

## The Upper Envelope of Voronoi Surfaces and Its Applications\*

Daniel P. Huttenlocher,<sup>1</sup> Klara Kedem,<sup>1,2</sup> and Micha Sharir<sup>2,3</sup>

<sup>1</sup> Department of Computer Science, Cornell University, Ithaca, NY 14853, USA  
dph@cs.cornell.edu

<sup>2</sup> Department of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel  
klara@math.tau.ac.il  
sharir@math.tau.ac.il

<sup>3</sup> Courant Institute of Mathematical Sciences, New York University,  
New York, NY 10012, USA

**Abstract.** Given a set  $S$  of *sources* (points or segments) in  $\mathbb{R}^d$ , we consider the surface in  $\mathbb{R}^{d+1}$  that is the graph of the function  $d(x) = \min_{p \in S} \rho(x, p)$  for some metric  $\rho$ . This surface is closely related to the Voronoi diagram,  $\text{Vor}(S)$ , of  $S$  under the metric  $\rho$ . The upper envelope of a set of these *Voronoi surfaces*, each defined for a different set of sources, can be used to solve the problem of finding the minimum Hausdorff distance between two sets of points or line segments under translation. We derive bounds on the number of vertices on the upper envelope of a collection of Voronoi surfaces, and provide efficient algorithms to calculate these vertices. We then discuss applications of the methods to the problems of finding the minimum Hausdorff distance under translation, between sets of points and segments.

### 1. Introduction

Voronoi diagrams are a central construction in computational geometry and serve as an important tool for solving many practical problems in various fields [7],

---

\* The first author was supported in part by NSF PYI Grant IRI-9057928 and matching funds from General Electric, Kodak, and Xerox, and by U.S. Air Force Contract AFOSR-91-0328. The second and third authors were supported by the Fund for Basic Research administered by the Israeli Academy of Sciences. The second author was also supported by a fellowship from the Pikkowski-Valazzi Fund and by the Eshkol Grant 04601-90 from the Israeli Ministry of Science and Technology. The third author was also supported by Office of Naval Research Grant N00014-90-J-1284, by NSF Grant CCR-89-01484, by grants from the U.S.–Israeli Binational Science Foundation, from the Basic Research Fund administered by the Israeli Academy of Sciences, and from G.I.F., the German–Israeli Foundation for Scientific Research and Development.

[11], [20], [23], [25]. In this paper we investigate several problems in which a set of Voronoi diagrams in  $d$ -space are “superimposed” on one another. It is convenient to replace each such Voronoi diagram, induced by some set  $S$ , by a corresponding *Voronoi surface* in  $(d + 1)$ -space. The height of this surface at a point  $x$  is simply the distance from  $x$  to the nearest source of  $S$  (see [14]). We discuss the structure of the upper envelope of  $m$  such surfaces, provide bounds on the number of vertices on this upper envelope, and develop efficient algorithms for computing the vertices.

Given a set  $S = \{p_j | j = 1, \dots, n\}$  of sources (points or line segments) in  $\mathbb{R}^d$ , and some metric  $\rho(\cdot, \cdot)$ , the Voronoi diagram of  $S$ , denoted by  $\text{Vor}(S)$ , is the decomposition of  $\mathbb{R}^d$  into “Voronoi cells”  $C_1, \dots, C_n$ , where each cell  $C_j$  contains those points of  $\mathbb{R}^d$  that are closer to  $p_j$  than to any other source (with closeness measured using the metric  $\rho$ ). Consider now the function

$$d(x) = \min_{q \in S} \rho(x, q). \tag{1}$$

The graph of this function,  $\{(x, d(x)) | x \in \mathbb{R}^d\}$ , is a surface which we call the *Voronoi surface* of  $S$ . Throughout this paper we use a slight abuse of notation and refer to both the function and its graph using the notation  $d(x)$ . The surface  $d(x)$  is at a local minimum (of zero) exactly when  $x$  is coincident with some source  $p_j \in S$ , and is at a local maximum for certain points that lie along the boundary of cells of  $\text{Vor}(S)$ .

Let  $\{S_i | i = 1, \dots, m\}$  be  $m$  sets of sources, let  $n_i = |S_i|$ ,  $i = 1, \dots, m$ , be the number of sources in  $S_i$ , and let  $n = \sum_{i=1}^m n_i$ . Denote the Voronoi surface of the set  $S_i$  by  $d_i(x)$ . The upper envelope of these surfaces is (the graph of) the function

$$f(x) = \max_{i=1, \dots, m} d_i(x). \tag{2}$$

Thus  $f(x)$  gives the largest distance from  $x$  to its  $m$  nearest neighbors, one from each set  $S_i$ . We investigate the structure of this upper envelope. Again, we use  $f(x)$  to refer to both the function and its graph. We show that all the local minima of  $f(x)$  are achieved at  $k$ -faces of the upper envelope ( $k \leq d - 1$ ), and that the number of local minima is proportional to the number of *vertices* of the upper envelope (where a vertex is a point at which  $d + 1$  Voronoi surfaces meet). We bound the number of such vertices and provide efficient algorithms for computing them, when  $d = 2, 3$ . These algorithms also yield all local minima on the upper envelope, thereby providing the global minimum of  $f(x)$ , which is needed to solve the applications mentioned in the abstract and discussed in this paper.

Our main results on the upper envelope of Voronoi surfaces are as follows. Consider  $m$  sets of sources,  $S_1, \dots, S_m$ , with  $n_i = |S_i|$  and  $n = \sum_{i=1}^m n_i$ , and let the distance function  $\rho(\cdot, \cdot)$  be the  $L_2$  norm.

- If the sources are all points in  $\mathbb{R}^2$ , then the maximum number of vertices,  $V$ , of  $f(x)$  is  $O(mn\alpha(mn))$  (where  $\alpha(n)$  is the inverse Ackermann function).
- If the sources are all points in  $\mathbb{R}^3$ , then  $V = O(mn^2\alpha(mn))$ . This result is based on a new bound on the complexity of the union of convex polyhedra in 3-space which share a common point (see the final item below).

- If the sources are points and line segments in  $\mathfrak{R}^2$ , then  $V$  is  $O(n^2 \cdot 2^{\alpha(n)})$ .
- In two-dimensional cases the vertices of  $f(x)$  can be computed in time  $O(V \log V)$ . In three-dimensional cases the vertices of  $f(x)$  can be computed in time  $O(n^2 m^{1+\epsilon})$  for any  $\epsilon > 0$  ( $\epsilon$  appears in the constant).
- Given  $s$  convex polyhedra with a total of  $t$  faces, all of which contain a common point, we show that the boundary of their union has complexity  $O(st\alpha(st))$ .

The upper envelope of Voronoi surfaces can be used to solve the problem of determining the minimum Hausdorff distance under translation between two sets  $A$  and  $B$  of points in the plane [18]. The Hausdorff distance is a max–min distance that measures the mismatch between two point sets using some metric  $\rho$ . Thus the minimum Hausdorff distance under translation gives us a translation that minimizes the largest deviation of any point of one set from the other set. Given two sets  $A$  and  $B$  with  $p$  and  $q$  points, respectively, using our technique for finding the upper envelope of Voronoi surfaces of point sets in the plane, the translation minimizing the Hausdorff distance can be found in time  $O(pq(p+q) \log pq)$ . Our results also apply to point sets in  $\mathfrak{R}^3$ , and thus provide a solution to the problem of finding the translation that minimizes the Hausdorff distance between two point sets,  $A$  and  $B$ , in space. The running time of the method is  $O((pq)^2(p+q)^{1+\epsilon})$  for any  $\epsilon > 0$ , where  $p$  and  $q$  are again the number of points in  $A$  and  $B$ , respectively.

One generalization of the minimum Hausdorff distance is to compare sets of segments rather than sets of points. This problem is not directly reducible to the problem of computing the upper envelope of Voronoi surfaces. In fact solving this problem calls for computing the upper envelope of uncountably many Voronoi surfaces (see Section 3.2 for details). However, for the  $L_1$  and  $L_\infty$  metrics we show that the problem can be solved by taking the surfaces formed by the Minkowski sums of Voronoi surfaces and line segments. This yields an algorithm that computes the minimum Hausdorff distance between sets of line segments in the plane in time  $O((pq)^2 \alpha(pq))$ , where  $|A| = p$  and  $|B| = q$ . Recently (after the original submission of this paper), a similar bound has been shown for the  $L_2$  metric, by applying Meggido's parametric search technique [1]. These results compare favorably with the results of Alt *et al.* [2], who compute the minimum Hausdorff distance under translation between sets of line segments in the plane, under the  $L_2$  metric, in time  $O((pq)^3(p+q) \log(pq))$ .

The organization of the remainder of this paper is as follows. In the next section we analyze the complexity of the upper envelope of  $m$  Voronoi surfaces and show how to compute it efficiently. We consider Voronoi surfaces of point sets in  $\mathfrak{R}^2$  and  $\mathfrak{R}^3$ , as well as line segments in  $\mathfrak{R}^2$ . In Section 3 we discuss applications of Voronoi surfaces to the Hausdorff distance problems described above.

## 2. The Upper Envelope of Voronoi Surfaces

Recall from (1) the definition of the Voronoi surface of a set of sources,  $S$ , which we denote (together with its graph) by  $d(x)$ . By projecting  $d(x)$  onto its domain (the  $xy$ -plane in the planar case and the  $xyz$ -space in the three-dimensional case) we

obtain a decomposition of this domain into cells. By definition, this decomposition is exactly the Voronoi diagram of  $S$ . We define the *combinatorial complexity* of  $d(x)$  to be the number of  $k$ -faces of  $\text{Vor}(S)$  (over all  $k \leq d - 1$ ). We similarly define the combinatorial complexity of the upper envelope of  $m$  Voronoi surfaces,  $f(x)$  (equation (2)), to be the number of  $k$ -faces (over all  $k \leq d - 1$ ) of the projection of  $f(x)$  onto its domain. In this case a face is a connected component of the intersection of any fixed subset of faces of the surfaces  $d_i(x)$ , that appears on  $f(x)$ . We now determine the combinatorial complexity of  $f(x)$ , so defined, for sets of points in  $\mathbb{R}^2$ , for sets of line segments in  $\mathbb{R}^2$ , and for sets of points in  $\mathbb{R}^3$ .

