International Journal of Computational Geometry
& Applications
Vol. 23, No. 6 (2013) 443–459

DOI: 10.1142/S0218195913600121

World Scientific
www.worldscientific.com

ON THE FARTHEST LINE-SEGMENT VORONOI DIAGRAM*

EVANTHIA PAPADOPOULOU†
Faculty of Informatics, USI - Università della Svizzera italiana
Lugano, Switzerland
evanthia.papadopoulou@usi.ch

SANDEEP KUMAR DEY
Faculty of Informatics, USI - Università della Svizzera italiana
Lugano, Switzerland
deys@usi.ch

Received 25 July 2013
Revised 29 January 2014
Communicated by Kun-Mao Chao and Tsan-Sheng Hsu, Guest Editors

ABSTRACT

The farthest line-segment Voronoi diagram illustrates properties surprisingly different from its counterpart for points: Voronoi regions may be disconnected and they are not characterized by convex-hull properties. In this paper we introduce the *farthest hull* and its *Gaussian map* as a closed polygonal curve that characterizes the regions of the farthest line-segment Voronoi diagram, and derive tighter bounds on the (linear) size of this diagram. With the purpose of unifying construction algorithms for farthest-point and farthest line-segment Voronoi diagrams, we adapt standard techniques to construct a convex hull and compute the farthest hull in $O(n \log n)$ or output sensitive $O(n \log h)$ time, where n is the number of line-segments and h is the number of faces in the corresponding farthest Voronoi diagram. As a result, the farthest line-segment Voronoi diagram can be constructed in output sensitive $O(n \log h)$ time. Our algorithms are given in the Euclidean plane but they hold also in the general L_p metric, $1 \leq p \leq \infty$.

Keywords: Voronoi diagram; farthest site; line segment; Gaussian map; farthest hull; output-sensitive algorithm; L_p metric.

*A preliminary version appeared in *Proc.* 23*rd International Symposium of Algorithms and Computation*, ISAAC 2012. Preliminary results in Sec. 5 appeared in *Proc.* 9*th International Symposium on Voronoi Diagrams in Science and Engineering*, ISVD 2012.
†Research supported in part by the Swiss National Science Foundation, project 200021-127137 and the ESF EUROCORES program EuroGIGA/VORONOI, SNF 20GG21-134355.

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 different properties from its counterpart for points.[2] For example, Voronoi regions are not characterized by convex-hull properties and they may be disconnected; in particular, a single Voronoi region may consist of $\Theta(n)$ disjoint faces. Nevertheless, the graph structure of the diagram remains a tree, and its structural complexity is $O(n)$.[2] An abstract framework on the farthest-site Voronoi diagram, applicable to disjoint line-segments, was given by Mehlhorn *et al.*[14] Related is the farthest-polygon Voronoi diagram, later addressed by Cheong *et al.*[6] For information on Voronoi diagrams see the book of Aurenhammer *et al.*[3]

In this paper we further study the structural properties of the farthest line-segment Voronoi diagram. We introduce the *farthest line-segment hull* (for brevity, *farthest hull* or simply *hull*), and its *Gaussian map*, a closed polygonal curve that characterizes the faces of the farthest line-segment Voronoi diagram similarly to the way an ordinary convex hull characterizes the regions of the farthest-point Voronoi diagram. The hull, through its Gaussian map, encodes the unbounded bisectors of the diagram while it maintains their cyclic order. Using it, we derive tighter upper and lower bounds on the (linear) structural complexity of the diagram improving the existing bounds of Aurenhammer *et al.*[2] by a constant factor. In particular, we show an upper bound of $6n - 6$ on the number of faces of the farthest line segment Voronoi diagram and a lower bound of $5n - 6$. We provide $O(n \log n)$-time and output sensitive $O(n \log h)$-time algorithms for the construction of the farthest line-segment hull, where h is the hull complexity ($h \leq 6n - 6$) by adapting standard techniques for the construction of a convex hull. Then, the farthest line-segment Voronoi diagram can be constructed similarly to its counterpart for points in additional $O(h \log h)$ time following the algorithm of Aurenhammer *et al.*[2] The farthest line-segment hull remains identical in the L_p metric for $1 < p < \infty$, and extends naturally to $p = 1, \infty$.

The farthest line-segment Voronoi diagram finds applications in computing the smallest disk that overlaps all line-segments in a given set. For example, line-segments may represent wires in a VLSI layer or in a different type of network, while disks represent defects. In the VLSI *critical area extraction* problem,[15] random manufacturing defects cause *open faults* when they overlap entire sets of wires or contacts on a layer under consideration. Thus, the farthest line-segment Voronoi diagram can be used for computing the *Probability of Fail* of a VLSI layer in the presence of random manufacturing defects. It is also necessary in defining and computing the *Hausdorff Voronoi diagram* of clusters of line segments, which finds similar applications in VLSI design automation, see e.g., Refs. 15 and 17, and references therein.

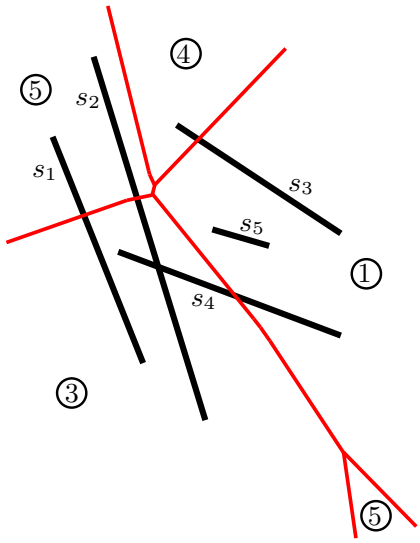

Fig. 1. A farthest line-segment Voronoi diagram in the Euclidean plane.

