance function. The region of a line segment s_i is defined as

$$reg(s_i) = \{x \in \mathbb{R}^2 \mid d(x, s_i) \geq d(x, s_j), 1 \leq j \leq n\}.$$

The regions of all the segments in S, together with their bounding edges and vertices, define a partition of $\mathbb{R}^2$ which is called the farthest-segment Voronoi diagram of S, or $FV(S)$ for short. Figure 1 displays this diagram for six line segments. Encirculated numbers indicate affiliation of regions to segments.

For any two distinct segments s_i and s_j, their regions $reg(s_i)$ and $reg(s_j)$ are separated by their bisector $sep(s_i, s_j)$, which is the locus of all points $x \in \mathbb{R}^2$ equidistant from s_i and s_j. It is well known [14, 11] that $sep(s_i, s_j)$ is composed of constantly many pieces of straight lines and parabolas. If s_i and s_j are disjoint then $sep(s_i, s_j)$ is an unbounded and connected curve. Two such curves, intersecting at point p, make up the bisector if s_i and s_j cross properly at p. Finally, if s_i and s_j have a common endpoint then $sep(s_i, s_j)$ contains a two-dimensional portion. This portion is, by convention, replaced by the piece of the angle bisector of s_i and s_j that it includes, and a single separating curve is obtained. Note that the $O(1)$ points delimiting the individual pieces of $sep(s_i, s_j)$ will not be considered vertices of $FV(S)$ in the present paper.

Define a face of $FV(S)$ as a maximal interior-connected subset of a region of $FV(S)$. As for the standard case of point sites, we have the following property.

Lemma 1 *All faces of $FV(S)$ are unbounded.*

Proof. For $x \in \mathbb{R}^2$ and $s_i \in S$, let p be the point on s_i closest to x. Then $x \in reg(s_i)$ holds if and only if the closed disk $D(x)$ with center x and p on its boundary intersects all the segments in $S \setminus \{s_i\}$. Let R be the ray starting at x and directed away from p. For all $y \in R$, we have $D(y) \supset D(x)$. Thus $R \subset reg(s_i)$ follows from $x \in reg(s_i)$. This implies that $reg(s_i)$ is either empty or all its faces are unbounded. □

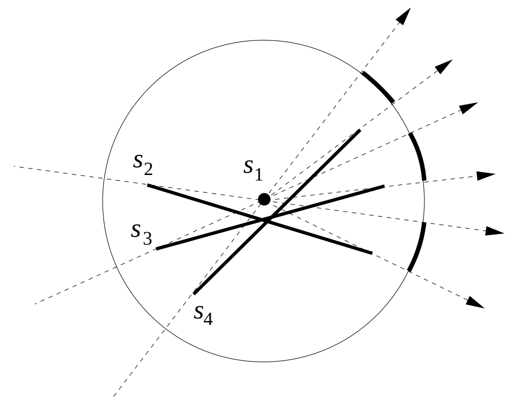

Figure 2: Directions where segment s_1 is active

Corollary 2 *The interior of $reg(s_i)$ is non-empty (and is unbounded in direction φ) if and only if there exists an open halfplane (normal to φ) which intersects all segments in S but s_i.*

Corollary 2 shows that emptiness of regions is not reflected by the extremal properties of the set S. For instance, both endpoints of segment s_2 in Figure 1 are extremal but $reg(s_2)$ is empty. On the other hand, segment s_4 avoids the boundary of the convex hull of S but still gives rise to a non-empty region. Observe that $reg(s_4)$ is disconnected and breaks into two faces.

Let us describe an example where the region of an individual segment consists of $\Theta(n)$ faces. See Figure 2. Segment s_1 degenerates to a point, and segments $s_2, \ldots, s_n$ are arranged around s_1 in a cyclic fashion. Now let a directed line g through s_1 rotate, and consider the open halfplane H to the left of g. Whenever g points at some direction not emphasized in bold, H intersects all segments except s_1. (Otherwise, H avoids another segment beside s_1.) Thus, by Corollary 2, $reg(s_1)$ is unbounded in the corresponding normal directions. On the other hand, for points x being sufficiently close to s_1 in any of these normal directions, s_1 will no longer be the segment in S farthest from x. We conclude that $reg(s_1)$ splits into $n-1$ faces, one for each interval of directions.

This property of $reg(s_1)$ can be maintained while untangling the crossings between segments: For $i = 3, \ldots, n$, we translate the segment s_i towards s_1 and shorten it so that it still spans the same angle as seen from s_1, until s_i does not intersect any of the segments $s_2, \ldots, s_{i-1}$.

The proof of the following assertion is postponed to Section 3. It holds for arbitrary (possibly non-disjoint) segments, and should be seen in contrast to the $\Theta(n^2)$ worst-case size of the closest-segment Voronoi diagram where segment crossings necessarily give rise to diagram vertices.

