On Dynamic Voronoi Diagrams and the Minimum Hausdorff Distance for Point Sets Under Euclidean Motion in the Plane*

Daniel P. Huttenlocher[(1)], Klara Kedem[(1,2)] and Jon M. Kleinberg[(1)]

[(1)]Department of Computer Science
Cornell University, Ithaca, NY 14853, U.S.A.

[(2)]Department of Computer Science
Tel-Aviv University, Tel-Aviv 69978, Israel

Abstract

We show that the dynamic Voronoi diagram of k sets of points in the plane, where each set consists of n points moving rigidly, has complexity $O(n^2k^2\lambda_s(k))$ for some fixed s, where $\lambda_s(n)$ is the maximum length of a (n,s) Davenport-Schinzel sequence. This improves the result of Aonuma et. al., who show an upper bound of $O(n^3k^4\log^* k)$ for the complexity of such Voronoi diagrams. We then apply this result to the problem of finding the minimum Hausdorff distance between two point sets in the plane under Euclidean motion. We show that this distance can be computed in time $O((m+n)^6\log(mn))$, where the two sets contain m and n points respectively.

*This work was supported in part by NSF grant IRI-9057928 and matching funds from General Electric and Kodak, and in part by AFOSR under contract AFOSR-91-0328. The second author was also supported by the Eshkol grant 04601-90 from The Israeli Ministry of Science and Technology.

1. Introduction

Determining the degree to which two shapes differ from one another is a central problem in pattern recognition and machine vision. Recently a number of papers in computational geometry have investigated various aspects of the problem of measuring the difference between shapes (e.g., [ABB, AMWW, AST, ACHKM, HK, HKS]). In general, two geometric objects A and B are considered to have the same shape if there exists a transformation $g \in G$ such that $g(A) = B$, where G is a given transformation group. The difference between shapes is then measured using some function $D(A, B) \geq 0$, where a value of zero is attained exactly when A and B have the same shape.

In this paper we consider the problem of comparing point sets under the group of planar Euclidean motions (i.e., the composition of translation and rotation in the plane). Such transformations are often also referred to as 'rigid body' motion. We measure the difference between two point sets A and B by minimizing the Hausdorff distance under all relative Euclidean motions of the two sets. (The Hausdorff distance is a max-min distance, and is defined below.) We then show that the minimum Hausdorff distance under Euclidean motion can be computed in time $O((m+n)^6 \log(mn))$, where the sets A and B contain m and n points respectively.

The minimum Hausdorff distance has been used to compare a number of different kinds of shapes. The minimum distance between sets of points under translation in the plane can be computed in time $O(mn(m+n)\alpha(mn)\log(mn))$ where m and n are, as above, the numbers of points in the two sets being compared [HK, HKS]. If the sets contain line segments as well as points, then the difference between two sets under translation can be computed in time $O(m^2n^2\log^3(mn))$ [AST] for L_2 as the underlying metric, and in time $O(m^2n^2\alpha(mn))$ [HKS], for L_1 and L_∞ as the underlying metrics (where $\alpha(n)$ is the extremely slowly growing inverse Ackermann function). Under rigid body motion, the minimum Hausdorff distance between two polygons can be computed in time $O((mn)^4(m+n)\log(m+n))$ [ABB].

In [HKS] the problem of computing the minimum Hausdorff distance under translation between sets A and B with m and n points respectively, is reduced to that of computing the upper envelope of $m+n$ Voronoi surfaces of $O(m+n)$ source points each. This upper envelope problem can be solved by computing the unions of certain cells in pairwise Voronoi diagrams. In this paper we also show a relation between computing the minimum Hausdorff distance and problems concerning Voronoi diagrams. In particular, we show how the minimum Hausdorff distance under Euclidean motion can be solved using the *dynamic* Voronoi diagram of a special point set, in which subsets of the points move rigidly.