This paper is organized as follows. In Sec. 2, we give definitions and properties of the farthest hull and its Gaussian map. In Sec. 3, we provide improved upper and lower bounds on the structural complexity of the farthest line-segment Voronoi diagram. In Sec. 4, we adapt standard convex-hull techniques and describe worst case-optimal and output-sensitive algorithms for constructing the farthest hull. In Sec. 5, we discuss the farthest hull in the L_p metric, $1 \leq p \leq \infty$.

2. Preliminaries, Properties, and Definitions

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 a single point. 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

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

The collection of all farthest Voronoi regions of segments in S, together with their bounding edges and vertices, give 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*.

A Voronoi edge bounding two neighboring regions, $\mathit{freg}(s_i)$ and $\mathit{freg}(s_j)$, is portion of the bisector $b(s_i, s_j)$, which is the locus of points equidistant from s_i and s_j. For non-crossing line-segments in general position, $b(s_i, s_j)$ is an unbounded curve that consists of a constant number of pieces, where each piece is portion of an elementary bisector between the endpoints and open portions of s_i, s_j. In L_2, individual pieces of this curve are line-segments, rays, and parabolic arcs. If two

segments intersect at a point p, their bisector consists of two such curves intersecting at p. If the segments share a common endpoint, the bisector may contain two dimensional regions, in which case, standard conventions are applied to simplify the bisector to a curve. To avoid two-dimensional bisectors in the L_p metric, for $1 < p < \infty$, a single line-segment is typically treated as three entities, two endpoints and an open line-segment; the entire two-dimensional region of the bisector is assigned to the common endpoint.[9,11–13] In the following, we assume the ordinary Euclidean distance. However, all definitions remain identical in the general L_p metric, $1 < p < \infty$. The L_∞ (L_1) metric is considered in Sec. 5.

2.1. *Defining the farthest-hull and its Gaussian map*

The *farthest hull* is a closed polygonal curve that encodes the faces and the unbounded bisectors of $FVD(S)$ while it maintains their cyclic order. The cyclic order of the farthest hull is reflected by its *Gaussian map*. These concepts are defined in the following.

Definition 1. A line ℓ through the endpoint p of a line-segment $s \in S$ is called a *supporting line* of S if an open halfplane induced by ℓ, denoted $H(\ell)$, 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 defines a vertex on the farthest hull. The unit normal of ℓ pointing towards $H(\ell)$, is called the *unit vector* of ℓ and is denoted by $\nu(\ell)$. If the line ℓ through s is supporting, i.e., $H(\ell)$ intersects all segments in $S \setminus \{s\}$, line-segment s is said to *admit a supporting line* and it defines a segment on the farthest hull of unit vector $\nu(s) = \nu(\ell)$, called a *hull segment*.

A single line-segment s may result in two hull segments of two opposite unit vectors. This is the case when the 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 the open halfplane induced by the line ℓ through $\overline{pq}$, at opposite side of ℓ than s_i, s_j, denoted by $H(\overline{pq})$, 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 towards $H(\overline{pq})$ is called the *unit vector* of $\overline{pq}$, $\nu(\overline{pq})$.

Aurenhammer *et al.*[2] showed that a segment s_i has a non-empty farthest Voronoi region (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. Thus, $freg(s_j) \neq \emptyset$ if and only if s_i, or an endpoint of s_i, admits a supporting line. The unbounded bisectors of $FVD(S)$ correspond exactly to the supporting segments of S. The angular ordering of the unit vectors of S implies an ordering in the set of supporting and hull segments of S that corresponds exactly to the cyclic ordering of the faces of $FVD(S)$ at infinity.

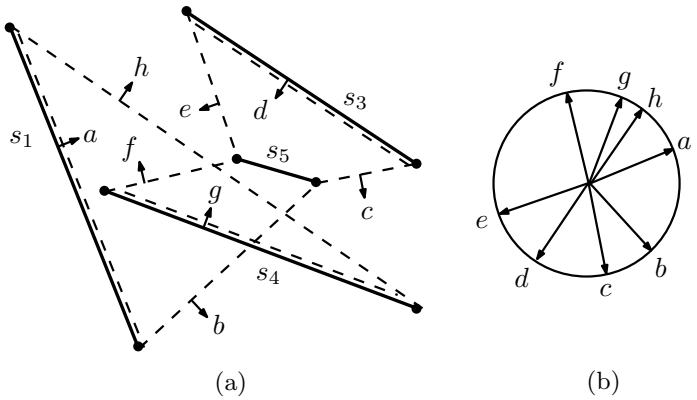

Fig. 2. The farthest line-segment hull of segments in Fig. 1 and its Gaussian map.

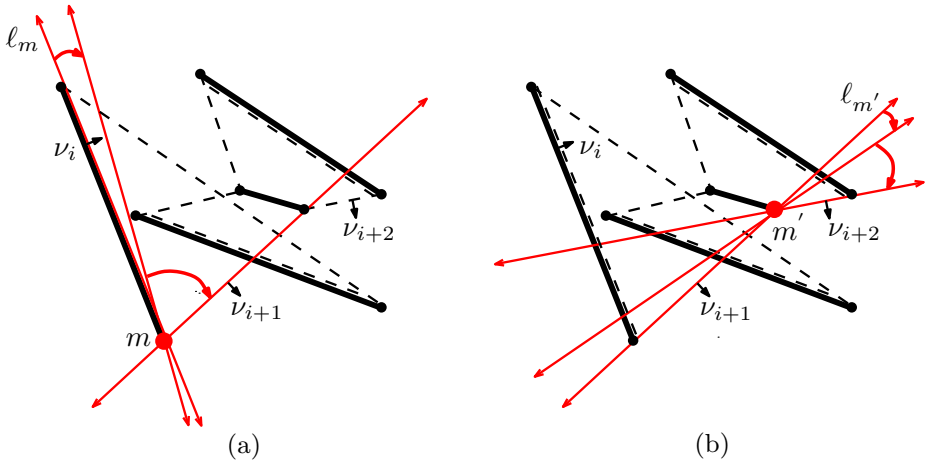

Fig. 3. Proof of Theorem 1.

