On the Farthest Line-Segment Voronoi Diagram

Evanthia Papadopoulou* and Sandeep Kumar Dey

Faculty of Informatics, USI - Università della Svizzera Italiana, Lugano, Switzerland
{evanthia.papadopoulou,deys}@usi.ch

Abstract. The farthest line-segment Voronoi diagram shows properties surprisingly different from the farthest point Voronoi diagram: Voronoi regions may be disconnected and they are not characterized by convex-hull properties. In this paper we introduce the *farthest line-segment hull* and its *Gaussian map*, a closed polygonal curve that characterizes the regions of the farthest line-segment Voronoi diagram similarly to the way an ordinary convex hull characterizes the regions of the farthest-point Voronoi diagram. We also derive tighter bounds on the (linear) size of the farthest line-segment Voronoi diagram. With the purpose of unifying construction algorithms for farthest-point and farthest line-segment Voronoi diagrams, we adapt standard techniques for the construction of a convex hull to compute the farthest line-segment hull in $O(n \log n)$ or output-sensitive $O(n \log h)$ time, where n is the number of segments and h is the size of the hull (number of Voronoi faces). As a result, the farthest line-segment Voronoi diagram can be constructed in output sensitive $O(n \log h)$ time.

1 Introduction

Let S be a set of n simple geometric objects in the plane, such as points or line segments, called sites. The *farthest-site Voronoi diagram* of S is a subdivision of the plane into regions such that the region of a site s is the locus of points farther away from s than from any other site. Surprisingly, the farthest line-segment Voronoi diagram illustrates properties different from its counterpart for points [1]. For example, Voronoi regions are not characterized by convex-hull properties and they may be disconnected; a Voronoi region may consist of $\Theta(n)$ disconnected faces. Nevertheless, the graph structure of the diagram remains a tree and its structural complexity is $O(n)$. An abstract framework on the farthest-site Voronoi diagram (which does not include the case of intersecting line-segments) was given in [11]. Related is the farthest-polygon Voronoi diagram, later addressed in [4].

In this paper we further study the structural properties of the farthest line-segment Voronoi diagram. We introduce the *farthest line-segment hull* and its

* Research supported in part by the Swiss National Science Foundation, grant 200021-127137 and the ESF EUROCORES program EuroGIGA/VORONOI, SNF 20GG21-134355.

K.-M. Chao, T.-s. Hsu, and D.-T. Lee (Eds.): ISAAC 2012, LNCS 7676, pp. 187–196, 2012.

Gaussian map, a closed polygonal curve that characterizes the regions of the farthest line-segment Voronoi diagram similarly to the way an ordinary convex hull characterizes the regions of the farthest-point Voronoi diagram. Using the farthest line-segment hull we derive tighter upper and lower bounds on the (linear) structural complexity of the diagram improving the bounds in [1] by a constant factor. We provide $O(n \log n)$ and output sensitive $O(n \log h)$-time algorithms for the construction of the farthest line-segment hull, where h, $h \in O(n)$, is the size of the hull, by adapting standard approaches for the construction of an ordinary convex hull. Then the farthest line-segment Voronoi diagram can be constructed in additional $O(h \log h)$ time as given in [1] or in additional expected-$O(h)$ time by adapting the randomized incremental construction for points in [5]. The concept of the farthest hull is applicable to the entire L_p metric, $1 \leq p \leq \infty$, and it is identical for $1 < p < \infty$.

The farthest line-segment Voronoi diagram finds applications in computing the smallest disk that overlaps all given line-segments. It is necessary in defining and computing the *Hausdorff Voronoi diagram* of clusters of line segments, which finds applications in VLSI design automation, see e.g., [12] and references therein.

2 Definitions and the Farthest Hull

Let $S = \{s_1, \ldots, s_n\}$ be a set of n arbitrary line-segments in the plane. Line-segments may intersect or touch at single points. The distance between a point q and a line-segment s_i is $d(q, s_i) = \min\{d(q, y), \forall y \in s_i\}$, where $d(q, y)$ denotes the ordinary distance between two points q, y in the L_p metric $1 \leq p \leq \infty$. The farthest Voronoi region of a line-segment s_i is

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

The collection of all farthest Voronoi regions, together with their bounding edges and vertices, constitute the *farthest line-segment Voronoi diagram* of S, denoted as *FVD(S)* (see Fig. 1). Any maximally connected subset of a region in *FVD(S)* is called a *face*.