To obtain our results on the minimum Hausdorff distance under Euclidean motion, we improve a known bound on the complexity of the following dynamic Voronoi diagram problem. Consider k sets of points, $T_1, \ldots, T_k$, where each set consists of n points in the plane, and each set is allowed to move rigidly in some constrained motion. We show that the complexity of the dynamic Voronoi diagram of $T = \cup_{i=1}^{k} T_i$ is $O(n^2k^2\lambda_s(k))$ for some fixed s, where $\lambda_s(n)$ is the maximum length of a (n, s) Davenport-Schinzel sequence [ASS, HS]). This improves a result by Aonuma, Imai, Imai and Tokuyama [AIIT] who show an upper bound of $O(n^3k^4\log^* k)$ for the complexity of such Voronoi diagrams. When k is bounded by a constant then we get a bound of $O(n^2)$, whereas [AIIT] get

$O(n^3)$. If there are n sets each with one point (which corresponds to the general dynamic Voronoi diagram problem) then our bound becomes $O(n^2\lambda_s(n))$, which is the same as that attained by [FL, GMR]. Our results on dynamic Voronoi diagrams depend on a new bound on the length of a special Davenport-Schinzel sequence (see Lemma 1 below), a result which is interesting in itself.

This paper is organized as follows. In the following section we define the minimum Hausdorff distance under Euclidean motion, and show its relation to dynamic Voronoi diagrams. In Section 3 we provide bounds on the above-described dynamic Voronoi diagram of k sets moving rigidly. In Section 4 we show that the minimum Hausdorff distance under Euclidean motion can be determined by considering the combinatorial changes to certain unions of Voronoi cells. We use our results on dynamic Voronoi diagrams to bound the number of such changes. Section 5 then presents an algorithm for computing the minimum Hausdorff distance under Euclidean motion, and analyzes its run time and storage requirements.

2. The Minimum Hausdorff Distance

The Hausdorff distance between two sets, $A = \{a_1, \ldots, a_m\}$ and $B = \{b_1, \ldots, b_n\}$, where each a_i, b_j is a point, is given by

$$H(A,B) = \max(h(A,B), h(B,A)) \tag{1}$$

where

$$h(A,B) = \max_{a\in A}\min_{b\in B}\rho(a,b), \tag{2}$$

and $\rho(a,b)$ is some 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 ρ). 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 $O(mn)$ for two point sets of size m and n respectively; with some care, this can be improved to $O((m+n)\log(m+n))$ [ABB].

Given two point sets A and B in the plane, we note that all relative configurations of these two sets under rigid motion can be obtained by the composition of a rotation about the origin and a translation. Hence without loss of generality we let the set A just rotate about the origin, while the set B just translates. We then define the minimum Hausdorff distance under Euclidean motion, $D(A,B)$, between the sets A and B as

$$D(A,B) = \min_{x,\theta} H(r_\theta(A), B\oplus x) \tag{3}$$

where $r_\theta(A) = \{R_\theta a_i \mid a_i \in A\}$ (and $R_\theta a_i$ denotes the product of the rotation matrix R_θ with a_i), $B\oplus x = \{b_j + x \mid b_j \in B\}$ (and $+$ denotes vector sum), and H is the Hausdorff distance as defined in equation (1).

For applications in pattern matching and model-based recognition, it is important that a shape comparison function obey the metric properties (non-negativity, identity, symmetry and the triangle inequality) [ACHKM, Mu]. These properties correspond to our intuitive notions of shape resemblance. than they are to one another. For example, without the triangle inequality two highly dissimilar 'model' shapes can both be judged similar to some unknown shape, which is highly counter-intuitive. We note that $D(A, B)$ obeys the metric properties, and this can be shown by straightforward substitution arguments analogous to those in [HKS] where it is shown that the minimum Hausdorff distance under translation obeys metric properties.

A problem closely related to computing $D(A, B)$ is that of determining whether there exists an approximate congruence between two sets of points under rigid body motion [AMWW]. Formally, the problem is to determine whether there is a Euclidean motion E of B, a bijection $l : B \rightarrow A$, and a distance

$$d = \max_{b \in B} \rho(E(b), l(b)), \tag{4}$$

such that the distance d is less than some specified value ϵ. It is assumed that $\rho(.,.)$ is either L_2 or L_∞. For two point sets in the plane with p points each, it can be determined in time $O(p^8)$ whether there exists a motion E and a bijection l such that $d < \epsilon$ [AMWW]. In our method we measure the difference between two sets by minimizing the maximal mismatch between two point sets under Euclidean motion. The key difference is that the distance computation is not concerned with *matching pairs* of points. It can be argued for each measure which one fits a given application better, however, we compute the minimum Hausdorff distance in time $O(p^6 \log p)$ as opposed to $O(p^8)$ for the approximate congruence. Moreover, our measure is a metric whereas the approximate congruence is not.