### 2.1. Point Sets in the Plane

Let  $S_1, \dots, S_m$  be a collection of  $m$  sets of points in the plane. Here we determine the complexity of the upper envelope of the  $m$  Voronoi surfaces,  $d_i(x)$ , defined by these sets of points. In order to characterize the complexity of  $f(x)$  we make use of the following key observation, which follows immediately from the definition.

**Observation 1.** *At a given point  $x_0$ ,  $f(x_0) = d_i(x_0)$  exactly when the nearest source to  $x_0$  in  $\text{Vor}(S_i)$  is farther from  $x_0$  than is the nearest source to  $x_0$  in any other diagram  $\text{Vor}(S_j)$ .*

We use this key observation to determine the (possibly empty) portion of each cell of  $\text{Vor}(S_i)$  over which  $f(x) = d_i(x)$ . The union of these portions of all the cells of the  $m$  Voronoi diagrams then defines the ( $xy$ -projection of the) entire surface  $f(x)$ . The combinatorial complexity of the surface is by definition that of its  $xy$ -projection, and the surface is completely determined by this projection.

Fix one Voronoi diagram  $\text{Vor}(S_i)$ , and let  $F$  be the Voronoi cell of a given source  $q \in S_i$ . Now consider all the Voronoi diagrams  $\text{Vor}(S_i \cup S_j)$  for all  $j \neq i$ . Clearly,  $q$  is a source in each of these diagrams. Denote by  $Q_j$  the Voronoi cell of  $q$  in  $\text{Vor}(S_i \cup S_j)$  (see Fig. 1). It is easy to see that  $Q_j \subseteq F$  for each  $j$ . Let  $Q = \bigcup_{j \neq i} Q_j$ .

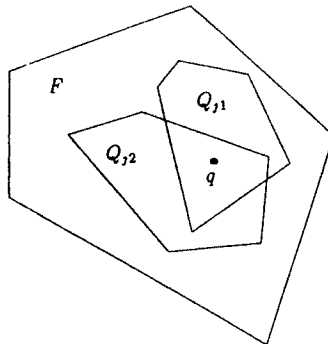


Fig. 1. The cells  $Q_j$  in the Voronoi cell  $F$  of a given source  $q \in S_i$ .

**Lemma 2.** *Using the above notation, the upper envelope  $f(x)$  is equal to  $d_i(x)$  for  $x \in F$  if and only if  $x \in F - Q$ .*

This result follows immediately from Observation 1 and the definition of the Voronoi diagram.

Let  $c_j$  denote the number of edges of each Voronoi cell  $Q_j, j \neq i$ , where  $Q_j$  is the cell of  $\text{Vor}(S_i \cup S_j)$  containing the point  $q \in S_i$ . Let  $c_F = \sum_j c_j$ . Since each Voronoi cell is star-shaped with respect to its source, we can introduce polar coordinates  $(r, \theta)$  with  $q$  as the origin, and regard the boundary of each  $Q_j$  as the graph of a function  $r = \gamma_j(\theta)$ . The boundary of  $Q = \bigcup_{j \neq i} Q_j$  is then the upper envelope of these functions.

**Lemma 3.** *The combinatorial complexity of the boundary of  $Q$  is linear in  $c_F$  for Voronoi diagrams based on  $L_1$  and  $L_\infty$  metrics; it is  $O(c_F \alpha(c_F))$  for all other  $L_p$  metrics.*

*Proof.* For Voronoi diagrams based on the  $L_2$  metric the Voronoi edges are line segments in arbitrary directions. Since any pair of them intersect at most once, their upper envelope is of complexity  $O(c_F \alpha(c_F))$  [16]. In the case of  $L_1$  and  $L_\infty$  metrics, the cell boundaries consist of line segments at only four orientations. Using a divide-and-conquer algorithm for computing upper envelopes [6] we combine the four families of segments (merging in pairs three times) to obtain an upper envelope whose complexity is linear in  $c_F$ .

For  $L_p, p \neq 1, 2, \infty$ , we show that any pair of edges of two Voronoi faces  $Q_k$  in  $\text{Vor}(S_i \cup S_k)$  and  $Q_l$  in  $\text{Vor}(S_i \cup S_l)$ , each containing  $q$ , can intersect at most once. Thus the complexity of the boundary of  $Q$  is just as for the  $L_2$  metric. Indeed, let  $e_k \subseteq \partial Q_k$  and  $e_l \subseteq \partial Q_l$  be two Voronoi edges. Assume  $q_k, q_l$  are the respective points in  $S_k, S_l$  that together with  $q$  induce the edges  $e_k, e_l$  in  $\text{Vor}(S_i \cup S_k), \text{Vor}(S_i \cup S_l)$ , respectively. Any intersection of  $e_k, e_l$  is thus a vertex of the Voronoi diagram  $\text{Vor}(\{q, q_k, q_l\})$ . The three Voronoi cells in this diagram can have exactly one common vertex: if the points are collinear, then the vertex is at infinity, otherwise the three point sources determine exactly one position where the convex shape of  $L_p$  touches all the three points, and this determines the only possible Voronoi vertex. □

Let  $N = \sum_F c_F$  denote the total number of edges bounding the faces  $Q_j$ , summed over all the Voronoi cells in all the diagrams.

**Lemma 4.**  $N = O(mn)$ .

*Proof.* For one Voronoi diagram,  $\text{Vor}(S_i)$ , we take all the Voronoi diagrams  $\text{Vor}(S_i \cup S_j), j \neq i$ , to create all the boundaries of all the  $Q_j$ 's within each face of  $\text{Vor}(S_i)$ . The number of Voronoi edges of each  $\text{Vor}(S_i \cup S_j)$  is  $O(n_i + n_j)$ . Summing over all indices  $j \neq i$ , gives

$$\sum_{F \in \text{Vor}(S_i)} c_F = O\left(\sum_{j \neq i} (n_i + n_j)\right) = O((m-1)n_i + (n - n_i)) = O(mn_i + n).$$

Summing this over all  $S_i, i = 1, \dots, m$ , we get  $\sum_i O(mn_i + n) = O(mn)$ . □

The main result of this section now follows immediately from Lemmas 2–4.

**Theorem 5.** *The combinatorial complexity of the upper envelope of Voronoi surfaces of  $m$  point sets in the plane, with a total of  $n$  points, is  $O(mn)$  when using the  $L_1$  or  $L_\infty$  metrics,  $O(mn\alpha(mn))$  for all other  $L_p$  metrics.*

We now proceed to show a lower bound on the number of vertices of  $f(x)$  in the worst case.

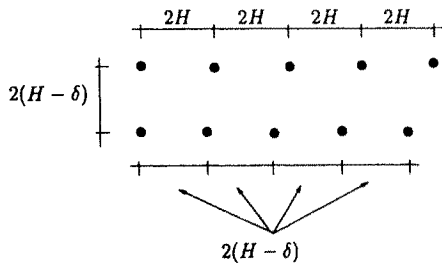
**Lemma 6.** *The number of vertices on the upper envelope of Voronoi surfaces is  $\Omega(mn)$  in the worst case.*

*Proof.* The construction we demonstrate here is for  $L_\infty$ -Voronoi diagrams, but it can be easily adapted for Voronoi diagrams based on other metrics.

The sets  $S_i$  are created in the following manner. We take all the sets  $S_i$  to have the same number of sources  $n_i = n/m$ . We fix two parameters,  $H \gg \delta > 0$ . We first construct  $S_1$ , placing its points on two parallel rows (see Fig. 2).

In the first row, lying on a line  $y = y_0$ , there are  $n_1/2$  points,  $2H$  apart from each other. The second row also has  $n_1/2$  points which lie on the line  $y = y_0 - 2(H - \delta)$ , for some sufficiently small  $\delta > 0$ , and are placed  $2(H - \delta)$  apart from each other.  $\text{Vor}(S_1)$  thus consists of two sets of parallel vertical edges (each having  $(n_1 - 1)/2$  edges), a set of  $n_1/2$  horizontal Voronoi edges, and other edges which are not important for the construction. (We choose the points on the two parallel lines to be slightly differently spaced in order to avoid degenerate edges in the construction.) Given this set of sources, the Voronoi surface  $d_1(x)$  has constant height on each vertical and horizontal Voronoi edge. For the vertical edges separating sources in the first row,  $d_1(x) \equiv H$ . For the vertical edges separating sources in the second row,  $d_1(x) \equiv H - \delta$ . For the horizontal Voronoi edges,  $d_1(x)$  is also  $H - \delta$ .

We now make  $m$  copies of  $S_1$ . The first  $m/2$  copies are translates of  $S_1$  to the right, in increments of some small amount  $\epsilon$ , where  $\epsilon > 2\delta$  (so that the lower horizontal edges will be visible on the upper envelope) and  $\epsilon \cdot m/2 < 2H$ . To get the set  $S_{m/2+1}$ , we translate  $S_1$  by  $H - \delta$  in the positive  $x$  direction; each  $S_i$ ,  $i = m/2 + 2, \dots, m$ , is a translation of  $S_{i-1}$ , in the positive  $y$  direction by  $\epsilon$ .



**Fig. 2.** The lower bound construction (the circles are points of  $S_1$ ).

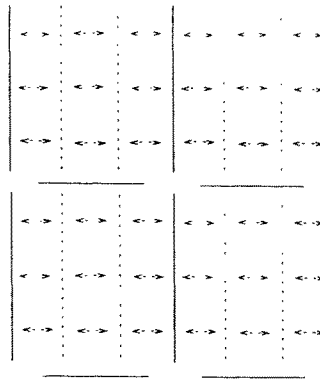


Fig. 3. The upper envelope  $f(x)$  of the worst-case construction.