Theorem 1. *Let S be a set of line-segments in the plane. The sequence of the hull segments and the supporting segments of S, ordered according to the angular order of their unit vectors, forms a closed, possibly self-intersecting, polygonal curve.*

We call the polygonal curve, subject of Theorem 1, the *farthest line-segment hull* of S (for brevity, the *farthest hull*), and denote it by f-hull(S). Figure 2(a) illustrates the farthest hull for the Voronoi diagram of Fig. 1. Its ordering is revealed by the angular order of unit vectors in Fig. 2(b). Note that segment s_2 from Fig. 1 is missing in the hull because its Voronoi region is empty.

Proof. Consider the circular list C of the unit vectors of S. By Defs. 1 and 2, unit vectors are unique, i.e., no two different supporting lines or segments can have the same unit vector. Let $\nu_i = \nu(e_i)$, $\nu_{i+1} = \nu(e_{i+1})$, $\nu_{i+2} = \nu(e_{i+2})$ be three consecutive unit vectors in C, in say clockwise order, where e_i, e_{i+1}, e_{i+2} are their corresponding supporting segments or hull segments (see Fig. 3). Consider the line through e_i, l_m,

where m is the endpoint of e_i such that if we rotate ℓ_m infinitesimally clockwise around m, ℓ_m will become a supporting line of S through m (see Fig. 3 (a)). Vertex m is chosen according to whether e_i is a hull segment or a supporting segment. As we rotate ℓ_m clockwise around m it must remain a supporting line of S, leaving the segment s_m incident to m lying entirely in the complement halfplane of $H(\ell_m)$, until $\nu(\ell_m) = \nu_{i+1}$. At that time, since unit vectors are unique, ℓ_m must be the line through e_{i+1}, and thus, e_{i+1} must be incident to m (by Defs. 1 and 2). However, if we continue to rotate ℓ_m clockwise past ν_{i+1}, ℓ_m stops being a supporting line. Thus, e_{i+2} cannot be incident to m. In fact, e_{i+2} must be incident to the other endpoint of e_{i+1}, which is denoted by m' in Fig. 3(b). The rotation continues around m'. By this argument, any two consecutive edges on f-hull(S) must be incident to the same vertex while no three consecutive ones can. Thus, f-hull(S) must form a closed polygonal curve. The vertices of the polygonal curve may repeat, that is, multiple vertices of f-hull(S) may correspond to the same point in S. Two different hull edges may correspond to the same segment. □

The vertices of the farthest hull are the endpoints of S that admit a supporting line. Note that one such endpoint may induce several hull vertices. The edges are of two types: supporting segments and hull segments. Two different hull segments may correspond to the same segment in S. Any maximal chain of supporting segments between two consecutive hull segments must be convex. If the line-segments in S degenerate to points, the farthest hull corresponds exactly to the convex hull of S.

Consider the Gaussian map, for short Gmap, of f-hull(S) onto the unit circle K_o, denoted as Gmap(S). Every hull edge e is mapped to a point on 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 its incident edges (see Fig. 2). The Gaussian map can be viewed as a cyclic sequence of vertices of the farthest hull, where each vertex is represented as an arc of the unit circle. Each point along an arc of the Gmap reveals the unit vector of a supporting line. The Gaussian map provides an encoding of all the supporting lines of the farthest hull. It also provides an encoding of the unbounded bisectors, the faces of the farthest line-segment Voronoi diagram, and the directions along which every face of the diagram is unbounded. It can be readily obtained by the circular list of the unit vectors of S. For previous uses of the Gaussian map, see, e.g., Refs. 5 and 16.

Corollary 1. *FVD(S) has exactly one unbounded Voronoi edge (bisector) for every supporting segment of f-hull(S), which is unbounded in the direction of its unit vector. Unbounded bisectors in FVD(S) are cyclically ordered following exactly the cyclic ordering of Gmap(S).*

Consider the maximal arcs along the Gmap between pairs of consecutive unit vectors of supporting segments. There are two types of maximal arcs: (1) *segment arcs*, which correspond to hull segments and consist of a segment unit vector and the two incident arcs of the segment endpoints; and (2) *single-vertex* arcs, which

are single arcs bounded by the unit vectors of incident supporting segments and correspond to single hull vertices. By Corollary 1, there is a 1-1 correspondence between the faces of $FVD(S)$ and maximal arcs of the Gmap.

Using the Gmap we can define the *upper* (resp., *lower*) farthest hull similarly to an ordinary upper (resp., lower) convex hull. In particular, let the portion of Gmap(S) above (resp., below) the horizontal diameter of the unit circle be referred to as the *upper* (resp., *lower*) Gmap. We can define the upper (resp., lower) farthest hull as the portion corresponding to the lower (resp., upper) Gmap, which is similar to an ordinary upper (resp., lower) convex hull. Note that in order to match the standard notion of an upper hull we map the lower Gmap to the upper farthest hull.

2.2. *Reviewing duality*

The standard point-line duality transformation T 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. Aurenhammer *et al.*[2] showed that it can be used to identify the unbounded bisectors of *FVD(S)*. A segment $s_i = uv$ is sent into the wedges w_i and w'_i that lie below and above respectively both lines $T(u)$ and $T(v)$, referred to as the lower and upper wedge respectively. Note that wedges w_i and w'_i must contain the vertical ray that emanates from their apex and is directed to $-\infty$ and to ∞ respectively. Define E to be the boundary of the union of the lower wedges $w_1, \ldots, w_n$ and E' to be the boundary of the union of the upper wedges $w'_1, \ldots, w'_n$. Then the faces of $FVD(S)$, which are unbounded in directions 0 to π in cyclic order, correspond to the edges of E in x-order.[2] Respectively for the edges of E' and the Voronoi faces unbounded in directions π to 2π.