Any Voronoi edge bounding two neighboring regions, $freg(s_i)$ and $freg(s_j)$, is portion of the bisector $b(s_i, s_j)$, which is the locus of points equidistant from s_i and s_j. For line-segments in general position that are non-intersecting, $b(s_i, s_j)$ is an unbounded curve that consists of a constant number of simple pieces as induced by elementary bisectors between the endpoints and open portions of s_i, s_j. If segments intersect at a point p the bisector consists of two such curves intersecting at point p. If segments share a common endpoint the bisector may contain two dimensional regions in which case standard conventions get applied. Typically, a single line-segment is treated as three entities: two endpoints and an open line-segment; the entire equidistant area is assigned to the common endpoint. For more information on line-segment bisectors, see e.g., [9,10] for the L_2 metric, [14] for the L_∞ metric, and [8] for points in L_p.

The *farthest line-segment hull*, for brevity the *farthest hull*, is a closed polygonal curve that encodes the unbounded bisectors of *FVD(S)* maintaining their

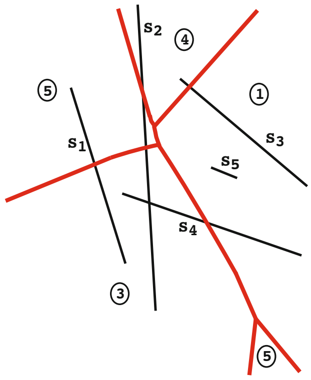

Fig. 1. A farthest line-segment Voronoi diagram in the Euclidean plane

cyclic order. In the following, we assume the ordinary Euclidean distance, however, definitions remain identical in the entire L_p, $1 < p < \infty$, metric. For the L_∞ (L_1) version of the farthest hull see [6].

Definition 1. *A line l through the endpoint p of a line-segment s, $s \in S$, is called a* supporting line *of S if and only if an open halfplane induced by l, denoted $H(l)$, intersects all segments in S, except s (and possibly except additional segments incident to p). Point p is said to* admit a supporting line *and it induces a vertex on the farthest hull. The unit normal of l pointing away from $H(l)$, is called the* unit vector *of l and is denoted $\nu(l)$. A line-segment s, $s \in S$, such that the line l through s is a supporting line of S and $H(l)$ intersects all segments in $S \setminus \{s\}$, is called a* hull segment*; the unit vector of l is also denoted $\nu(s)$.*

A single line-segment s may result in two hull segments of two opposite unit vectors if the supporting line through s intersects all segments in S.

Definition 2. *The line segment $\overline{pq}$ joining the endpoints p, q, of two line segments $s_i, s_j \in S$ is called a* supporting segment *if and only if an open halfplane induced by the line l through $\overline{pq}$, denoted $H(\overline{pq})$, intersects all segments in S, except s_i, s_j (and possibly except additional segments incident to p, q). The unit normal of $\overline{pq}$ pointing away from $H(\overline{pq})$ is called the* unit vector *of $\overline{pq}$, $\nu(\overline{pq})$.*

As shown in [1], a segment s has a non-empty Voronoi region in *FVD(S)* if and only if s or an endpoint of s admits a supporting line. The unbounded bisectors of *FVD(S)* correspond exactly to the supporting segments of S.

Theorem 1. *Let f-hull(S) denote the sequence of the hull segments and the supporting segments of S, ordered according to the angular order of their unit vectors. f-hull(S) forms a closed, possibly self intersecting, polygonal curve, which is called the* farthest line-segment hull *of S (for brevity, the* farthest hull*).*

Proof. (Sketch). The unit vector of any supporting line, hull segment, or supporting segment of S must be unique (by Definitions 1, 2). Thus, *f-hull(S)* admits a well defined ordering as obtained by the angular order of the unit vectors of its edges. Given a farthest hull edge e_i of unit vector $\nu_i = \nu(e_i)$, let $\nu_{i+1} = \nu(e_{i+1})$ be the unit vector following ν_i in a clockwise traversal of the circular list of all unit vectors of S. It is not hard to argue that e_{i+1} must be incident to an endpoint of e_i, m, by considering a supporting line through m, l_m, which starts as the line through e_i and rotates clockwise around m until it hits e_{i+1}. During the rotation l_m always remains a supporting line of S. Vertex m is chosen between the endpoints of e_i according to whether e_i is a hull or a supporting segment.