The upper envelope  $f(x)$  of these surfaces is depicted in Fig. 3. The solid lines are (a view from above) of the surface  $d_1(x)$  (the Voronoi diagram of  $S_1$ ), and the dashed lines are all the other surfaces, each of which is a translated copy of  $d_1(x)$ . There are  $\Theta(n/m \cdot m)$  vertical edges over which the height of  $f(x)$  is  $H$ . In addition, the surfaces  $d_i(x)$  have  $\Theta(n)$  slightly lower (height  $H - \delta$ ) horizontal edges, arranged along  $m$  horizontal lines. On  $f(x)$ , these lines disappear below the vertical edges and reappear at small intervals between them. This produces  $\Omega(mn)$  edges on  $f(x)$ .

Although the value of  $d_i(x)$  on the vertical and horizontal edges is not constant for other  $L_p$  metrics the same construction of points leads to the same lower bound for these metrics as well. □

As a corollary, we obtain a worst-case lower bound on the combinatorial complexity of the upper envelope of Voronoi surfaces for all  $L_p$  metrics. In particular, we obtain:

**Corollary 7.** *The worst-case combinatorial complexity of the upper envelope of Voronoi surfaces of  $m$  point sets in the plane, with a total of  $n$  points, for the  $L_1$  and  $L_\infty$  metrics, is  $\Theta(mn)$ .*

We next turn to efficient computation of this envelope:

**Theorem 8.** *The upper envelope  $f(x)$  can be computed in time  $O(mn \log mn)$ .*

*Proof.* The proof of Theorem 5, which bounds the complexity of  $f(x)$ , is constructive. The algorithm follows this construction and has two basic steps. First, compute the “pair-diagrams”  $\text{Vor}(S_i \cup S_j)$  for each  $1 \leq i < j \leq m$ . Second, for each source  $q$  in each  $S_i$ ,  $1 \leq i \leq m$ , compute the boundary of  $Q = \bigcup_{j \neq i} Q_j$ , where each  $Q_j$  is the cell about the source  $q$  in  $\text{Vor}(S_i \cup S_j)$ . The boundary of  $Q$  is then just the upper envelope of segments in polar coordinates about  $q$ . This boundary specifies the structure of  $f(x)$  as described in Lemma 2.

The first step takes total time  $O(mn \log mn)$ , because the total size of all pairs

$S_i \cup S_j$  is  $\sum_{i,j}(n_i + n_j) = O(mn)$ . The second step takes time  $O(c_F \log c_F)$  for each source  $q$  [17]. Summing over all sources of all Voronoi diagrams, and making use of the fact that  $\sum_F c_F = O(mn)$ , yields the stated running times.  $\square$

**Lemma 9.** *Let  $q \in S_i$  be some source, let  $F$  be the Voronoi cell of  $\text{Vor}(S_i)$  containing  $q$ , and let  $Q$  be the region around  $q$  as defined above. Then the local minima of  $f(x)$ , for  $x \in F - Q$ , are obtained on the boundary of  $Q$ .*

*Proof.* The Voronoi cells  $Q_j$  and  $F$  are star-shaped with respect to  $q$  (convex polygons in the case of  $L_2$ ); see Fig. 1.  $Q$  is therefore also star-shaped with respect to  $q$ , and  $Q \subseteq F$  because each  $Q_j \subseteq F$ . The function  $d_i(x)$  is unimodal for  $x \in F$ , with the minimum  $d_i(x) = 0$  at  $x = q$ . Thus the local minima of  $f(x)$  for  $x \in F - Q$  must occur as close to the source as possible and due to the star-shaped property of  $Q$  this happens on the boundary of  $Q$ . Note that the boundary of  $Q$  may partially coincide with the boundary of  $F$ , and that this fact does not change the argumentation.  $\square$

Hence, the local minima of  $f(x)$  are all obtained on its edges or vertices. This enables us to obtain all local minima (and thus the global minimum) of  $f(x)$  at no extra cost.

## 2.2. Sets of Line Segments in the Plane

Let  $S_1, \dots, S_m$  be  $m$  sets of line segments in the plane, such that the segments in each set have pairwise disjoint relative interiors. Let the number of segments in each set  $S_i$  be  $n_i$ , and let  $n = \sum_{i=1}^m n_i$ . Since segments can share endpoints, we regard each  $S_i$  as a collection of (pairwise disjoint) open segments and their endpoints, and define the Voronoi diagram of such a new collection as in [25]. (The distance functions we consider here are the  $L_1$ ,  $L_2$ , and  $L_\infty$  metrics.) Actually, this analysis applies to any collection of pairwise disjoint open line segments and points, provided that the endpoints of the segments are included among the points.

We define  $d_i(x)$  and  $f(x)$  as in Section 1. Let  $q$  be a given open segment of some set  $S_i$  (the case where  $q$  is a point is simpler and is left for the reader), and let  $F$  be the Voronoi cell of  $q$  in  $\text{Vor}(S_i)$ . Let  $S_j$ ,  $j \neq i$ , be another set of segments. As before, we examine the face related to  $q$  in  $\text{Vor}(S_i \cup S_j)$ . However, the open segments of  $S_i$  and  $S_j$  are not necessarily pairwise disjoint. In fact, the source  $q \in S_i$  can be intersected by all the  $n_j$  segments of  $S_j$  in the worst case. Since the complexity of the Voronoi diagram is determined by the number of its pairwise disjoint sources, we decompose each open segment in  $S_i \cup S_j$  into the new open subsegments and the new endpoints formed by the intersections of segments in  $S_i$  and  $S_j$ .

We also make use of the following fact:

**Fact 10** [20]. *A Voronoi cell of a “line-segments Voronoi diagram” has the weak star-shape property, in the sense that if  $x$  is a point in the cell  $F$  of a source  $q$ , and*



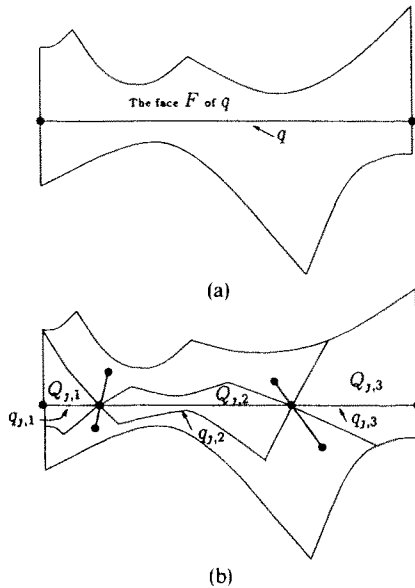


Fig. 4.  $Q_{j,l}$  in  $F$ .

if the point  $q'$  on  $q$  is the point closest to  $x$ , then the segment  $xq'$  is wholly contained in  $F$ .

Let  $k_j$  denote the number of new sources into which  $q$  is decomposed by segments of  $S_j$ , let the new sources themselves be denoted by  $q_{j,l}$ , and let the cells of  $q_{j,l}$  in  $\text{Vor}(S_i \cup S_j)$  be denoted by  $Q_{j,l}$  for  $l = 1, \dots, k_j$  (it actually suffices to consider only the new subsegments of  $q$ , because the new points along  $q$  have trivial cells in  $\text{Vor}(S_i \cup S_j)$ , as is easily checked). It is easy to see that  $Q_{j,l} \subseteq F$  for each  $j \neq i$  and for all  $l$ . Figure 4 illustrates this decomposition. Figure 4(a) shows the face  $F$  of  $q$  in  $\text{Vor}(S_i)$  and Fig. 4(b) shows the decomposition of the segment  $q$  into segments  $q_{j,l}$ , the dividing segments of  $S_j$ , the boundary of  $F$ , and the boundaries of  $Q_{j,l}$ . We denote by  $Q$  the union of all cells involving subsegments of  $q$  in all the Voronoi diagrams  $\text{Vor}(S_i \cup S_j)$ ,  $j \neq i$ , that is

$$Q = \bigcup_{j \neq i} \bigcup_{l=1}^{k_j} Q_{j,l}.$$

**Lemma 11.** Using the above notation,  $f(x) = d_i(x)$  for  $x \in F$  if and only if  $x \in F - Q$ .

*Proof.* We prove first that if  $x \in F - Q$ , then  $f(x) = d_i(x)$ . Again we apply the key observation from Section 2.1. Let  $x \in F$ , so the source closest to  $x$  in  $S_i$  is  $q$ . Due to Fact 10, all faces  $F$  and  $Q_{j,l}$  have the weak star-shape property with respect to  $q$ . If  $x \notin Q$ , then  $x \notin Q_{j,l}$  for each  $j \neq i$  and for each  $l = 1, \dots, k_j$ . Let  $q'$  be the point on  $q$  nearest to  $x$ . For each  $j \neq i$  let  $q_{j,l}$  be the subsegment (or endpoint) of  $q$  in

the decomposition of  $S_i \cup S_j$  which contains  $q'$ . Since  $x \notin Q_{j,l}$ , it follows that there is a source  $q^*$  in the decomposed set  $S_i \cup S_j$  such that  $q^*$  is closer to  $x$  than  $q_{j,l}$  is. However,  $q^*$  cannot be a subsegment of a segment in  $S_i$ ; indeed, since  $x$  is in  $F$ , the only way in which this can happen is when  $q^*$  is a subsegment of  $q$ ; but then since it is disjoint from  $q_{j,l}$ , it does not contain  $q'$ , so the distance of  $x$  from  $q^*$  is strictly greater than its distance from  $q_{j,l}$ , a contradiction. Hence  $q^*$  is a subsegment of some  $q_j \in S_j$  or a point  $q_j \in S_j$ . Therefore, for each  $j \neq i$ ,

$$d_j(x) = \rho(x, q_j) \leq \rho(x, q^*) < \rho(x, q) = d_i(x).$$

The other direction of the proof follows from similar arguments. □

Note that in the case where  $q$  is an endpoint, there is only one cell  $Q_{j,l}$  around  $q$  for any single Voronoi diagram  $\text{Vor}(S_i \cup S_j)$ , but the rest of the analysis remains the same.