The boundary E forms a Davenport-Schinzel sequence of order three;[2] this property, however, does not imply a linear complexity bound for E. Aurenhammer *et al.*[2] showed a $4n + 2$ upper bound on the complexity of E based on results of Edelsbrunner *et al.*,[10] which directly implies a linear $8n + 4$ upper bound on the number of faces of $FVD(S)$. For non-crossing line-segments, the order of the Davenport-Schinzel sequence is 2, which immediately implies a linear complexity bound.[6]

There is a clear equivalence between E and the upper Gmap. The edges of E in increasing x-order correspond exactly to the arcs of the upper 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 lower Gmap.

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. Let a *start-vertex* and an *end-vertex* stand

for the right and the left endpoint of a line-segment respectively. In terms of a lower wedge, a start-vertex is the left ray of the wedge and an end-vertex is the right ray.

Let an *interval* $[a_i, a_{i+1}]$ denote the portion of the upper 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 *nontrivial* i.e., it contains at least one arc in addition to a, a'. The following lemma is easy to derive using the duality transformation.

Lemma 1. *Let* $[a_i, a_{i+1}]$ *be a nontrivial* interval *of segment* $s_a = (a', a)$ *on upper 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., an 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 upper 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 upper Gmap.*

To count the number of vertex re-appearances along the upper Gmap we use the following charging scheme for a nontrivial 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 must be an end-vertex, by Lemma 1.

Lemma 2. *The re-appearance along the upper Gmap of an endpoint of a segment* s_a *is charged to a unique segment endpoint* u (*a start-vertex or an end-vertex of the upper Gmap*) *such that no other re-appearance of a segment endpoint on the upper Gmap can be charged to* u*.*

Proof. Let $[a_i, a_{i+1}]$ be a nontrivial interval of the upper Gmap and let u be the endpoint that has been 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 must be in $[a_i, a_{i+1}]$. Suppose (for the sake of contradiction) that u can be charged by the reappearance of some other vertex c. Assuming that u is a start-vertex, c must also be a start-vertex, and an interval $[cu...c]$ must exist such that $cu \in [a_i, a_{i+1}]$. 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 *maximal arcs* along the Gmap

Lemma 3. *The number of maximal arcs along the upper* (*resp., lower*) *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.*

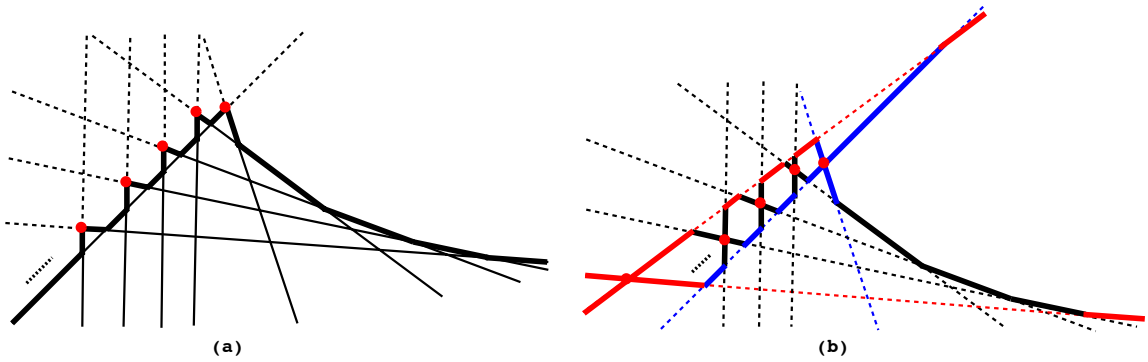

Fig. 4. (a) Upper Gmap of $3n-2$ arcs as an upper envelope of lower wedges. (b) Gmap of $5n-6$ arcs.

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

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

The number of maximal arcs involving s_a is exactly one plus the number of (nontrivial) intervals involving the endpoints of s_a. Summing over all segments, the total number of maximal arcs on the upper Gmap is at most n plus the total number of vertices that may get charged due to a nontrivial interval (i.e., the reappearance 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 vertices along the upper Gmap cannot be charged at all, thus, in total $3n-2$. Similarly for the lower Gmap.

Figure 4(a) illustrates an example, in dual space, of n line-segments (lower wedges) whose upper Gmap (boundary of the wedge union) consists of $3n-2$ maximal arcs. There are exactly n hull segments plus $2n-2$ charges for vertex (wedge) reappearances. □

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 $5n-6$.*

Proof. The upper bound is derived by Lemma 3 and Corollary 1. A lower bound of $5n-6$ faces can be derived by the example, in dual space, illustrated in Fig. 4(b), where a set of n line-segments is 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. Thus, a total of $5n-4$, minus the two common elements of the upper and lower Gmap. □

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 by Aurenhammer *et al.*,[2] based on results of Edelsbrunner *et al.*[10] For disjoint segments the corresponding bound is $2n-2$, which is tight.[6]

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 hull, within the same time complexity. For example, we can adapt Chan's output sensitive approach and compute the farthest hull in output sensitive $O(n \log h)$ time, where h is the size of the hull. As a result, the farthest line-segment Voronoi diagram can be computed in output sensitive $O(n \log h)$ time. Recall that h equals the number of faces on the farthest line-segment Voronoi diagram. In the following, we give algorithmic details for adapting standard convex hull techniques to compute the farthest hull. An alternative two-phase $O(n \log n)$ algorithm, not based on convex hull techniques, can be derived using the duality transformation.[2] Our goal is to unify techniques for the construction of the farthest-point and farthest line-segment Voronoi diagram.