Theorem 3 *Let S be a set of n line segments in the plane. $FV(S)$ contains $O(n)$ faces, edges, and vertices.*

Another property distinguishes $FV(S)$ from its closest-segment counterpart.

Lemma 4 *The graph formed by the edges of $FV(S)$ is connected (and thus, by Lemma 1, is a tree).*

Proof. Assume, without loss of generality, that no region in $FV(S)$ (and thus in $FV(S')$, for any $S' \subset S$) is empty. For $1 \leq i \leq n$, we let G_i denote the graph formed by the edges of $FV(\{s_1, \ldots, s_i\})$. If all n segments pass through a common point p then G_n is obviously connected through p. Otherwise, we prove connectedness of G_n by induction on n, as follows.

The base cases $G_1 = \emptyset$ and $G_2 = sep(s_1, s_2)$ are valid, so let us assume that G_{n-1} is connected, too. Let f be some face of $reg(s_n)$. We show that f has at least one vertex. This implies the lemma, because the parts of G_{n-1} that get disconnected when deleting edges in the interior of f will be re-connected in G_n with edges of f.

Face f having a boundary part but no vertex in common with some region $reg(s_i)$ implies that the entire bisector $sep(s_n, s_i)$ must belong to the boundary of f. This is equivalent to claiming that every closed disk D centered on $sep(s_n, s_i)$ and touching s_n and s_i must intersect the interiors of all the other segments. We disprove this claim by considering some segment s_k, $k \neq n, i$, that does not pass through the intersection point (if any) of s_n and s_i. By the assumptions made above, such a segment exists and we have $reg(s_k) \neq \emptyset$. So there exists some closed disk D that touches s_k at a point p and intersects all the other segments. Shrink D by moving its center directly towards p until D touches at least one of s_n and s_i, say s_n, at a single point q. This will happen because not both s_n and s_i can pass through p. If D still intersects the interior of s_i, then continue shrinking D by moving its center directly towards q, until this intersection is lost. Beside s_i, disk D now still touches s_n (at q), so its center is on $sep(s_n, s_i)$. However, D avoids the interior of s_k (and possibly other segments as well). This completes the proof. $\square$

By the arguments in the proof above, no intersection point between segments arises as a vertex of $FV(S)$, unless all segments pass through a common point. Observe also that the unbounded edges of $FV(S)$ (the leaves of the tree G_n) become straight rays rather than parabolic curves at places sufficiently remote from the convex hull of S, where they are defined by perpendicular bisectors of segment endpoints.

Finally, let us make an observation on the following (simpler) variant of $FV(S)$. When the distance to each segment $s_i \in S$ is measured to the farthest (instead of the closest) point on s_i, then this distance is always realized at one of the two endpoints of s_i. The resulting farthest-segment Voronoi diagram is identical to the well-investigated farthest-point Voronoi diagram of the segment endpoints. Each segment owns at most two, possibly non-adjacent, faces of the diagram.

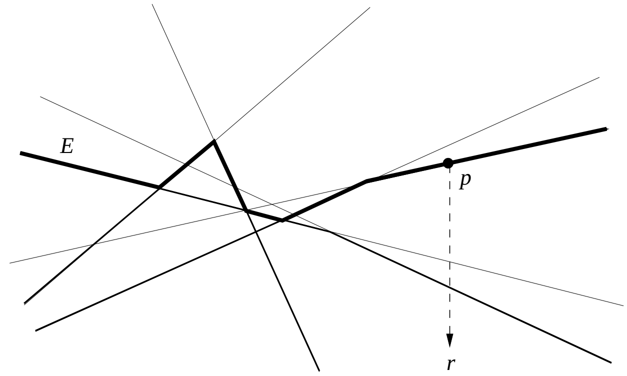

Figure 3: Dual wedges and their union

3 Dual setting

It is easier to study certain combinatorial and algorithmic properties of $FV(S)$ in a dual setting. Without loss of generality, let no line segment in S be vertical. We apply a standard point-line duality, T, which transforms a point $p = \binom{a}{b} \in \mathbb{R}^2$ into the (non-vertical) line $T(p) : y = ax - b$, and vice versa.

For our purposes, we send a segment $s_i = \overline{uv}$ into the wedge W_i that lies below both lines $T(u)$ and $T(v)$. A non-vertical halfplane H, bounded from below by the line g, is sent into the vertical ray $r(H)$ that emanates from $T(g)$ and is directed to $-\infty$. Simple analytic calculations show that $s_i \subset H$ is equivalent to $W_i \supset r(H)$.