We now proceed to estimate the number of vertices of  $f(x)$ . We denote by  $c_F$  the total number of edges of the cells  $Q_{j,l}$  for  $j \neq i$  and  $l = 1, \dots, k_j$  (i.e., the cells  $Q_{j,l} \subseteq F$ ).

**Lemma 12.** *The combinatorial complexity of the boundary of  $Q$  is  $O(c_F \alpha(c_F))$  for the  $L_1, L_\infty$  metrics, and is  $O(c_F \cdot 2^{\alpha(c_F)})$  for the  $L_2$  metric.*

*Proof.* As above, we only concentrate on the case where  $q$  is an open segment. For each segment  $q \in S_i$ , the star-shape property stated in Fact 10 allows us to interpret the boundary of each  $Q_{j,l}$  as a pair of graphs of partially defined functions over  $q$ , one graph on each side of  $q$ . In the case of  $L_1$  and  $L_\infty$  the Voronoi edges are straight line segments in arbitrary directions, so the upper envelope of the boundaries of the  $Q_{j,l}$ 's has complexity  $O(c_F \alpha(c_F))$ . For the  $L_2$  metric, the Voronoi edges are either straight lines or parabolic arcs, hence each pair of edges intersect at most twice, and the complexity of the boundary of  $Q$  is therefore

$$O(\lambda_4(c_F)) = O(c_F \cdot 2^{\alpha(c_F)}). \quad \square$$

Again we sum these bounds over all faces  $F$  of all the Voronoi diagrams  $\text{Vor}(S_i)$  in order to obtain the complexity of  $f(x)$ . If  $N = \sum_F c_F$ , then we get that the complexity of  $f(x)$  for sets of points and line segments is  $O(N \alpha(N))$  for the metrics  $L_1, L_\infty$ , and is  $O(N \cdot 2^{\alpha(N)})$  for the  $L_2$  metric.

**Lemma 13.**  $N = O(n^2)$ .

*Proof.* The number of Voronoi edges of each  $\text{Vor}(S_i \cup S_j)$  (with segments decomposed as above) is  $O(n_i \cdot n_j)$ . Summing this over all indices  $j \neq i$  gives

$$\sum_i \sum_{F \in \text{Vor}(S_i)} c_F = O\left(\sum_i \sum_j n_i \cdot n_j\right) = O(n^2). \quad \square$$

This lemma and the previous arguments yield:

**Theorem 14.** *The combinatorial complexity of the upper envelope of Voronoi surfaces of  $m$  sets of points and line segments in the plane, with a total of  $n$  sources, is  $O(n^2\alpha(n))$  when using the  $L_1, L_\infty$  metric, and is  $O(n^2 \cdot 2^{\alpha(n)})$  for the  $L_2$  metric. This upper envelope can be computed in time  $O(n^2 \log n)$  for the  $L_1, L_\infty$  metrics, and in time  $O(n^2\alpha(n) \log n)$  for the  $L_2$  metric.*

Note that, as in Section 2.1, the algorithmic bounds also follow directly from the previous lemma (we use Yap’s algorithm [25] to construct the pair diagrams  $\text{Vor}(S_i \cup S_j)$ ).

**Remarks.** The results of this section also hold for  $m$  sets of convex polygonal sources (rather than segments), for which the total number of edges of all of the sources is  $n$ . Moreover, the results for all the planar problems described above can be extended to convex polygonal distance functions instead of  $L_p$  metrics. A convex distance function is a pseudometric defined in terms of a convex body  $C$ , such that the distance from  $p$  to  $q$  is the smallest  $\delta$  such that, if a reference point in  $C$  is placed on  $p$ , then  $q \in \delta C$  [9], [15], [20].

### 2.3. Point Sets in 3-Space

We now consider the case where the sets  $S_1, \dots, S_m$  are point sets in 3-space. The Voronoi surfaces,  $d_i(x)$ , and their upper envelope  $f(x)$ , are defined as in the previous sections, though now they are surfaces in four-dimensional space. We restrict ourselves to the  $L_2$  metric only. The key observation of Section 2.1 applies here too and the analysis is similar to the previous two sections. However, the Voronoi cells are now three-dimensional convex polyhedra. Let  $q \in S_i$  be a source point, then  $F$ , the Voronoi cell of  $q$  in  $\text{Vor}(S_i)$ , is the set of all points such that each  $x \in F$  is closer to  $q$  than to any other source in  $S_i$ . Such a three-dimensional Voronoi cell can have up to  $n_i - 1$  faces, where  $n_i = |S_i|$ .

Following the previous sections, we construct all the pairwise Voronoi diagrams  $\text{Vor}(S_i \cup S_j)$ ,  $j = 1, \dots, M, j \neq i$ . We denote by  $Q_j$  the cell associated with  $q$  in  $\text{Vor}(S_i \cup S_j)$ . Clearly,  $Q_j \subseteq F$ , and the number of faces of a given Voronoi cell  $Q_j$  is bounded by  $n_i + n_j$ . We further observe that only at most  $n_j$  of these faces are contributed by the sources of  $S_j$ , the rest being portions of the original faces of the cell  $F$ . We define  $Q = \bigcup_{j \neq i} Q_j$ , and, as in the planar case, the upper envelope,  $f(x)$ , equals  $d_i(x)$  for  $x \in F$  if and only if  $x \in F - Q$ . Moreover, the local minima of  $f(x)$ , for  $x \in F - Q$ , are again obtained on the boundary of  $Q$ .

We turn to estimating the combinatorial complexity of  $f(x)$ . We fix  $q$  and  $F$  as above. Let  $c_j$  denote the number of faces of each cell  $Q_j, j \neq i$ , and let  $c_F = \sum_{j \neq i} c_j$ . There is a simple way to get an upper bound on the combinatorial complexity of  $Q$ . This bound is not the best possible, as is argued later, but we derive it anyway as a “warm-up” exercise.

We introduce spherical coordinates  $(r, \theta_1, \theta_2)$  with  $q$  as the origin, and regard

the boundary of each  $Q_j$  as the graph of a function  $r = \gamma_j(\theta_1, \theta_2)$ . We replace the graphs of these functions by a collection of  $O(c_F)$  triangles, which form triangulations of the boundaries of the  $Q_j$ . The upper envelope of these triangles (in our spherical coordinate system) gives the boundary of  $Q$ . An easy adaptation of the analysis given in [13] and [22] for the standard envelope of triangles implies that the maximal combinatorial complexity of  $Q$  is  $\Theta(c_F^2 \alpha(c_F))$ , and that it can be computed in time  $O(c_F^2 \alpha(c_F))$ .

We next improve this upper bound to  $O(mc_F \alpha(c_F))$ . To obtain this bound, we first prove the following result, which may be of independent interest:

**Theorem 15.** *The combinatorial complexity of the union of  $m$  convex polyhedra, all containing a common point  $O$  and having a total of  $t$  faces, is  $O(mt\alpha(mt))$ .*

*Proof.* Let  $K_1, \dots, K_m$  be  $m$  convex polyhedra in 3-space, all containing a common point, say the origin. Let  $K = \bigcup_{i=1}^m K_i$ . Denote by  $t_i$  the number of faces of  $K_i$ ,  $i = 1, \dots, m$ , so that  $t = \sum_{i=1}^m t_i$ . We divide the polyhedra into two subcollections of approximately equal size, referred to as the “red” polyhedra and the “blue” ones. Let  $R, B$  denote the union of the red and of the blue polyhedra, respectively; obviously  $K = R \cup B$ .

Our goal is to express the complexity of  $K$  in terms of the complexities of  $R$  and  $B$ ; the resulting recurrence gives us the desired bound. For this, we estimate the *increase* in the number of red faces caused by the addition of the blue polyhedra to the already existing union  $R$ , and vice versa, using a modified “combination lemma” (see [13]). Assume we start with  $R$ —the union of the red polyhedra. We add the faces of the blue polyhedra one at a time, exhausting the faces of each blue polyhedron before starting a new one, thus constructing  $K$  incrementally. Denote by  $\mathcal{U}$  the unbounded cell in the arrangement of the faces of all the red polyhedra and the blue faces added up until the previous step. At the end of the insertion process  $\mathcal{U}$  coincides with the complement of  $K$ .

Let  $F$  be the currently added blue face. Let  $G$  be a red face appearing on  $\partial\mathcal{U}$ . Suppose that  $G$  is affected by the addition of  $F$ . Since we want to estimate the increase in the number of red faces in  $\partial\mathcal{U}$  we are only interested in intersections of  $F$  and  $G$  that divide  $G$  into two or more subfaces which still show up on  $\partial\mathcal{U}$ ; only in this case do we experience an increase in the number of relevant red faces. Figure 5 shows such an intersection as seen on  $G$ :  $F \cap \text{int}(G)$  consists of six edges  $e_1, \dots, e_6$ .  $G$  is split by  $F$  into six subfaces. The dashed face does not show up on  $\partial\mathcal{U}$  (assume it just got *hidden* by adding  $F$  to the arrangement), and  $G_1, \dots, G_5$  do show up on  $\partial\mathcal{U}$ . We refer to each subface of  $G$  that shows up on  $\partial\mathcal{U}$  after the addition of  $F$  as a “surviving subface.” We refer to each edge of  $F \cap \text{int}(G)$  as a “splitting edge.”

We now show how to charge (all but one of) the surviving subfaces of  $G$  uniquely either to a splitting edge or to a reflex corner of  $G$ , and then we bound the number of such edges and corners.

The surviving subfaces of  $G$  can be thought of as forming the vertices of a “connectivity graph” whose edges connect pairs of surviving subfaces adjacent along a splitting edge (see Fig. 5). For each connected component of the connectivity graph we pick one node  $g$  and perform a depth-first search from  $g$ . Each

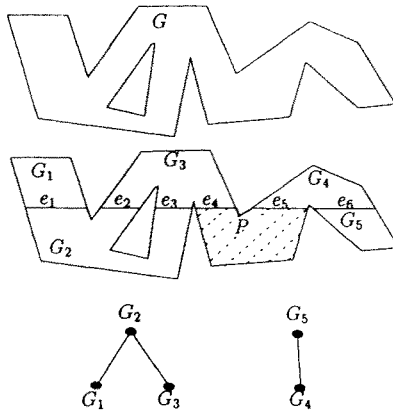


Fig. 5. An intersection as seen on  $G$ .

time the search reaches a new node we charge that surviving subspace to a splitting edge  $e$  through which we have crossed (if there are several such splitting edges, we charge one of them arbitrarily).