4.1. *Divide and conquer or incremental constructions*

We list properties that lead to a linear-time merging scheme for the farthest hulls of any two sets of segments, L, R, $L \cap R = \emptyset$, and their Gaussian maps. The merging scheme yields an $O(n \log n)$ divide-and-conquer approach and an $O(n \log n)$ two-stage incremental construction. Recall that the farthest hulls of L and R may have $\Theta(|L| + |R|)$ supporting segments between them.

For a farthest-hull edge $e \in L$, let the *R-vertex* of e be the point v in R such that $\nu(e)$ falls along the arc of v in Gmap(R). Respectively for e in R. Let $I = L, R$. A supporting line, an edge or a vertex of f-hull(I), a unit vector or an arc of Gmap(I), are called *valid* if they remain in the farthest hull of $L \cup R$ or its Gaussian map.

Lemma 4. *An edge e in f-hull(L) of unit vector $\nu(e)$ is valid (i.e., it remains present in f-hull($L \cup R$)) if and only if the corresponding R-vertex q lies in the halfplane $H(e)$.*

Proof. Suppose e is valid. Then the line $\ell(q)$, parallel to e, passing through q, must be a supporting line of R. Consider $H(e)$. If $q \in H(e)$ then ℓ_q must be contained entirely in $H(e)$. Since $\ell(q)$ is supporting to R, $H(e)$ must intersect all segments in R. Since $H(e)$ must also intersect all segments in L, except those inducing e, e must be valid. If $q \notin H(e)$, then ℓ_q lies entirely outside $H(e)$, and thus, the segment incident to q cannot not intersect $H(e)$, that is, e must be invalid. □

Corollary 2. *If $\nu(e)$ in Gmap(L) is invalid, the corresponding R-vertex q must be valid.*

Proof. If e is invalid, then $q \notin H(e)$. Thus, $H(\ell_q)$, where ℓ_q is the line parallel to e passing through q, must intersect all segments in L and all segments in R except the one incident to q. Thus, q must be valid. □

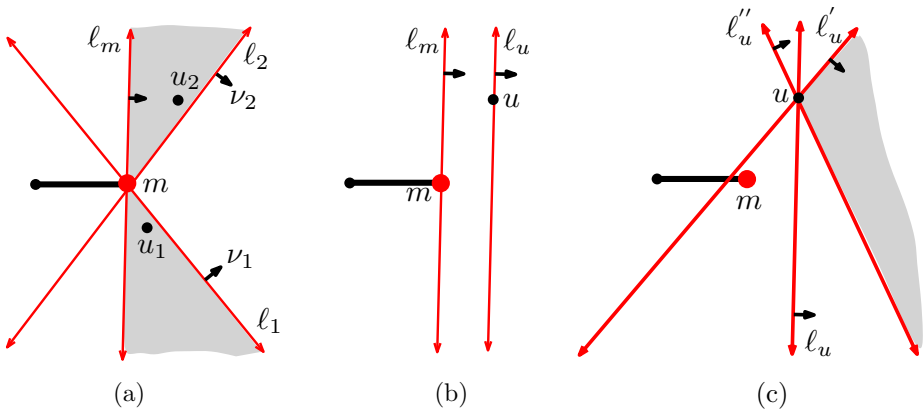

Fig. 5. Proof of Lemma 5.

Lemma 5. *An arc in Gmap(L) of hull vertex m in f-hull(L), is valid (i.e., it remains present in Gmap($L \cup R$)) if and only if either an incident unit vector 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 be valid. By Corollary 2, 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 incident unit vectors ν_1 and ν_2 in Gmap(L) are invalid. Since m is valid, there is a supporting line of $L \cup R$ passing through m, denoted ℓ_m, such that $\nu(\ell_m)$ is between ν_1 and ν_2 on the arc of m (see Fig. 5(a)). Let ℓ_1, ℓ_2 be the supporting lines of ν_1 and ν_2; let u_1, u_2 be the R-vertices of ν_1 and ν_2 respectively. Since ν_1, ν_2 are invalid, $u_1 \notin H(\ell_1)$ and $u_2 \notin H(\ell_2)$. Since m is valid, ℓ_m is a supporting line of $L \cup R$, thus, u_1 and u_2 must lie in $H(l_m) \cap H^c(\ell_1)$ and $H(l_m) \cap H^c(\ell_2)$ respectively, where $H^c(l)$ is the complement of $H(l)$. But these regions, which are shown shaded in Fig. 5(a), are disjoint, therefore, $u_1 \neq u_2$. Thus, there is a unit vector separating the arcs of u_1 and u_2 in Gmap(R), whose L-vertex is m.

Let u be the R-vertex of $\nu(\ell_m)$ in Gmap(R). Let ν'_1 and ν'_2 denote copies of the unit vectors ν_1 and ν_2 in Gmap(R). Let ν be a unit vector in R bounding the arc of u between ν'_1 and ν'_2, i.e., m is the L-vertex of ν. Such a unit vector must exist in Gmap(R) by the argument above. Since ℓ_m is valid, by Lemma 4, u must be in $H(\ell_m)$. Consider the line ℓ_u parallel to l_m passing through u (see Fig. 5(b)). It lies entirely in $H(\ell_m)$. If we rotate ℓ_u slightly clockwise and counterclockwise around u into lines ℓ'_u and ℓ''_u respectively, it is clear that m cannot be contained in both $H(\ell'_u)$ and $H(\ell''_u)$ (see Fig. 5(c), where $H(\ell'_u) \cap H(\ell''_u)$ is shown shaded). Thus, not both $\nu(\ell'_u)$ and $\nu(\ell''_u)$ can be valid. If we continue the rotation around u we will reach the unit vectors bounding the arc of u in Gmap(R), among which one may be ν'_1 or ν'_2. For the same reason, not both bounding vectors can be valid. Since both ν'_1 or ν'_2 are valid in Gmap(R), an invalid unit vector must exist between them. □