Define E to be the boundary of the union of the wedges $W_1, \ldots, W_n$ (consult Figure 3). Consider some point $p \in E$ such that there is a unique wedge W_i whose boundary contains p. Then the vertical ray r below p transforms to an open halfplane H which intersects all segments in S but s_i. By Corollary 2, this means that $reg(s_i)$ is unbounded in the direction with slope equal to the x-coordinate of r. We conclude that the edges of E (in x-order) correspond to the faces of $FV(S)$ (in cyclic order) which are unbounded in directions 0 to π.

As a consequence, the number of faces of $FV(S)$ is at most two times the maximal number of edges that can appear on E. The latter number, in turn, can be bounded by $4n+2$. By construction, each wedge W_i contains the vertical ray below its apex, which enables us to apply the result in [8] on the complexity of the union of such wedges. This finally provides a proof for Theorem 3, because $FV(S)$ can be interpreted as a planar graph with $O(n)$ faces and vertex degree at least three.

Though clearly being asymptotically optimal, the obtained upper bound $8n + 4$ on the number of faces of $FV(S)$ is not considered the tightest possible. A lower bound of $4n-4$ can be shown by an example. Namely, for $1 \leq i \leq n$, let the segment s_i have endpoints p_i and $-p_i$, where $p_i = \binom{i^2}{i^3}$. Using a similar rotation argument as in Section 2, we get that each region $reg(s_i)$, for $i \neq 1$, consists of exactly two faces, whereas the number of faces of $reg(s_1)$ is $2n-2$. These numbers do not change when the segments are moved slightly so that no three of them pass through a common point.

Let us remark that the boundaries of the wedges W_i are x-monotone unbounded Jordan curves which pairwise cross at most three times. Thus, from the theory of Davenport-Schinzel sequences [18], we get the (weaker) bound $\lambda_3(n) = O(n \log n)$ for the complexity of E, the upper envelope of these curves.

4 Construction algorithm

Our algorithm for computing $FV(S)$ proceeds in two steps: Finding all unbounded edges of $FV(S)$ in cyclical order and, starting from there, intersecting bisectors in an appropriate order to construct the diagram edge by edge.

By the results in Section 3, the former step is equivalent to constructing the boundary of the union of the dual wedges $W_1, \ldots, W_n$. (To be precise, this task has to be carried out twice, the second time after having rotated the input set S by an angle of π.) Standard techniques, namely, divide-and-conquer paired with a plane-sweep method for merging the boundaries of the union of two subsets of wedges, are applied. In fact, after having transformed S into $W_1, \ldots, W_n$ in time $O(n)$, this step basically mimics Mergesort for the x-values of the boundary vertices. In the merge phase, we may have to construct new vertices and delete old ones, on account of their y-values checked during the plane sweep. The running time is $O(n \log n)$.

To get the intuition for step two, let us resort to a three-dimensional interpretation of $FV(S)$. For each index i, $1 \leq i \leq n$, we view the distance $d(x, s_i)$ as a (convex) function $z = f_i(x)$ over $\mathbb{R}^2$. Then $FV(S)$, being defined by distances to farthest segments, corresponds to the pointwise maximum of the functions $f_i, \ldots, f_n$, that is, to the upper envelope of their graphs in $\mathbb{R}^3$. This envelope, in turn, is the graph of a convex function again. Therefore, a sweep across $\mathbb{R}^3$ with a horizontal plane starting at $z = \infty$ will correctly report the individual components of this envelope in descending z-order. (It should be observed that this technique is not applicable for computing the *closest*-segment diagram, due to lack of convexity for the corresponding *lower* envelope.) Viewed in the two-dimensional picture, we start with the unbounded edges of $FV(S)$ (which are available from step one) and compute the edges and vertices of $FV(S)$ in order of decreasing distance from their farthest segment(s) in S.

The resulting algorithm is easy enough to be described in some detail. We maintain a cyclically ordered list C of all detected though uncompleted edges. In addition, pairs (e, e') of edges adjacent in C are held in a priority queue Q, which is organized by decreasing distance $\Delta(e, e')$ defined as follows. Let $e \subset sep(s_i, s_j)$ and $e' \subset sep(s_j, s_k)$. If $i \neq k$ then $\Delta(e, e')$ is the distance of v to the farthest segment in S, where v is the first point of intersection of $sep(s_i, s_j)$ and $sep(s_j, s_k)$. If $i = k$ then $\Delta(e, e') = \infty$ (meaning highest priority). Observe that C and Q can be initialized in time $O(n)$.

The generic step processes the next pair (e, e') in Q and removes it from Q. If $\Delta(e, e') < \infty$ then the edges e and e', that end at the vertex v, are reported and removed from the list C, and a new edge being part of $sep(s_i, s_k)$ is inserted into C. If $\Delta(e, e') = \infty$ then we only report a single edge, determined by $sep(s_i, s_j)$, and no vertex, and remove e and e' from C. Finally, we update Q so as to reflect the (constantly many) changes of adjacency in C. These actions can be carried out in $O(\log n)$ time per pair, and lead to the construction of at least one edge of $FV(S)$.