Thus, *f-hull(S)* forms a polygonal chain, which may self intersect and may visit a vertex multiple times. The uniqueness of unit vectors implies that edges are visited only once, hence the polygonal chain must be closed. □

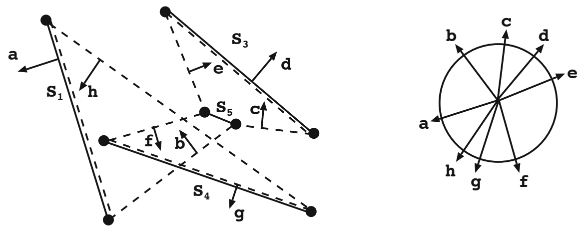

Fig. 2. The farthest line-segment hull for Fig. 1 and its Gaussian map

Fig. 2 illustrates the farthest hull and its angular ordering for the Voronoi diagram of Fig. 1. If line-segments in S degenerate to points, the farthest hull corresponds exactly to the convex hull of S. The vertices of the farthest hull are exactly the endpoints of S that admit a supporting line. The edges are of two types: supporting segments and hull segments. A supporting line of S is exactly a supporting line of the farthest hull. Any maximal chain of supporting segments between two consecutive hull segments must be convex.

Consider the *Gaussian map* of the farthest hull of S, for short *Gmap*, denoted *Gmap(S)*, which is a mapping of the farthest hull onto the unit circle K_o, such that every edge e is mapped to a point on the circumference of K_o as obtained by its unit vector $\nu(e)$, and every vertex is mapped to one or more arcs as delimited by the unit vectors of the incident edges (see Fig. 2). For more information on the Gaussian map see e.g., [3,13]. The Gaussian map can be viewed as a cyclic sequence of vertices of the farthest hull, each represented as an arc along the circumference of K_0. Each point along an arc of K_0 corresponds to the unit vector of a supporting line through the corresponding hull-vertex. The Gaussian map provides an encoding of all the supporting lines of the farthest hull as well as an encoding of the unbounded bisectors and the regions of the farthest line-segment Voronoi diagram. The portion of the Gaussian map above (resp. below) the horizontal diameter of K_o is referred to as the upper (resp. lower) Gmap. Thus, we can define the upper (resp. lower) farthest hull as the portion that corresponds to the upper (resp. lower) Gmap, similarly to an ordinary upper (resp. lower) convex hull.

Corollary 1. *FVD(S) has exactly one unbounded bisector for every supporting segment s of f-hull(S), which is unbounded in the direction opposite to* $\nu(s)$*. Unbounded bisectors in FVD(S) are cyclically ordered following exactly the cyclic ordering of Gmap(S).*

In [1] the unbounded bisectors of *FVD(S)* are identified through the point-line duality transformation T, which maps a point $p = (a, b)$ in the primal plane to a line $T(p) : y = ax - b$ in the dual plane, and vice versa. A segment $s_i = uv$ is sent into the wedge w_i that lies below (resp. above) both lines $T(u)$ and $T(v)$ (see Fig. 3 of [1]), referred to as the lower (resp. upper) wedge. Let E (resp. E') be the boundary of the union of the lower (resp. upper) wedges $w_1, \ldots, w_n$. As shown in [1], the edges of E (in x-order) correspond to the faces of *FVD(S)*, which are unbounded in directions 0 to π (in cyclic order). Respectively for the edges in E' and the Voronoi faces unbounded in directions π to 2π. In this paper we point out the equivalence between E and the lower Gmap. The edges of E in increasing x-order correspond exactly to the arcs of the lower Gmap in counterclockwise order; the vertices of E are exactly the unit vectors of the Gmap; the apexes of wedges in E are the unit vectors of hull segments. Respectively for E' and the upper Gmap. As pointed out in [1], E forms a Davenport-Schinzel sequence of order 3, thus the same holds for the lower Gmap. This observation does not imply a linear complexity bound, however. A $4n + 2$ upper bound on the size of E was shown in [1] based on [7]. For non-crossing segments, the order of the sequence is 2, which directly implies a linear complexity bound (see also [4]).