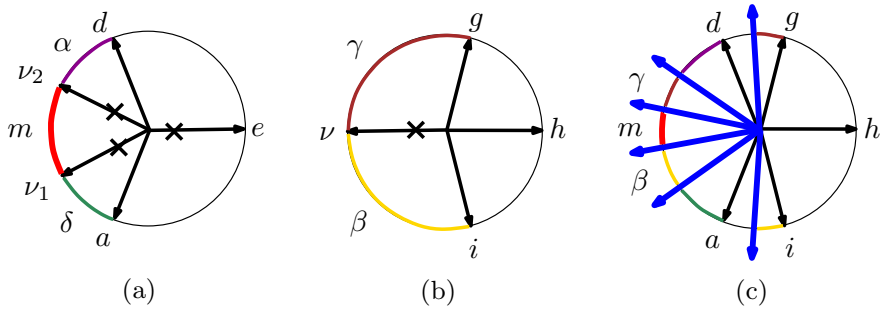

Fig. 6. (a) Gmap(L); (b) Gmap(R); (c) Gmap($L \cup R$), a merge vector is inserted between every two consecutive valid vertices of opposite sets.

Let Gmap(L)∪Gmap(R) denote the circular list of unit vectors and arcs derived by superimposing Gmap(L) and Gmap(R). We can determine valid and invalid unit vectors as well as valid and invalid hull vertices (arcs) using Lemmas 4 and 5. Then Gmap(L)∪Gmap(R) corresponds to the circular list of the vertices in f-hull($L \cup R$). Recall that by Lemma 5, any invalid unit vector represents a valid vertex in the opposite set. For any pair of consecutive vertices, one in L and one in R, a new unit vector, referred to as a *merge vector*, needs to be inserted corresponding to a new supporting segment between the two hulls joining these two vertices. After inserting the merge vectors into Gmap(L)∪Gmap(R), all invalid vectors can be removed, resulting into Gmap($L \cup R$). The merging process is clearly linear. It is illustrated in Fig. 6; invalid vectors are indicated by x's; merge vectors are indicated by longer arrows.

Lemma 6. *Gmap(L) and Gmap(R) can be merged into Gmap($L \cup R$) in linear time.*

The merging process implies a simple $O(n \log n)$-time divide-and-conquer algorithm to compute the farthest hull of S.

In an incremental process, merging is also performed, however, one of the sets is a single segment, say $R = \{s\}$. Then 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 the upper Gmap(L); (2) Move sequentially counterclockwise along the 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; and (3) Move clockwise along the 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 the upper Gmap(L) is reached. In the process, all relevant supporting 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 easy 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 bottom-most horizontal supporting line. By combining an $O(n \log n)$ construction algorithm and Jarvis march,[4] we can obtain an $O(n \log h)$ output sensitive algorithm to construct the farthest hull of line-segments. There is one point in Chan's approach[4] 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 these tangents can work within the same overall time complexity for one *wrapping phase* of the Jarvis march.

During a wrapping phase,[4] 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, 5 until ν_{next} is encountered. No portion of Gmap(S_i) between ν and ν_{next} is 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.

4.3. *Computing the farthest line segment Voronoi diagram*

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 standard $O(h \log h)$ algorithm of Aurenhammer *et al.*,[2] which starts with the circular list on unbounded bisectors and processes Voronoi vertices in order of decreasing *weight* (the radius of the implied circle). We conjecture that the diagram can also be computed by an expected-$O(h)$ randomized algorithm, after the farthest hull is available, as inspired by the corresponding algorithms for points on a convex hull.[7,8] Several new complications arise, however, due to the multiple appearances of sites

along a farthest hull, which go beyond the scope of this paper. Recall that a single segment may contribute $\Theta(n)$ distinct hull elements in the worst case.

5. The Farthest Hull in the L_p Metric

The concept of the farthest hull is valid in the general L_p metric, $1 \leq p \leq \infty$. Moreover, the Euclidean farthest hull remains identical for any p, $1 < p < \infty$, due to properties of L_p disks and L_p bisectors. An L_p, $1 < p < \infty$, disk of infinite radius is an ordinary halfplane,[11] thus, the condition for a segment to have a non-empty Voronoi region, and therefore Theorem 1, remain identical for $1 < p < \infty$.

In this section, we define the L_∞ farthest hull (see Fig. 7). The L_1 metric is equivalent to L_∞ under 45° rotation. A halfplane bounded by an axis-parallel line is called an *axis-parallel halfplane*. The common intersection of two axis parallel halfplanes with perpendicular bounding lines is called a *quadrant*. The following lemma is a simple adaptation from the Euclidean case.[2]

Lemma 7. *In L_∞, a segment s_i has a non-empty farthest Voronoi region if and only if there exists either an axis-parallel halfplane or a quadrant L which touches s_i and intersects (or touches) all other segments in S.*

By Lemma 7, all faces of $FVD(S)$ are unbounded in one of the eight possible directions that are implied by rays of slope $\pm 1, 0, \infty$. Figure 9 shows an example of an L_∞ farthest line segment Voronoi diagram.