This accounts for all subspaces except the starting node of each connected component of the connectivity graph. If there are  $k$  connected components, then there must be some hidden subspaces of  $G$  dividing it into these components, so that each hidden subspace touches two or more surviving components at splitting edges, and all surviving components are connected to each other in this manner by hidden subspaces. It follows that these hidden subspaces must have  $k - 1$  pairwise disjoint paths along their boundaries, where each such path connects together the endpoints of two splitting edges (like the path connecting  $e_4$ ,  $P$ , and  $e_5$  shown by the bold line in Fig. 5). Each such hidden boundary path must have a reflex corner that has just disappeared from the unbounded cell and will never be seen again (like the corner marked  $P$  in Fig. 5). We charge the starting node of all but one of the  $k$  connected components of the connectivity graph to these  $k - 1$  reflex vertices.

Thus we have charged uniquely the increase in the number of surviving faces (the total number minus one) to splitting edges and to reflex vertices. The number of charges of the second kind is clearly  $O(mt)$ —these vertices are formed as intersections of original polyhedra edges with polyhedra faces, and convexity implies that each edge intersects at most two faces of any other polyhedron. Lemma 17, established below, shows that the number of charges of the first kind is  $O(mt\alpha(mt))$ .

We thus obtain the following recurrence:

$$|K| \leq |R| + |B| + cmt\alpha(mt)$$

for some absolute constant  $c > 0$ , where  $|K|$ ,  $|R|$ , and  $|B|$  denote the number of faces of these polyhedra. Unwinding this recurrence through our divide-and-conquer process yields a total of  $O(mt\alpha(mt))$ , as is easily checked. Since every vertex of  $K$  has degree at least three, Euler's formula implies that the overall combinatorial complexity of  $K$  is also  $O(mt\alpha(mt))$ . □

We turn to bounding the number of charged splitting edges (charges of the first kind). Consider the arrangement  $A_F$  on the blue face  $F$ , currently being added, which is induced by the straight line segments formed by intersecting  $F$  with all original red faces and previously added blue faces. Every splitting edge is an edge of  $A_F$ . We assign each charged splitting edge  $e$  to that face  $\phi_e$  of  $A_F$  that is incident to  $e$ , such that  $\phi_e$  is on the boundary of the unbounded cell  $\mathcal{U}$  (it is easily checked that only one of the two faces incident to  $e$  has this property). We thus obtain a collection of “marked” faces on  $F$ .

The following lemma is a crucial step in our analysis:

**Lemma 16.** *All marked faces  $\phi_e$  belong to the zone in  $A_F$  of the boundary of  $F$ , where the zone of a line in an arrangement is all the cells in the arrangement adjacent to this line (see [12]).*

*Proof.* Suppose to the contrary that there exists a marked face  $\phi_e$  that is internal in  $A_F$ . Let  $e$  be the charged splitting edge to which  $\phi_e$  is assigned— $e$  splits a red face  $G$  into two pieces, say  $G_1$  and  $G_2$ , such that both  $G_1$  and  $G_2$  appear on the boundary of  $\mathcal{U}$ . Let  $C$  be the cone spanned by all line segments connecting the origin to points on  $\phi_e$ . Since  $\phi_e$  is an internal face on  $A_F$ , each edge on its boundary is in the intersection of  $F$  with some red polyhedron or some previously added blue polyhedron. It follows that the boundary of the cone, not including  $\phi_e$ , is fully contained in the union of all red and previously added blue polyhedra (excluding the one containing  $F$ ), because it consists of segments connecting the origin to points on the boundaries of these polyhedra (see Fig. 6 where the segments labeled  $F$  and  $G$  are the cross section of the respective faces). Adding  $\phi_e$  “closes” the cone such that  $C$  is in the union of all the polyhedra, added up until now. Since the origin is contained in the polyhedra bounded by  $G$  and since  $\phi_e$  lies on  $\partial\mathcal{U}$ , it is easy to conclude that either  $G_1$  or  $G_2$  must be contained in the cone (at least locally near  $e$ ), hence in the union of the previous polyhedra, which contradicts the definition of  $e$  as a charged splitting edge.  $\square$

We now obtain the final lemma needed in the proof of Theorem 15.

**Lemma 17.** *The total number of charged splitting edges is  $O(mt\alpha(mt))$ .*

*Proof.* Fixing the blue face as above, let  $t_F$  denote the number of segments forming the arrangement  $A_F$ . The combinatorial complexity of the zone in  $A_F$  of the edges

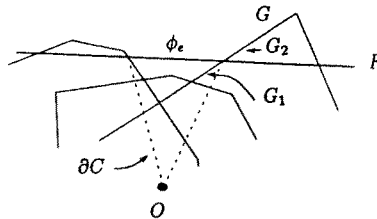


Fig. 6. A cross section of  $F$ ,  $G$ ,  $\phi_e$ , and the cone  $C$ .

of  $F$  is  $O(t_F \alpha(t_F))$  (see [12]). If we sum this over all blue faces  $F$ , we obtain that the total number of charged splitting edges, over the insertion of all blue polyhedra, is  $O(N\alpha(N))$ , where  $N = \sum_F t_F$ .

We claim that  $N = O(mt)$ . Indeed, fix a blue polyhedron  $K_j$ , and consider the sum  $N_j = \sum_{F \in K_j} t_F$ . Every segment counted by one of these  $t_F$ 's is an edge of the intersection of  $K_j$  with some red or previous blue polyhedron  $K_i$ . By Euler's formula, the number of such edges for a fixed  $K_i$  is  $O(t_i + t_j)$ . Summing this over all  $i$ , we obtain  $N_j = \sum_i O(t_i + t_j) = O(mt_j + t)$ , and summing that over all  $j$  we thus obtain  $N = \sum_j O(mt_j + t) = O(mt)$ , as asserted.  $\square$

We next outline an algorithm for computing the union  $K$ . We apply a divide-and-conquer scheme: partition the given collection of polyhedra into  $r$  roughly equal sets  $\mathcal{K}_1, \dots, \mathcal{K}_r$ , so that each set consists of roughly  $s/r$  polyhedra; here  $r$  is a sufficiently large constant, to be determined later. Let  $t_i$  denote the overall number of faces on the polyhedra in  $\mathcal{K}_i$ , for  $i = 1, \dots, r$ , where  $\sum_{i=1}^r t_i = t$ . For each  $1 \leq i < j \leq r$  define the subcollection  $\mathcal{K}_{ij} = \mathcal{K}_i \cup \mathcal{K}_j$ . We compute recursively the unions  $U_{ij}$  of the polyhedra in each of the subcollections  $\mathcal{K}_{ij}$ , and then merge them together to form the overall union  $K$ .

Assume we have computed all the unions  $U_{ij}$ . The merge step is performed as follows. Let  $v$  be a vertex on  $\partial K$ . It is either a vertex of an original polyhedron (which we call a vertex of type (i)), an intersection of an edge of some original polyhedron with a face of another (a vertex of type (ii)), or the intersection point of the relative interiors of faces of three distinct polyhedra (type (iii)). It is easy to compute vertices of  $\partial K$  of the first two types. We simply take each vertex  $v$  of the given polyhedra and test whether any of the other polyhedra contains  $v$  in order to determine the vertices of the first type. Similarly, we take each edge  $e$  of the given polyhedra and compute its intersection with each of the other polyhedra. Then we compute, for each edge  $e$ , the union of the corresponding intervals along  $e$ . The vertices of that union are vertices of type (ii) on  $\partial K$ .

The more difficult part is computing vertices of type (iii). Let  $v$  be such a vertex, which is an intersection point of three faces on the boundaries of three distinct polyhedra,  $K_1, K_2, K_3$ . If these polyhedra belong to just one or two of the families  $\mathcal{K}_i$ , then  $v$  will appear as a vertex of at least one of the unions  $U_{ij}$ . So suppose that these three polyhedra belong to three distinct subfamilies, and we may assume, with no loss of generality, that  $K_i \in \mathcal{K}_i$  for  $i = 1, 2, 3$ . To find all vertices  $v$  of this form, we merge  $U_{12}$  with  $U_{13}$  as follows. For each face  $F$  of any polyhedron (say,  $K_1$ ) of  $\mathcal{K}_1$ , we consider the portions  $F_{12}, F_{13}$  of  $F$  that lie on  $\partial U_{12}, \partial U_{13}$ , respectively. We compute the complement of the union  $F_{12} \cup F_{13}$  within  $F$ , using, e.g., the standard line-sweeping technique of [8], and this yields the portion of  $F$  that lies on the boundary of the union  $U_{123}$  of  $\mathcal{K}_1 \cup \mathcal{K}_2 \cup \mathcal{K}_3$ . Repeating this procedure for all faces of polyhedra in  $\mathcal{K}_1$ , we clearly obtain all vertices  $v$  of the form we seek.