3 Improved Combinatorial Bounds

In this section we give tighter upper and lower bounds on the number of faces of the farthest line-segment Voronoi diagram for arbitrary line segments. Let a *start-vertex* and an *end-vertex* respectively stand for the right and the left endpoint of a line segment. Let an *interval* $[a_i, a_{i+1}]$ denote the portion of the lower Gmap between two consecutive (but not adjacent) occurrences of arcs for segment $s_a = (a', a)$, where a, a' denote the start-vertex and end-vertex of s_a respectively. Interval $[a_i, a_{i+1}]$ is assumed to be *non-trivial* i.e., it contains at least one arc in addition to a, a'. The following lemma is easy to derive using the duality transformation of [1].

Lemma 1. *Let* $[a_i, a_{i+1}]$ *be a non-trivial* interval *of segment* $s_a = (a', a)$ *on lower Gmap(S). We have the following properties: 1. The vertex following* a_i *(resp. preceding* a_{i+1}*) in* $[a_i, a_{i+1}]$ *must be a start-vertex (resp. end-vertex). 2. If* a_i *is a start-vertex (resp.* a_{i+1} *is end-vertex), no other start-vertex (resp. end-vertex) in the interval* $[a_i, a_{i+1}]$ *can appear before* a_i *or past* a_{i+1} *on the lower Gmap, and no end-vertex (resp. start-vertex) in* $[a_i, a_{i+1}]$ *can appear before* a_i *(resp. past* a_{i+1}*) on the lower Gmap.*

We use the following charging scheme for a non-trivial interval $[a_i, a_{i+1}]$: If a_i is a start-vertex, let u be the vertex immediately following a_i in $[a_i, a_{i+1}]$; the appearance of a_{i+1} is charged to u, which must be a start-vertex by Lemma 1. If a_{i+1} is an end-vertex, let u be the vertex in $[a_i, a_{i+1}]$ immediately preceding a_{i+1}; the appearance of a_i is charged to u, which by Lemma 1, must be an end-vertex.

Lemma 2. *The re-appearance along the lower Gmap of any endpoint of a segment s_a is charged to a unique vertex u of the lower f-hull, such that no other re-appearance of a segment endpoint on the lower Gmap can be charged to u.*

Proof. Let $[a_i, a_{i+1}]$ be a non-trivial interval of the lower Gmap and let u be the vertex charged the re-appearance of a_{i+1} (or a_i) as described in the above charging scheme. By Lemma 1, all occurrences of u are in $[a_i, a_{i+1}]$. Suppose (for a contradiction) that u can be charged the re-appearance of some other vertex c. Assuming that u is a start vertex, c must be a start vertex and an interval $[cu...c]$ must exist such that $cu \in [a_i, a_{i+1}]$, and thus $[cu...c] \in [a_i, a_{i+1}]$. But then u could not appear outside $[cu...c]$, contradicting the fact that u has been charged the reappearance of a_{i+1}. Similarly for an end vertex u. □

By Corollary 1, the number of faces of *FVD(S)* equals the number of supporting segments of the farthest hull, which equals the number of *maximal arcs* along the Gmap between consecutive pairs of unit vectors of supporting segments. There are two types of maximal arcs: *segment arcs*, which consist of a segment unit vector and its two incident arcs of the segment endpoints, and *single-vertex* arcs, which are single arcs bounded by the unit vectors of the two incident supporting segments.

Lemma 3. *The number of maximal arcs along the lower (resp. upper) Gmap, and thus the number of faces of FVD(S) unbounded in directions 0 to π (resp. π to 2π), is at most 3n-2. This bound is tight.*

Proof. Consider the sequence of all occurrences of a single segment $s_a = (a', a)$ on the lower Gmap. It is a sequence of the form

$$\ldots a \ldots aa' \ldots a' \ldots \quad \text{or} \quad \ldots a \ldots a \ldots a' \ldots a' \ldots$$

The number of maximal arcs involving s_a is exactly one plus the number of (non-trivial) intervals involving the endpoints of s_a. Summing over all segments, the total number of maximal arcs on the lower Gmap is at most n plus the total number of vertices that may get charged due to a non-trivial interval (i.e., the re-appearance of a vertex). By Lemma 2, a vertex can be charged at most once and there are 2n vertices in total. However, the first and last vertex along the lower Gmap cannot be charged at all. Thus, in total $3n - 2$. Similarly for the upper Gmap.