The number of edges of $FV(S)$ is $O(n)$ by Theorem 3, and we can summarize as follows.

Theorem 5 *Let S be an arbitrary set of n line segments in the plane. The farthest-segment Voronoi diagram of S can be constructed in $O(n \log n)$ time and $O(n)$ space.*

The algorithm is asymptotically optimal; it covers the case where all segments in S are points, and where it thus implicitly constructs the convex hull of n points in the plane. Because it uses only basic data structures and techniques, the algorithm is a candidate for practical implementation. It outperforms alternative approaches like divide-and-conquer or incremental insertion not only by its simplicity but also because—apart from the starting phase—only parts of the diagram are computed which do not have to be deleted later.

The question of whether the farthest-segment Voronoi diagram of a convex polygon can be computed faster is left open. We conjecture that an $O(n)$ algorithm is possible, as this is true for the closest-segment case, the medial axis of a convex polygon [1].

References

[1] A. Aggarwal, L.J. Guibas, J. Saxe, P.W. Shor. *A linear time algorithm for computing the Voronoi diagram of a convex polygon*. Discrete & Computational Geometry 4 (1989), 591-604.

[2] F. Aurenhammer. *Voronoi diagrams—a survey of a fundamental geometric data structure*. ACM Computing Surveys 23 (1991), 345-405.

[3] F. Aurenhammer, H. Edelsbrunner. *An optimal algorithm for constructing the weighted Voronoi diagram in the plane*. Pattern Recognition 17 (1984), 251-257.

[4] F. Aurenhammer, R. Klein. *Voronoi diagrams*. Handbook of Computational Geometry, Chapter 5, J.Sack,

G.Urrutia (eds.), Elsevier Science Publishing, 2000, 201-290.

[5] G. Barequet, M.T. Dickerson, R.L.S Drysdale. *2-Point site Voronoi diagrams*. Discrete Applied Mathematics 122 (2002), 37-54.

[6] O. Daescu, J. Luo, D. Mount. *Proximity problems on line segments spanned by points*. Computational Geometry: Theory and Applications 33 (2006), 115-129.

[7] R.L.S Drysdale, A. Mukhopadhyay. *An $O(n \log n)$ algorithm for the all-farthest-segments problem for a planar set of points*. Proc. 18^{th} Canadian Conf. Computational Geometry, to appear.

[8] H. Edelsbrunner, H.A. Maurer, F.P. Preparata, A.L. Rosenberg, E. Welzl, D. Wood. *Stabbing line segments*. BIT 22 (1982), 274-281.

[9] S. Fortune. *A sweepline algorithm for Voronoi diagrams*. Algorithmica 2 (1987), 153-174.

[10] C. Gold, P.R. Remmele, T. Roos. *Voronoi diagrams of line segments made easy*. Proc. 7^{th} Canadian Conf. Computational Geometry, 1995, 223-228.

[11] M.I. Karavelas. *A robust and efficient implementation for the segment Voronoi diagram*. Proc. 1^{st} Int. Symp. on Voronoi Diagrams in Science and Engineering, Tokyo, 2004, 51-62.

[12] D. Kirkpatrick. *Efficient computation of continuous skeletons*. Proc. 20^{th} Ann. IEEE Found. Comput. Sci., 1979, 18-27.

[13] D.T. Lee, V.B. Wu. *Multiplicative weighted farthest neighbor Voronoi diagrams in the plane*. Proc. Int. Workshop on Discrete Mathematics and Algorithms, Hong Kong, 1993, 154-168.

[14] D.T. Lee, R.L.S. Drysdale. *Generalization of Voronoi diagrams in the plane*. SIAM J. Comput. 10 (1981), 73-87.

[15] D.T. Lee, C.K. Wong. *Voronoi diagrams in the L_1 (L_∞) metrics with 2-dimensional storage applications*. SIAM J. Comput. 9 (1980), 200-211.

[16] A. Mukhopadhyay, S. Chatterjee, B. Lafreniere. *On the all-farthest-segments problem for a planar set of points*. Proc. 22^{nd} European Workshop on Computational Geometry, 2006, 71-74.

[17] A. Okabe, B. Boots, K. Sugihara, S.N. Chiu. Spatial Tesselations: Concepts and Applications of Voronoi Diagrams (2^{nd} ed.). Wiley Series in Probability and Statistics, 1999.

[18] M. Sharir, P.K. Agarwal. Davenport-Schinzel Sequences and Their Geometric Applications. Cambridge University Press, 1995.

[19] C.K. Yap. *An $O(n \log n)$ algorithm for the Voronoi diagram of a set of simple curve segments*. Discrete & Computational Geometry 2 (1987), 365-393.