Proof. Let t be a point in an arbitrary face f of $FVD(S)$ belonging to the region of segment s_i. Then there is a square $D(t)$ centered at t, touching s_i, that is intersecting (or touching) all segments $s_j \in S \setminus \{s_i\}$. If $D(t)$ touches s_i with a vertical (resp., horizontal) side let r be a horizontal (resp., vertical) ray starting at t directed away from s_i. If $D(t)$ touches s_i with a corner at a single point p, let r be the ± 1-slope ray through the incident diagonal of $D(t)$, starting at t and directed away from p. Since for any point y along r, $D(t) \subset D(y)$, and $D(y)$ touches s_i in exactly the same way as $D(t)$, $D(y)$ must continue to intersect all other segments in S and thus, r must be entirely contained in $freg(s_i)$ and in particular f. Thus, face f must be unbounded along the direction of r. □

Let $l_i, i = n, s, e, w$ be four axis parallel *bounding lines* of S, where l_n (resp., l_s) is the horizontal line passing through the bottom-most upper-endpoint (resp., the topmost lower-endpoint) of all segments in S, and l_e (resp., l_w) is the vertical line passing through the leftmost right-endpoint (resp., the rightmost left-endpoint) of all segments in S. The bounding lines partition the plane into four quadrants, labeled $1-4$ in counterclockwise order, where Quadrant 1 is formed by (l_s, l_w) and faces *north-east* (see Fig. 7). Quadrant 2 is formed by (l_s, l_e), Quadrant 3 by (l_n, l_e), and Quadrant 4 by (l_n, l_w). If a closed rectangular region R is induced by the four bounding lines, then no segment s that lies partially or entirely in R can

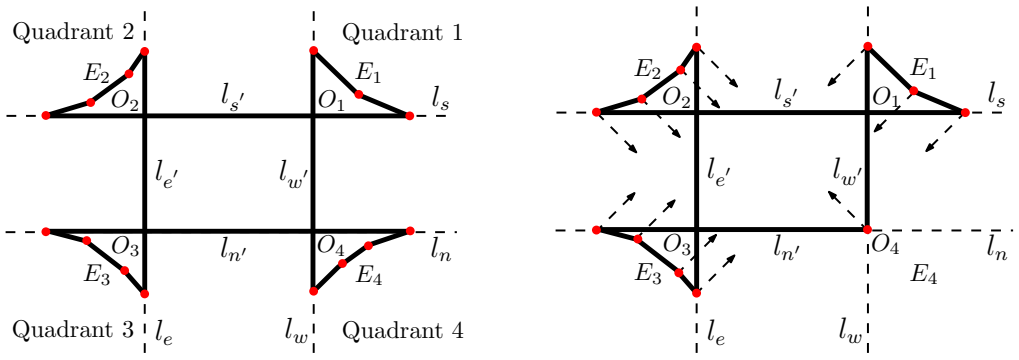

Fig. 7. The farthest-hull in the L_∞ metric.

have a non-empty Voronoi region in $FVD(S)$. Among the remaining segments, none may have endpoints in the four quadrants, thus, the quadrants are either empty of the remaining segments, or there are segments that straddle them entirely (see also Fig. 9). Let $E_i, i = 1, 2$ (resp., $i = 3, 4$) denote the upper (resp., lower) envelope of the set of line segments straddling Quadrant i. If no segments straddle Quadrant i, let $E_i = O_i$, where O_i is the corner point (origin) of Quadrant i (see Fig. 7).

Definition 3. The L_∞ *farthest hull* of S is the closed polygonal curve obtained by $E_1, l_{s'}, E_2, l_{e'}, E_3, l_{n'}, E_4, l_{w'}$, where E_i is the envelope of Quadrant i and $l_{j'}$ is the portion of the bounding line l_j between two incident envelopes.

The bounding lines need not always appear in the normal order depicted in Fig. 7, e.g., l_n may lie above (or on) l_s and l_e may lie to the right of l_w, however, the farthest hull definition remains identical in all cases (see e.g., Fig. 8). The L_∞ farthest line-segment Voronoi diagram has exactly one unbounded face for each edge of the farthest hull and its graph structure is a tree (similarly to L_2). It always contains four faces, facing north, south, east, and west, each induced by one bounding line, l_n, l_s, l_e, l_w respectively. For any nontrivial envelope, it contains one face for each segment of the envelope, which is bounded by parallel slope-± 1 rays. Each vertex v_i of the farthest hull corresponds to an unbounded bisector, which is a ray of slope ± 1 emanating from v_i. Figure 7(b) illustrates the unbounded bisectors of $FVD(S)$ as rays emanating from the vertices of the farthest hull.

Theorem 3. *The L_∞ farthest Voronoi diagram of n arbitrary line segments has at most $n + 8$ faces and this is tight. For non-crossing segments this number is eight.*

Proof. The total number of faces of the diagram is four (for the bounding lines) plus the total number of edges along the envelopes of the farthest hull. A segment may contribute an edge to at most two envelopes. However, only the first and the last edge of a pair of envelopes may be attributed to the same segment (by definition of an envelope). Thus, the total number of edges along the envelopes is at most $n+4$ and the total number of faces in the diagram is at most $n + 8$. This bound is tight

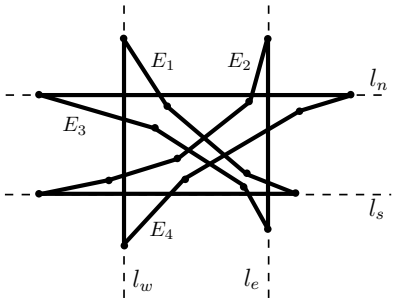

Fig. 8. A non-standard L_∞ farthest hull.

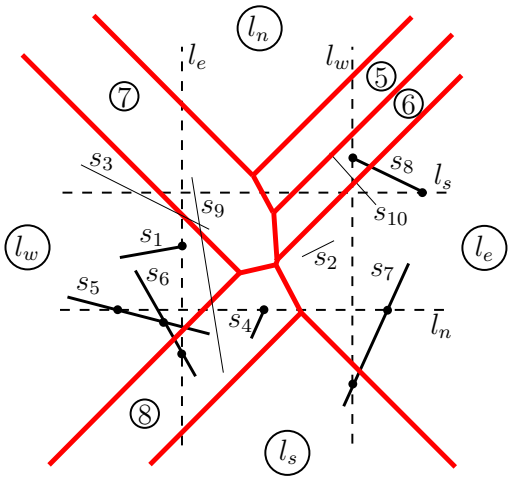

Fig. 9. Farthest Voronoi diagram of segments in L_∞.