Figure 3a illustrates an example, in dual space, of n line segments (lower wedges) whose lower Gmap (boundary of the wedge union) consists of 3n-2 maximal arcs. There are exactly 2n-2 charges for vertex (wedge) re-appearances and exactly n hull segments. □

Theorem 2. *The total number of faces of the farthest line-segment Voronoi diagram of a set S of n arbitrary line segments is at most $6n-6$. A corresponding lower bound is 5n-4.*

Proof. The upper bound is derived by Lemma 3 and Corollary 1. A lower bound of 5n-4 faces can be derived by the example, in dual space, of Figure 3b. In Figure 3b, n line segments are depicted as lower and upper wedges using the point-line duality. There are $2n$ hull segments, $2(n-2)+1$ charges for vertex reappearances on lower wedges, and $n-1$ charges on upper wedges. □

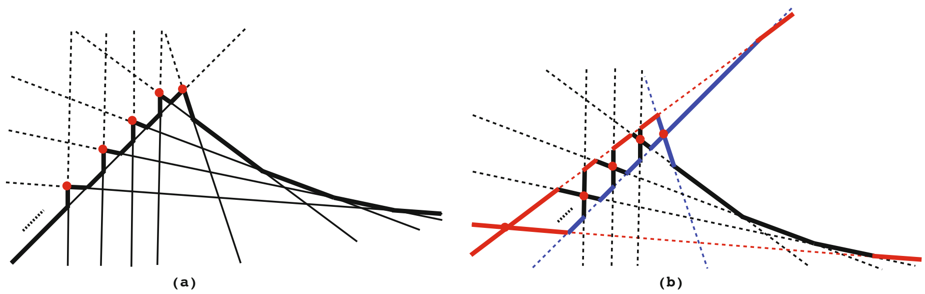

Fig. 3. (a)Lower Gmap of 3n-2 arcs. (b)Gmap of 5n-4 arcs.

Theorem 2 improves the $8n+4$ upper bound and the $4n-4$ lower bound on the number of faces of the farthest line-segment Voronoi diagram given in [1], as based on [7]. For disjoint segments the corresponding bound is $2n-2$ [4].

4 Algorithms for the Farthest Line-Segment Hull

Using the Gmap, we can adapt most standard techniques to compute a convex hull with the ability to compute the farthest line-segment hull, within the same time complexity. Our goal is to unify techniques for the construction of farthest-point and farthest line-segment Voronoi diagrams. By adapting Chan's output sensitive algorithm we derive an $O(n \log h)$ output-sensitive algorithm to compute the farthest line-segment Voronoi diagram. A two-phase $O(n \log n)$ algorithm in dual space, which is based on divide and conquer paired with plane sweep is outlined in [1].

4.1 Divide and Conquer or Incremental Constructions

We list properties that lead to a linear-time merging scheme for the farthest hulls of two disjoint sets of segments, L, R, and their Gaussian maps. The merging scheme leads to an $O(n \log n)$ divide-and-conquer approach and to an $O(n \log n)$ two-stage incremental construction. Recall that there may be $\Theta(|L|+|R|)$ supporting segments between the two farthest hulls. Given a unit vector $\nu(e)$ of

Gmap(L), let its R-vertex be the vertex v in R such that $\nu(e)$ falls along the arc of v in *Gmap(R)*. Respectively for *Gmap(R)*. A supporting line, an edge or a vertex of the farthest hull of L (resp. R) and their corresponding unit vector or arc are called *valid* if they remain in the farthest hull of $L \cup R$.

Lemma 4. *An edge e of f-hull(L) and its unit vector $\nu(e)$ remain valid if and only if the R-vertex of $\nu(e)$ in Gmap(R) lies in $H(e)$.*

Proof. Let q, in segment s_q, be the R-vertex of $\nu(e)$. The line parallel to e passing through q, $l_q(e)$, must be a supporting line of *f-hull(R)*. $H(e)$ must intersect all segments in L except those inducing e. If $q \in H(e)$ then $l_q(e) \in H(e)$ and thus $H(e)$ must intersect all segments in R, thus e must remain valid. If $q \notin H(e)$ then $l_q(e)$ lies entirely outside $H(e)$ and s_q does not intersect $H(e)$, thus e must be invalid. □

Lemma 5. *A vertex m of f-hull(L) remains valid if and only if either an incident farthest hull edge remains valid, or m is the L-vertex of an invalid unit vector in Gmap(R).*