However, the set of type (iii) vertices that we have computed in this manner is a superset of the set of actual type (iii) vertices of  $\partial K$ , and we still need to sift away spurious vertices. This is done as follows. We consider the collection of all edges  $e$  formed by the intersection (or union) of any two original polyhedra. We

also include in this collection edges of the original polyhedra. Let  $e$  be an edge in this collection, and suppose that  $e$  is formed by the intersection of two polyhedra,  $K_1, K_2$ , which, with no loss of generality, we assume to belong to  $\mathcal{K}_1, \mathcal{K}_2$ , respectively (the cases where  $K_1$  and  $K_2$  belong to the same family  $\mathcal{K}_i$ , or where  $e$  is an edge of an original polyhedron, are simpler to handle). Our task is to compute all portions of  $e$  that appear on the boundary  $\partial K$ . In the union  $U_{12}$ , the edge  $e$  may have already been split into several subintervals, so that the complement of their union lies strictly inside  $U_{12}$ . In addition, the merging procedure of, say  $U_{12}$  with  $U_{1i}$ , for any  $i = 3, \dots, r$ , will produce additional subintervals along  $e$ , which lie strictly inside the union  $U_{12i}$ . We can thus collect all these subintervals, and compute their union along  $e$ ; its complement gives all portions of  $e$  that appear on  $\partial K$ , and thus also all vertices of  $\partial K$  that appear on  $e$ . Repeating this over all edges  $e$  yields the collection of all actual vertices and edges of  $\partial K$ .

This almost produces the complete structure of  $\partial K$ . What remains is to “hook” together different connected components of the boundaries of nonsimply connected faces of  $\partial K$ , but this is also easily done, within the same time bound as above, by performing sweep over each original polyhedron face; we leave details of this procedure to the reader.

Using the method just described, we obtain

**Theorem 18.** *The union of  $m$  convex polyhedra in 3-space with a total of  $t$  faces, which have a point in common, can be computed in time  $O(m^{1+\epsilon}t)$  for any  $\epsilon > 0$  (where the constant of proportionality depends on  $\epsilon$ ).*

*Proof.* We follow the algorithm outlined above step by step. We first estimate the cost of the merge step. The vertices of  $\partial K$  of types (i) and (ii) can easily be computed in time  $O(mt \log t)$ . For vertices of type (i) we take each vertex  $v$  of the given polyhedra and test whether any of the other polyhedra contains  $v$ . Each such test is done in logarithmic time, and there are a total of  $O(mt)$  such tests to perform. Similarly, for vertices of type (ii), for each of the  $t$  edges  $e$  of the given polyhedra, we compute its intersection with each of the other polyhedra in time logarithmic in  $t$ . We get a total of  $O(mt)$  intervals, over all edges  $e$ . Computing the union of these intervals, for each edge  $e$ , is done in overall time  $O(mt \log t)$ .

For vertices of type (iii), the cost of computing  $F_{123} = F_{12} \cup F_{13}$  is

$$O((|F_{12}| + |F_{13}| + |F_{123}|) \log t)$$

[8], where  $|F_\alpha|$  denotes the complexity of the planar map  $F_\alpha$ . We sum this over all faces  $F$  of polyhedra in  $\mathcal{K}_1$ . Applying Theorem 15, we observe that

- $\sum_F |F_{12}|$  is bounded by the complexity of  $U_{12}$ , which is

$$O\left(\frac{2m}{r}(t_1 + t_2)\alpha(mt)\right).$$

- Similarly,  $\sum_F |F_{13}|$  is bounded by  $O((2m/r)(t_1 + t_3)\alpha(mt))$ .



- $\sum_F |F_{123}|$  is bounded by the complexity of  $U_{123}$ , which is

$$O\left(\frac{3m}{r} (t_1 + t_2 + t_3)\alpha(mt)\right).$$

We now repeat this procedure for all triples of distinct collections  $\mathcal{X}_i, \mathcal{X}_j, \mathcal{X}_k$ , and sum up the above bounds, to obtain a total running time of

$$O(mrt\alpha(mt) \log t) = O(mt\alpha(mt) \log t).$$

Sifting the spurious vertices from the superset of vertices of type (iii) is done in time  $O(mt\alpha(mt) \log t)$ : We have a total of  $O(mt)$  original edges of the polyhedra and edges formed by the intersection of any two original polyhedra. The latter can all be computed in time  $O(mt \log t)$ . Let  $e$  be an edge in this collection, and assume, without loss of generality,  $e \in \mathcal{X}_1 \cup \mathcal{X}_2$ . We gather all the subintervals of  $e$  which are in the union  $U_{12}$  and, throughout the merge process, we gather all the subintervals of  $e$  in  $U_{12i}, i = 3, \dots, r$ ; if the number of subintervals is  $x$ , then computing their union along  $e$  is done in time  $O(x \log x)$ . Since the total number of subintervals on all the edges in the collection is  $O(mt\alpha(mt))$ , we have established the time bound claimed for the sifting operation.

Hence, if  $T(m, t)$  denotes the time needed to compute the union of  $m$  convex polyhedra (sharing a common point) with a total of  $t$  faces, then we get the following recurrence:

$$T(m, t) \leq \sum_{1 \leq i < j \leq r} T\left(\frac{2m}{r}, t_i + t_j\right) + cmt\alpha(mt) \log t,$$

where  $c$  is an absolute positive constant. The first term in the equation above is the time needed for computing all the pairwise unions  $U_{ij}$ , and the second term (the overhead), is the time needed to compute all the  $\binom{r}{3}$  triples  $U_{ijk}$  by merging the relevant pairs.

We unfold the recurrence into a tree-like structure. At the second level of the tree a collection  $\mathcal{X}_i \cup \mathcal{X}_j$  is subdivided into  $r$  subcollections  $\mathcal{X}'_x, x = 1, \dots, r$ .  $t'_x$  denotes the total number of faces of the polyhedra in  $\mathcal{X}'_x$ . Thus at the second level we have inequalities of the form

$$T\left(\frac{2m}{r}, t_i + t_j\right) \leq \sum_{1 \leq k < l \leq r} T\left(\frac{4m}{r^2}, t'_k + t'_l\right) + c \frac{2m}{r} (t_i + t_j)\alpha(mt) \log t.$$

The sum of the overhead terms at the second level of the tree is therefore

$$\begin{aligned} \sum_{1 \leq i < j \leq r} c \frac{2m}{r} (t_i + t_j)\alpha(mt) \log t &= c \frac{2m}{r} \alpha(mt) \log t \sum_{1 \leq i < j \leq r} (t_i + t_j) \\ &= \frac{2(r-1)}{r} cmt\alpha(mt) \log t. \end{aligned}$$

The sum of the overhead terms at the  $j$ th level is similarly bounded by

$$\left(\frac{2(r-1)}{r}\right)^{j-1} cmt\alpha(mt) \log t,$$

so the total cost at all levels is thus

$$O\left(\left(\frac{2(r-1)}{r}\right)^h mt\alpha(mt) \log t\right),$$

where  $h$  is the depth of recursion, and is given by  $(r/2)^h = m$ . It follows that

$$T(m, t) = O((r-1)^h t\alpha(mt) \log t) = O(m^{\varepsilon(r)} t\alpha(mt) \log t),$$

where  $\varepsilon(r) = \log(r-1)/\log(r/2)$ , which we can make arbitrarily close to 1 by making  $r$  sufficiently large. □

Returning to the combinatorial complexity of  $f(x)$ , we now conclude

**Theorem 19.** *The complexity of the upper envelope,  $f(x)$ , of Voronoi surfaces in four dimensions is  $O(mn^2\alpha(mn))$ , and it can be computed in time  $O(n^2m^{1+\varepsilon})$  for any  $\varepsilon > 0$ .*

*Proof.* The proof is immediate. We use the notations from the beginning of this section. By Theorem 15, the complexity of  $Q$  is  $O(mc_F\alpha(c_F))$  and, by Theorem 18, we compute the boundary of  $Q$  in time  $O(m^{1+\varepsilon}c_F)$ . Computing  $Q$  for all the sources, and using the fact that  $\sum_F c_F = n^2$ , yields the stated bounds. □

### 3. Applications of the Envelope of Voronoi Surfaces

We now describe some applications that involve locating the global minimum on the upper envelope of Voronoi surfaces. We consider the problem of comparing two shapes, specified as sets of points or line segments. This is a central problem in pattern recognition, computer vision, and robotics. We use the minimum Hausdorff distance under translation (defined below) to determine the degree to which one shape differs from another. This technique both specifies a distance (shape difference) and specifies the best relative position (translation) of a “model” set  $A$  with respect to an “image” set  $B$ .

The Hausdorff distance between two sets,  $A = \{a_1, \dots, a_p\}$  and  $B = \{b_1, \dots, b_q\}$ , where each  $a_i, b_j$  is either a point or a line segment, is given by

$$H(A, B) = \max(h(A^*, B^*), h(B^*, A^*)), \tag{3}$$

where  $A^*$  (resp.  $B^*$ ) is the union of all points and segments in  $A$  (resp.  $B$ ),

$$h(A^*, B^*) = \max_{a \in A^*} \min_{b \in B^*} \rho(a, b), \tag{4}$$

and  $\rho(a, b)$  is the underlying metric. The function  $h(A^*, B^*)$  is the *directed* Hausdorff distance from  $A^*$  to  $B^*$ , and measures the distance of the point of  $A^*$  that is farthest from any point of  $B^*$  (under  $\rho$ ). It is well known that the function  $H(A, B)$  is a metric over the set of all closed, bounded sets. The Hausdorff distance,  $H(A, B)$ , can be trivially computed in time  $O(pq)$  for two point sets of size  $p$  and  $q$ , respectively; with some care, this can be improved to  $O((p + q) \log(p + q))$  [2]. In the special case where the sets  $A$  and  $B$  are respectively the vertices of a convex  $p$ -gon and a convex  $q$ -gon in  $\mathbb{R}^2$ , it has been shown that  $H(A, B)$  can be computed in time  $O(p + q)$  [5].

Without loss of generality we fix the set  $A$  and allow only  $B$  to translate. Thus we define the minimum Hausdorff distance,  $D(A, B)$ , between the sets  $A$  and  $B$  as

$$D(A, B) = \min_x H(A, B \oplus x), \tag{5}$$

where  $B \oplus x = \{b_j + x | b_j \in B\}$  (vector sum for point sets and Minkowski sum if  $b_j$  is a segment) and  $H$  is the Hausdorff distance as defined in (3). For example, Fig. 7 shows two sets of points in the plane, where the set  $A$  is illustrated by small circles and the set  $B$  by crosses.  $H(A, B)$  is large because there are points of  $A$  that are not near any points of  $B$  and vice versa.  $D(A, B)$  is small, however, because there is a translation of  $B$  that makes each point of  $A$  very close to some point of  $B$  and vice versa. For point sets in  $\mathbb{R}^1$ , it has recently been shown that  $D(A, B)$  can be computed in time  $O((p + q) \log(p + q))$  [24]. Here we investigate problems for sets of points in  $\mathbb{R}^2$  and  $\mathbb{R}^3$ , as well as for sets of line segments in  $\mathbb{R}^2$ .