Computing $D(A, B)$

We now describe how to compute the minimum Hausdorff distance $D(A, B) = \min_{x,\theta} H(r_\theta(A), B \oplus x)$ for A and B, where $A = \{a_1, \ldots, a_m\}$ and $B = \{b_1, \ldots, b_n\}$ are two sets of points in the plane. First we define a function $f(x, \theta)$ that specifies the value of $H(r_\theta(A), B \oplus x)$, and then we outline how the minimum value of this function can be computed. In order to compute the minimum efficiently, we require some additional tools which are developed in the following sections.

The distance between a pair of points $a_i \in A$ and $b_j \in B$, where a_i undergoes a rotation by θ and b_j undergoes a translation by x, is simply

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

The function $d_i(x, \theta)$ is then defined to be the (pointwise) minimum of the functions $\delta_{i,j}(x, \theta)$ for a given point $a_i \in A$ and over all $b_j \in B$:

$$d_i(x, \theta) = \min_{b_j \in B} \delta_{i,j}(x, \theta). \tag{5}$$

This function gives the distance from a given point $a_i \in A$ to the closest point of B (the distance between a_i and the set B), as a function of the motion parameters x and θ.

The directed Hausdorff distance as a function of Euclidean motion, $h(r_\theta(A), B \oplus x)$, which we denote also by $f_A(x, \theta)$, is then given by the upper envelope of the d_i functions,

$$f_A(x, \theta) = \max_{a_i \in A} d_i(x, \theta). \tag{6}$$

The analogous definitions of

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

and

$$f_B(x,\theta) = \max_{b_j \in B} d'_j(x,\theta),$$

specify the directed distance $h(B \oplus x, r_\theta(A))$ as a function of Euclidean motion. Then the distance is simply

$$D(A,B) = \min_{x,\theta} \max(f_A(x,\theta), f_B(x,\theta)). \tag{7}$$

To simplify the discussion, we concentrate on the case of minimizing the directed distance as given by $f_A(x,\theta)$ in equation (6), rather than the Hausdorff distance as given by equation (7). We note that f_A is the maximum of m functions d_i, where each d_i is the minimum of n functions $\delta_{i,j}$. Similarly f_B is the maximum of n functions d'_j, each of which is the minimum of m functions $\delta_{i,j}$. Equation (7), which maximizes f_A and f_B, is then just the maximum of $m+n$ functions. Hence the bounds that we obtain for the directed distance, f_A, will be simple to generalize to f_B and also to the distance $D(A,B)$.

In order to determine the minimum value of $f_A(x,\theta)$ over all x, θ, we first consider the restricted case in which θ is fixed. That is, the set A is (temporarily) at some fixed orientation, and the set B is allowed to translate. This is thus the problem of computing the Hausdorff distance under translation, as considered by [HK, HKS]. Following [HKS], we note that if the set $R_\theta a_i \ominus B$ is denoted by $S_{i,\theta}$ (i.e., $S_{i,\theta} = \{R_\theta a_i - b_j | b_j \in B\}$) then

$$d_{i,\theta}(x) = \min_{p \in S_{i,\theta}} \rho(p,x),$$

where we use the notation $d_{i,\theta}(x)$ rather than $d_i(x,\theta)$ to stress the fact that θ is fixed. The graph of this function, $\{(x, d_{i,\theta}(x)) | x \in R^2\}$, is by definition the Voronoi surface of the set of *source* points $S_{i,\theta}$. That is, the function $d_{i,\theta}(x)$ specifies for each location x the distance to the nearest point of the set $S_{i,\theta}$. The local maxima of this surface (where x is equidistant from two or more points of $S_{i,\theta}$) occur by definition for values of x that lie along the edges of the Voronoi diagram of $S_{i,\theta}$.