Proof. A vertex incident to a valid farthest hull edge must clearly remain valid. By Lemma 5, if m is the L-vertex of an invalid edge in *f-hull(R)* then m must be valid. Conversely, suppose that m is valid but both edges e_i, e_{i+1} in *f-hull(L)* incident to m are invalid. Since m is valid, there is a supporting line of $L \cup R$ passing through m, denoted l_m, such that $\nu(l_m)$ is between $\nu(e_i)$ and $\nu(e_{i+1})$. Let u be the R-vertex of $\nu(l_m)$, where $\nu(l_m)$ is between $\nu(f_j)$ and $\nu(f_{j+1})$ in *Gmap(R)*. Since l_m is valid, u is in $H(l_m)$. Since $\nu(l_m)$ is between $\nu(f_j)$ and $\nu(f_{j+1})$, m cannot belong in $H(f_j) \cap H(f_{j+1})$. This implies that at least one of f_j and f_{j+1} must be invalid. □

Let *Gmap(LR)* denote the circular list of unit vectors and arcs derived by superimposing *Gmap(L)* and *Gmap(R)*. After determining valid and invalid unit vectors using Lemmas 4 and 5, *Gmap(LR)* represents a circular list of the vertices in *f-hull(L∪R)* (by Lemma 5). Thus, supporting segments between the two hulls can be easily derived by the pairs of consecutive valid vertices in *Gmap(LR)* that belong to different sets. In particular, to derive *Gmap(L∪R)* from *Gmap(LR)* do the following: For any two consecutive valid vectors, one in *Gmap*(L) and one in *Gmap*(R), insert a new unit vector (see Fig. 4). For any valid vertex m of *Gmap*(L) (resp. *Gmap*(R)) between two consecutive valid vectors of *Gmap*(R) (resp. *Gmap*(L)), insert new unit vectors for the corresponding supporting segments incident to m.

For $R = \{s\}$, the merging process can be refined as follows: Let $\nu_1(s)$ be the unit vector of s in its upper Gmap. 1. Perform binary search to locate $\nu_1(s)$ in *upper-Gmap(L)*. 2. Sequentially move counterclockwise along *upper-Gmap(L)* to test the validity of the encountered unit vectors until either a valid start-vertex of x-coordinate smaller than the start-vertex of s is found or the beginning of *upper-Gmap(L)* is reached. 3. Move clockwise along *upper-Gmap(L)* until either a valid end-vertex of x-coordinate larger than the end-vertex of s is found or the end of *upper-Gmap(L)* is reached. In the process, all relevant supporting

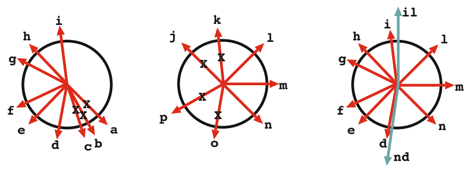

Fig. 4. Merging of two Gmaps

segments in the upper Gmap are identified. Similarly for the lower Gmap. This insertion procedure can give an alternative $O(n \log n)$ algorithm to compute the farthest hull by independently considering start-vertices and end-vertices in increasing x-coordinate followed by a merging step for the two Gmaps: First compute a partial Gmap of start vertices by inserting start-vertices in order of increasing x-coordinate. (Recall that insertion starts with a binary search). Then compute a partial Gmap of end-vertices by considering end-vertices in decreasing x-coordinate. Finally, merge the two partial Gmaps to obtain the complete Gmap. Because of the order of insertion, any portion of the partial Gmap traversed during an insertion phase gets deleted as invalid, thus the time complexity bound is achieved.