We note that, for pattern-matching applications, it is desirable that the cost function used to compare two shapes should be a metric on shapes [4], [20]. A straightforward substitution argument establishes that  $D(A, B)$  is indeed a metric. An additional discussion of the Hausdorff distance and its applications is given in [2] and [18].

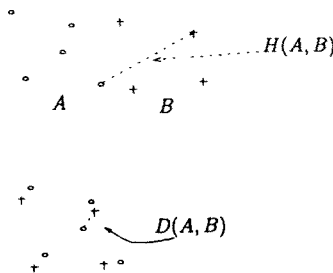


Fig. 7. Two sets of points illustrating the distances  $H(A, B)$  and  $D(A, B)$ .

A problem closely related to computing  $D(A, B)$  is that of finding the best approximate congruence under translation for two sets of  $p$  points,  $A$  and  $B$  [3]. Formally, the problem is to find the translation  $t$  of  $B$  and the bijection  $l: B \rightarrow A$  that minimizes

$$d = \max_{b \in B} \rho(b + t, l(b)). \tag{6}$$

In [3]  $\rho(\cdot, \cdot)$  is assumed to be either  $L_2$  or  $L_\infty$ . For point sets in the plane, the bijection, the translation, and  $d$  can be computed in time  $O(p^6 \log p)$  (compared with  $O(p^3 \log p)$  using the method described below for the minimum Hausdorff distance under translation).

### 3.1. Minimum Hausdorff Distance for Point Sets

We now describe how to compute the minimum Hausdorff distance

$$D(A, B) = \min_x H(A, B \oplus x)$$

for sets of points in  $\mathfrak{R}^2$  and  $\mathfrak{R}^3$ . The main idea is to consider the distances defined in (4) as functions that depend on the translation  $x$  of the set  $B$ . These functions, as we show below, are simply the Voronoi surfaces of the sets that result from translating  $B$  by each  $a_i$ , and from translating  $A$  by each  $b_j$ . Note that, because the sets  $A$  and  $B$  contain only points, in this section the unions  $A^*$  and  $B^*$  can be identified with  $A$  and  $B$ , respectively.

In more detail we can express the distance between a pair of points  $a_i \in A$  and  $b_j \in B$ , as  $b_j$  undergoes a translation  $x$ , by

$$\delta_{i,j}(x) = \rho(a_i, b_j + x) = \rho(a_i - b_j, x).$$

We then define the function  $d_i(x)$  to be the lower envelope of the functions  $\delta_{i,j}(x)$  for a fixed point  $a_i \in A$  and over all  $b_j \in B$ :

$$d_i(x) = \min_{b_j \in B} \delta_{i,j}(x). \tag{7}$$

If we denote the set  $a_i \ominus B$  by  $S_i$  (i.e.,  $S_i = \{a_i - b_j | b_j \in B\}$ ), then we obtain

$$d_i(x) = \min_{p \in S_i} \rho(p, x),$$

which is by definition (equation (1)) the Voronoi surface of  $S_i$ . Similarly, denoting the set  $A \ominus b_j$  by  $S'_j$ , the function

$$d'_j(x) = \min_{a_i \in A} \delta_{i,j}(x) = \min_{p \in S'_j} \rho(p, x)$$

is the lower envelope of the functions  $\delta_{i, f}(x)$  for a given  $b_j \in B$  and over all  $a_i \in A$ .

If we denote by  $f(x)$  the upper envelope of the functions  $d_i(x), d'_j(x)$ , then

$$f(x) = \max\left(\max_{a_i \in A} d_i(x), \max_{b_j \in B} d'_j(x)\right) = H(A, B \oplus x). \tag{8}$$

Hence

$$\min_x f(x) = \min_x H(A, B \oplus x) = D(A, B).$$

Thus, in order to determine the minimum Hausdorff distance between two sets  $A$  and  $B$ , where the set  $B$  is translated by  $x$ , we have to identify the value of  $x$  that minimizes the upper envelope of all the Voronoi surfaces defined by the sets  $S_i = a_i \ominus B$  and  $S'_j = A \ominus b_j$ . Hence we obtain

**Lemma 20.** *The number of local minima of  $f(x)$  is  $O(pq(p + q))$  for the metrics  $L_1$  and  $L_\infty$ , and  $O(pq(p + q)\alpha(pq))$  for all other  $L_p$  metrics.*

*Proof.* Using the notation in Section 2.1, we substitute in the proofs of Lemma 6 and Theorem 5  $m = p + q, n_i = p$  for the first  $q$  indices and  $n_i = q$  for the last  $p$  indices. Thus  $n = 2pq$  and the result then follows immediately.  $\square$

In order to determine  $D(A, B)$  we must identify the global minimum of  $f(x)$  which can be done by calculating all the local minima and inspecting each of them. Applying Theorem 8, we obtain the first main result of this section:

**Theorem 21.** *The minimum Hausdorff distance under translation between two sets of points in the plane (and the translation that achieves this minimum) can be computed in time  $O(pq(p + q) \log pq)$ , where  $p$  and  $q$  are the sizes of the two sets being compared.*

When the sets  $A = \{a_1, \dots, a_p\}$  and  $B = \{b_1, \dots, b_q\}$  consist of points in  $\mathfrak{R}^3$  and the metric is  $L_2$ , we use the analysis of Section 2.3. The sets  $a_i \ominus B$  are all of size  $q$  and the sets  $A \ominus b_j$  are all of size  $p$ . Thus, in the terminology of Section 2.3, we have  $m = p + q$  and  $n = 2pq$ , leading to

**Theorem 22.** *The minimum Hausdorff distance under translation between two sets of points in  $\mathfrak{R}^3$  based on the  $L_2$  metric (and the translation that achieves this minimum) can be computed in time  $O((pq)^2(p + q)^{1+\epsilon})$  for any  $\epsilon > 0$  (where the constant proportionally depends on  $\epsilon$ ); here  $p$  and  $q$  are the sizes of the two sets being compared.*

### 3.2. Minimum Hausdorff Distance for Sets of Segments

The problem of computing the minimum Hausdorff distance for sets of line segments and points can also be solved by the technique of upper envelopes of

Voronoi surfaces, although some extra care is needed here. The reason for the new difficulties is that now there is a substantial difference between the finite sets  $A, B$  and the infinite sets  $A^*, B^*$  (the unions of the elements of  $A$  and of  $B$ , respectively). Thus our technique would call for computing the upper envelope of uncountably many Voronoi surfaces, each having a set of sources obtained as the Minkowski differences of segments in  $A$  and a point in  $B^*$ , or of a point in  $A^*$  and the segments of  $B$ . However, if the underlying metric is  $L_1$  or  $L_\infty$ , we show that it suffices instead to form the upper envelope of only a finite number of surfaces, each obtained by computing the Minkowski sum of the Voronoi surface of  $\text{Vor}(A)$  with a segment of  $B$ , or the Minkowski sum of the Voronoi surface of  $\text{Vor}(B)$  with a segment of  $A$ .

Let  $A = \{a_1, \dots, a_p\}$  and  $B = \{b_1, \dots, b_q\}$  be two sets of points and segments in the plane, where we require that the segments are all open, and that if a set contains an (open) segment it also contains its endpoints; in other words, each closed line segment appears in a set as three distinct (and pairwise disjoint) sites—its relative interior and its endpoints (note that segments can share endpoints).

Define as above  $A^* = \bigcup_{a \in A} a$  and  $B^* = \bigcup_{b \in B} b$ . We want to compute

$$D(A, B) = \min_x H(A, B \oplus x) \\ = \min_x \max \left( \max_{y \in A^*} \min_{z \in B^*} \rho(y, z + x), \max_{z \in B^*} \min_{y \in A^*} \rho(z + x, y) \right),$$

where  $\rho$  is the underlying metric, which we assume in this section to be  $L_1$  or  $L_\infty$  only.

The above min-max-min expression is equal to

$$\min_x \max \left( \max_{a_i \in A} \max_{y \in a_i} \min_{z \in B^*} \rho(x, y - z), \max_{b_j \in B} \max_{z \in b_j} \min_{y \in A^*} \rho(x, y - z) \right).$$

Note that for any point  $y \in a_i$  and for every  $x$  we have

$$\min_{z \in B^*} \rho(x, y - z) = \min_{b_j \in B} \rho(x, y - b_j),$$

and that the right-hand side of this equation is the Voronoi surface of the line segment Voronoi diagram of (the reflected)  $B$  translated by  $y \in A^*$ . Similarly, for the other minimization, we have, for any point  $z \in b_j$  and for every  $x$ ,

$$\min_{y \in A^*} \rho(x, y - z) = \min_{a_i \in A} \rho(x, a_i - z),$$

which is the Voronoi surface of  $\text{Vor}(A)$  translated by  $z \in B^*$ . That is, in the inner minimization portions of these expressions, we can minimize over (translated)

objects in  $A$  or in  $B$  and not over their unions. We denote these Voronoi surfaces by

$$d_y(x) = \min_{b_j \in B} \rho(x, y - b_j), \quad d_z(x) = \min_{a_i \in A} \rho(x, a_i - z)$$

for each  $y \in A^*$ ,  $z \in B^*$ .