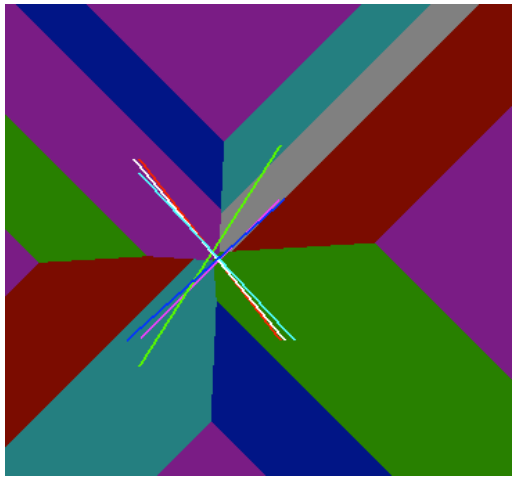

Fig. 10. Example of $n = 6$ segments producing $n + 8$ disjoint faces.

as shown in Fig. 10, where four line segments produce eight disjoint faces, each one contributing to exactly two non-adjacent envelope edges. In addition, the segment in purple induces all four bounding lines plus one envelope edge that is not incident to any bounding line, for a total of 5 faces. Any other segment can be placed to contribute exactly one face. □

Once the L_∞ farthest hull is available, the corresponding Voronoi diagram can be constructed in $O(h)$ time, where h is its complexity.

6. Concluding Remarks

The farthest Voronoi diagram of a set of n arbitrary line segments can be computed in output sensitive $O(n \log h)$ time, where $h \in O(n)$ is the number of its faces. This is achieved by first computing the farthest hull in $O(n \log h)$ time, and then computing the respective Voronoi diagram in additional $O(h \log h)$ time. We conjecture that the farthest line-segment Voronoi diagram can also be constructed in additional $O(h)$

deterministic or expected time (after computing the farthest hull) by following the respective approaches for computing the Voronoi diagram of an ordinary convex hull by Aggarwal *et al.*[1] or Chew,[7,8] while augmenting them with techniques to handle the multiplicity of sites along a farthest hull. The farthest hull is identical in the general L_p metric, $1 < p < \infty$, and extends naturally to $p = 1, \infty$.

References

1. A. Aggarwal, L. Guibas, J. Saxe and P. Shor, A Linear time algorithm for computing the Voronoi diagram of a convex polygon, *Discr. Comput. Geom.* **4**(1) (1989) 591–604.
2. F. Aurenhammer, R. L. S. Drysdale and H. Krasser, Farthest line-segment Voronoi diagrams, *Infor. Process. Lett.* **100**(6) (2006) 220–225.
3. F. Aurenhammer, R. Klein and D. T. Lee, *Voronoi Diagrams and Delaunay Triangulations* (World Scientific Publishing Company, Singapore, 2013).
4. T. M. Chan, Optimal output-sensitive convex-hull algorithms in two and three dimensions, *Discr. Comput. Geom.* **16** (1996) 361–368.
5. L. L. Chen, S. Y. Chou and T. C. Woo, Parting directions for mould and die design, *Comput.-Aided Design* **25**(12) (1993) 762–768.
6. O. Cheong, H. Everett, M. Glisse, J. Gudmundsson, S. Hornus, S. Lazard, M. Lee and H. S. Na, Farthest-polygon voronoi diagrams, *Comput. Geom.* **44**(4) (2011) 234–247.
7. L. P. Chew, Building Voronoi diagrams for convex polygons in linear expected time, Technical report, Dartmouth College, Hanover, USA (1990).
8. M. de Berg, O. Cheong, M. van Kreveld and M. Overmars, *Computational Geometry: Algorithms and Applications*, 3rd edn. (Springer-Verlag, 2008).
9. D. G. Kirkpatrick, Efficient computation of continuous skeletons, *Proc.* 20*th Annual Symp. Foundations of Computer Science* (1979), pp. 18–27.
10. H. Edelsbrunner, H. A. Maurer, F. P. Preparata, A. L. Rosenberg, E. Welzl and D. Wood, Stabbing line-segments, *BIT* **22**(3) (1982) 274–281.
11. D. T. Lee, Two-dimensional Voronoi diagrams in the L_p metric, *J. ACM* **27**(4) (1980) 604–618.
12. D. T. Lee and R. L. S. Drysdale, Generalization of Voronoi diagrams in the plane, *SIAM J. Comput.* **10**(1) (1981) 73–87.
13. M. I. Karavelas, A robust and efficient implementation for the segment Voronoi diagram, *Proc.* 1*st. Int. Symp. Voronoi Diagrams in Science and Engineering* (2004), pp. 51–62.
14. K. Mehlhorn, S. Meiser and R. Rasch, Furthest site abstract Voronoi diagrams, *Int. J. Comput. Geom. Appl.* **11**(6) (2001) 583–616.
15. E. Papadopoulou, Net-aware critical area extraction for opens in VLSI circuits via higher-order Voronoi diagrams, *IEEE Trans. Comput.-Aided Design* **30**(5) (2011) 704–716.
16. E. Papadopoulou and D. T. Lee, The Hausdorff Voronoi diagram of polygonal objects: A divide and conquer approach, *Int. J. Comput. Geom. Appl.* **14**(6) (2004) 421–452.
17. J. Xu, L. Xu and E. Papadopoulou, Computing the map of geometric minimal cuts, *Algorithmica*, published online, October 2012, DOI 10.1007/s00453-012-9704-9.