4.2 Output Sensitive Approaches

Using the Gmap, Jarvis march and quick hull are simple to generalize to construct the farthest hull within $O(nh)$ time, where h is the size of the farthest hull. For Jarvis march, unit vectors are identified one by one in say counterclockwise order starting at a vertex in some given direction, e.g., the bottommost horizontal supporting line. By combining an $O(n \log n)$ construction algorithm and Jarvis march as detailed in [2], we can obtain an $O(n \log h)$ output sensitive algorithm to construct the farthest hull. There is one point in [2] that needs modification to be applicable to the farthest hull, namely computing the tangent (here a supporting segment) between a given point and a convex hull (here a farthest hull, *f-hull(L)*). Note that unlike the ordinary convex hull, $\Theta(|L|)$ such supporting segments may exist, complicating a binary search. However, sequential search to compute those tangents can work within the same overall time complexity for one *wrapping phase* of the Jarvis march (see [2]). During a *wrapping phase*, a set T of at most hr tangents are computed, where r is the number of groups that partition the initial set of n points (here segments), each group being of size at most m, $m = \lceil n/r \rceil$. The set of tangents T can be computed in $O(n)$ time. In particular, given *Gmap*(S_i), $S_i \subseteq S$, a hull vertex p incident to segment s, and a supporting segment of unit vector ν, we can compute the next unit vector ν_{next} in *Gmap*$(S_i \cup \{s\})$ in counterclockwise order by sequentially scanning *Gmap*(S_i) starting at ν, applying the criteria of Lemmas 4 and 5 until ν_{next} is encountered. No portion of *Gmap*(S_i) between ν and ν_{next} needs to be encountered again during this wrapping phase. Thus, the $O(hr)$ supporting segments of one wrapping phase can be computed in $O(n)$ time. Note that if s is a hull segment, a supporting line through s may be included in T.

5 Concluding Remarks

Once the farthest hull is derived, the farthest line segment Voronoi diagram can be computed similarly to its counterpart for points. For example, one can use the simple $O(h \log h)$ algorithm of [1] or adapt the randomized incremental construction for the farthest-point Voronoi diagram of [5] in expected $O(h)$-time. The randomized analysis in [5] remains valid simply by substituting points along a convex hull with the elements of the farthest hull, where elements of the farthest hull are independent objects even when they refer to the same line segment. Adapting Chan's algorithm [2] for the construction of the farthest hull results in an output sensitive $O(n \log h)$ algorithm for the farthest line-segment Voronoi diagram.

References

1. Aurenhammer, F., Drysdale, R.L.S., Krasser, H.: Farthest line segment Voronoi diagrams. Information Processing Letters 100(6), 220–225 (2006)
2. Chan, T.M.: Optimal output-sensitive convex-hull algorithms in two and three dimensions. Discrete and Computational Geometry 16, 361–368 (1996)
3. Chen, L.L., Chou, S.Y., Woo, T.C.: Parting directions for mould and die design. Computer-Aided Design 25(12), 762–768 (1993)
4. Cheong, O., Everett, H., Glisse, M., Gudmundsson, J., Hornus, S., Lazard, S., Lee, M., Na, H.S.: Farthest-Polygon Voronoi Diagrams. arXiv:1001.3593v1 (cs.CG) (2010)
5. de Berg, M., Cheong, O., van Kreveld, M., Overmars, M.: Computational Geometry: Algorithms and Applications, 3rd edn. Springer (2008)
6. Dey, S.K., Papadopoulou, E.: The L_∞ farthest line segment Voronoi diagram. In: Proc. 9th Int. Symposium on Voronoi Diagrams in Science and Engineering (2012)
7. Edelsbrunner, H., Maurer, H.A., Preparata, F.P., Rosenberg, A.L., Welzl, E., Wood, D.: Stabbing Line Segments. BIT 22(3), 274–281 (1982)
8. Lee, D.T.: Two-dimensional Voronoi diagrams in the L_p metric. J. ACM 27(4), 604–618 (1980)
9. Lee, D.T., Drysdale, R.L.S.: Generalization of Voronoi Diagrams in the Plane. SIAM J. Comput. 10(1), 73–87 (1981)
10. Karavelas, M.I.: A robust and efficient implementation for the segment Voronoi diagram. In: Proc. 1st. Int. Symposium on Voronoi Diagrams in Science and Engineering, pp. 51–62 (2004)
11. Mehlhorn, K., Meiser, S., Rasch, R.: Furthest site abstract Voronoi diagrams. Int. J. of Comput. Geometry and Applications 11(6), 583–616 (2001)
12. Papadopoulou, E.: Net-aware critical area extraction for opens in VLSI circuits via higher-order Voronoi diagrams. IEEE Trans. on CAD 30(5), 704–716 (2011)
13. Papadopoulou, E., Lee, D.T.: The Hausdorff Voronoi diagram of polygonal objects: A divide and conquer approach. Int. J. of Computational Geometry and Applications 14(6), 421–452 (2004)
14. Papadopoulou, E., Lee, D.T.: The L_∞ Voronoi Diagram of Segments and VLSI Applications. Int. J. Comp. Geom. and Applications 11(5), 503–528 (2001)