Next we define the upper envelope  $f(x)$  of the surfaces  $d_y(x)$  and  $d_z(x)$  over all  $y \in A^*$  and  $z \in B^*$ :

$$f(x) = \max \left( \max_{y \in A^*} d_y(x), \max_{z \in B^*} d_z(x) \right),$$

which we can rewrite as

$$f(x) = \max \left( \max_{a_i \in A} \max_{y \in a_i} d_y(x), \max_{b_j \in B} \max_{z \in b_j} d_z(x) \right).$$

It follows that  $f(x)$  is the upper envelope of at most  $p + q$  surfaces, each defined either by  $D_i(x) = \max_{y \in a_i} d_y(x)$  or by  $D_j(x) = \max_{z \in b_j} d_z(x)$ .  $D_i(x)$  is the upper boundary of the volume obtained by sweeping the surface of  $\text{Vor}(-B)$  horizontally along  $a_i$ ;  $D_j(x)$  has a similar interpretation.

Let us fix a segment  $a_i \in A$ . Denote the endpoints of this segment by  $a'$  and  $a''$ . For each face  $F$  of  $d_{a_i}(x)$ , when  $y \in a_i$  moves from  $a'$  to  $a''$ , the face  $F$  is swept horizontally along the segment  $a_i = a'' - a'$ . The resulting swept volume is simply the Minkowski sum  $F \oplus a_i$ , which we denote by  $\mathcal{F}$ . For general metrics, including  $L_2$ , the structure of the swept volume is rather complicated, but in the  $L_1$  and  $L_\infty$  metrics the structure is quite straightforward. In these cases each such face  $F$  is a polygon, hence the boundary of  $\mathcal{F}$  is a prism whose two bases are parallel to  $F$  and all its other sides are parallelograms whose parallel edges are parallel to  $a_i$ . It follows that we can represent the swept surfaces as a collection of  $O(pq)$  triangles: every  $D_i$ , for  $i = 1, \dots, p$ , has  $O(q)$  triangles, and every  $D_j$ , for  $j = 1, \dots, q$ , has  $O(p)$  triangles, hence a total of  $O(pq)$ .

The upper envelope  $f(x)$  is the upper envelope of  $O(pq)$  triangles, so its complexity is  $O((pq)^2 \alpha(pq))$  [22], and it can be computed in time  $O((pq)^2 \alpha(pq))$  [13], getting us to:

**Theorem 23.** *The minimum Hausdorff distance under translation between two planar sets  $A$  and  $B$ , of  $p$  and  $q$  line segments respectively, based on the  $L_1$  or  $L_\infty$  metric, can be computed in time  $O((pq)^2 \alpha(pq))$ .*

This result has recently been extended to computing the minimum Hausdorff distance between sets of line segments under the  $L_2$  metric [1]. That paper uses Meggido's parametric search technique to obtain a running time of

$$O((pq)^2 \log^3(pq)).$$

It is interesting to compare the above theorem and the results of [1] with those of [2], that compute the minimum Hausdorff distance, under translation, between sets of line segments in the plane, in time  $O((pq)^3(p+q)\log(pq))$ . It is argued in [2] that approximation methods are needed because of the high complexity of the problem. However, our results and those of [1] already improve these bounds considerably and further improvements might still be possible.

#### 4. Summary

We have investigated the combinatorial complexity of the upper envelope of  $m$  superimposed *Voronoi surfaces* under various metrics, where each surface is induced by some set of sources  $S_i$ . For example, in the Euclidean metric, we have shown that when the sources are points in  $\mathcal{R}^2$  and the total number of sources is  $n = \sum_i |S_i|$ , then the combinatorial complexity of the upper envelope of  $m$  Voronoi surfaces is  $O(mn\alpha(mn))$ . This bound is relatively tight, as we have exhibited an  $\Omega(mn)$  lower bound. For points in  $\mathcal{R}^3$ , the corresponding complexity of the upper envelope is  $O(mn^2\alpha(mn))$ . Establishing this latter bound involves a new result on the complexity of the union of  $m$  convex polyhedra containing a common point. If the total number of faces of the polyhedra is  $t$ , then the complexity of the boundary of their union is  $O(mt\alpha(mt))$ .

The upper envelope of Voronoi surfaces can be used to solve min-max-min optimization problems efficiently. In particular, we considered the problem of computing the minimum Hausdorff distance between two sets under translation.

Recently there has been progress on improving the running time of methods for computing the minimum Hausdorff distance under translation. For the  $L_1$  and  $L_\infty$  metrics, and two sets  $A$  and  $B$  with, respectively,  $p$  and  $q$  points in the plane, the distance can be computed in time  $O(pq \log^2(pq))$  [10]. This raises the question whether the  $L_2$  problem can also be solved in nearly quadratic time. The technique of [10] does not seem to extend to the  $L_2$  case.

There have also been some recent results on the problem of computing the minimum Hausdorff distance for sets of points under Euclidean motion (as opposed to just translation). In [19] an  $O((p+q)^6 \log(pq))$ -time method was presented for planar point sets, which is based on bounding the number of changes in the dynamic Voronoi diagrams of unions of the sets  $S_i$ . This involved the derivation of a new bound on the complexity of dynamic Voronoi diagrams of unions of  $k$  rigidly moving subsets of sources. The resulting bound for the Hausdorff distance is still much too large to be of any practical use. Thus another interesting open question is to obtain better bounds for computing the minimum Hausdorff distance under rigid motion, and to obtain tight bounds for any of the minimum Hausdorff distance problems.

#### References

1. Agarwal, P. K., Sharir, M., and Toledo, S., Applications of parametric searching in geometric optimization, *Proc. Third ACM-SIAM Symposium on Discrete Algorithms*, 1992, pp. 72-82.



2. Alt, H., Behrends, B., and Blömer, J., Measuring the resemblance of polygonal shapes, *Proc. Seventh ACM Symposium on Computational Geometry*, 1991, pp. 186–193.
3. Alt, H., Mehlhorn, K., Wagener, H., and Welzl, E., Congruence, similarity, and symmetries of geometric objects, *Discrete & Computational Geometry*, **3** (1988), 237–256.
4. Arkin, E., Chew, L. P., Huttenlocher, D. P., Kedem, K., and Mitchell, J. S. B., An efficiently computable metric for comparing polygonal shapes, *Proc. First ACM–SIAM Symposium on Discrete Algorithms*, 1990, pp. 129–137.
5. Atallah, M. J., A linear time algorithm for the Hausdorff distance between convex polygons, *Information Processing Letters*, **17** (1983), 207–209.
6. Atallah, H. J., Dynamic computational geometry, *Proc. 24th IEEE Symposium on Foundations of Computer Science*, 1983, pp. 92–99. Also in *Computers & Mathematics with Applications*, **11** (1985), 1171–1181.
7. Aurenhammer, F., Voronoi diagrams—a survey, *ACM Computer Surveys*, **23** (1991), 345–405.
8. Bentley, J. L., and Ottmann, T., Algorithms for reporting and counting geometric intersection, *IEEE Transactions on Computers*, **28** (1979), 643–647.
9. Chew, L. P., and Drysdale, R. L., Voronoi diagrams based on convex distance functions, *Proc. First ACM Symposium on Computational Geometry*, 1985, pp. 235–244.
10. Chew, L. P., and Kedem, K., Improvements on approximate pattern matching problems, *Proc. Scandinavian Workshop on Algorithm Theory—SWAT '92*, O. Nurmi and E. Urkkonen, editors, Lecture Notes in Computer Science, Vol. 621, Springer-Verlag, Berlin, 1992, pp. 318–325.
11. Edelsbrunner, H., *Algorithms in Combinatorial Geometry*, Springer-Verlag, Heidelberg, 1987.
12. Edelsbrunner, H., Guibas, L., Pach, J., Pollack, R., Seidel, R., and Sharir, M., Arrangements of curves in the plane: topology, combinatorics, and algorithms, *Theoretical Computer Science*, **92** (1991), 319–336.
13. Edelsbrunner, H., Guibas, L. J., and Sharir, M., The upper envelope of piecewise linear functions, *Discrete & Computational Geometry*, **4** (1989), 311–336.
14. Edelsbrunner, H., and Seidel, R., Voronoi diagrams and arrangements, *Discrete & Computational Geometry*, **1** (1986), 25–44.
15. Fortune, S., A fast algorithm for polygon containment by translation, *Proc. ICALP 1985*, Lecture Notes in Computer Science, Vol. 194, Springer-Verlag, Berlin, 1985, pp. 189–198.
16. Hart, S., and Sharir, M., Non-linearity of Davenport–Schinzel sequences and of generalized path compression schemes, *Combinatorica*, **6** (1986), 151–177.
17. Hershberger, J., Finding the upper envelope of  $n$  line segments in  $O(n \log n)$  time, *Information Processing Letters*, **33** (1989), 169–174.
18. Huttenlocher, D. P., and Kedem, K., Efficiently computing the Hausdorff distance for point sets under translation, *Proc. Sixth ACM Symposium on Computational Geometry*, 1990, pp. 340–349.
19. Huttenlocher, D. P., Kedem, K., and Kleinberg, J. M., On dynamic Voronoi diagrams and the minimum Hausdorff distance for point sets under Euclidean motion in the plane, *Proc. Eighth ACM Symposium on Computational Geometry*, 1992, pp. 110–119.
20. Leven, D., and Sharir, M., Intersection and proximity problems and Voronoi diagrams, *Advances in Robotics*, Vol. 1, J. T. Schwartz and C. K. Yap, editors, Erlbaum, Hillsdale, NJ, 1987, pp. 187–228.
21. Mumford, D., The problem of robust shape descriptors, *Proc. First International Conference on Computer Vision*, 1987, pp. 602–606.
22. Pach, J., and Sharir, M., The upper envelope of piecewise linear functions and the region enclosed by convex plates, I: combinatorial analysis, *Discrete & Computational Geometry*, **4** (1989), 291–309.
23. Preparata, F. P., and Shamos, M. I., *Computational Geometry*, Springer-Verlag, New York, 1985.
24. Rote, G., Computing the minimum Hausdorff distance between two point sets on a line under translation, *Information Processing Letters*, **38** (1991), 123–127.
25. Yap, C. K., An  $O(n \log n)$  algorithm for the Voronoi diagram of the set of simple curve segments, *Discrete & Computational Geometry*, **2** (1987), 365–393.

Received February 18, 1991, and in revised form October 26, 1992.