The basic observation of [HKS] was that the upper envelope of a set of $d_{i,\theta}(x)$ functions could be completely described by the pairwise Voronoi diagrams $\mathrm{Vor}(S_{i,\theta} \cup S_{j,\theta})$ for all $i \neq j$. Consider a given diagram $\mathrm{Vor}(S_{i,\theta})$, and let F be the Voronoi cell of a given source point $s \in S_{i,\theta}$. Further consider all the Voronoi diagrams $\mathrm{Vor}(S_{i,\theta} \cup S_{j,\theta})$, 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 $\mathrm{Vor}(S_{i,\theta} \cup S_{j,\theta})$ (see Figure 1). It is easy to see that $Q_j \subseteq F$ for each j. Let $Q = \bigcup_{j \neq i} Q_j$, and denote the boundary of Q by ∂Q.

Fact 1 *Using the above notation, the upper envelope,* $\max_i d_{i,\theta}(x)$ *is equal to* $d_{i,\theta}(x)$ *for* $x \in F$ *if and only if* $x \in F - Q$. *Moreover, the minima of the upper envelope are all attained along* ∂Q. *[HKS].*

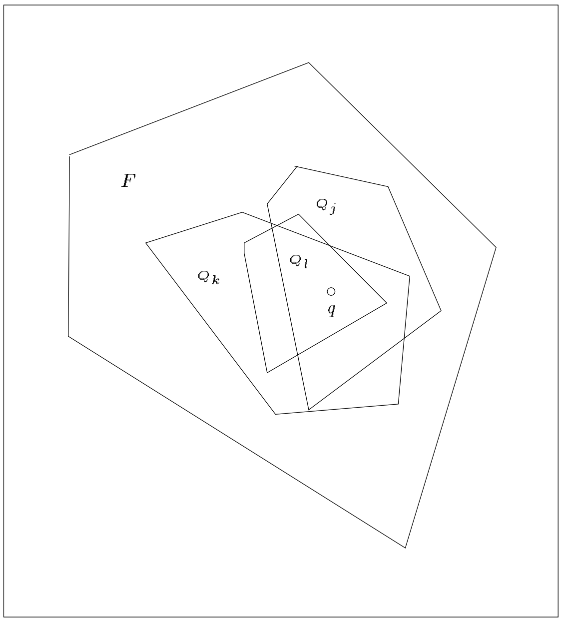

Figure 1: The cells Q_j in the Voronoi cell F of a given source $s \in S_i$.

Thus for any fixed θ, the local minima of $f_A(x, \theta)$ (and hence the global minimum) can be determined by simply computing the pairwise Voronoi diagrams of the sets $S_{i,\theta}$ for all $i \neq j$, and then computing ∂Q (the boundary of the union) for each source point q in each set $S_{i,\theta}$. In [HKS] it was shown that the total complexity of ∂Q over all the source points is $O(m^2 n\alpha(mn))$, where there are m sets $S_{i,\theta}$ each with n points. Moreover, the computation of the minimum can be done in time $O(m^2 n\alpha(mn)\log(mn))$.

As θ changes we can view the functions $d_i(x, \theta)$ as the evolution across 'time' (the parameter θ) of the Voronoi surface $d_i(x)$. That is, the points of each set S_i can be viewed as moving according to a circular motion parameterized by θ. (For notational simplicity we write S_i rather than $S_i(\theta)$.) As the points move in this way, the edges of each Q_j change continuously (in size and position) and some edges of the Q_j's are created and destroyed. This causes changes to the set of edges visible on ∂Q.

Each edge e on ∂Q is a portion of a bisector b_{pq}, for some p in some S_j (and $q \in S_i$, the source of each Q_j). Moreover the left (right) endpoint of e is caused by the intersection of b_{pq} with another bisector b_{rq} (b_{sq}). Thus, the triples pqr and pqs induce the endpoints of e. We label each edge by (p, q, r, s, θ_0), where e first appears on ∂Q at θ_0. The inclusion of θ_0 ensures that e is uniquely labeled. Thus e has an *epoch* of visibility, $[\theta_0, \theta_0']$ where θ_0' is the value of θ for which the edge ceases to be visible on ∂Q.

In Section 4 we show that over each edge, the minimum value of $f_A(x, \theta)$ can be computed in constant time. Thus, we are able to bound the total complexity of minimizing $f_A(x,\theta)$ in terms of the number of edges that ever appear on the boundary of each cell Q (one for each point q in each S_i). In order to bound this total number of edges we show a correspondence between the appearance of edges on ∂Q and topological events in certain dynamic Voronoi diagrams. These dynamic Voronoi diagrams are of a restricted form, because they are composed of unions of sets S_i, where each set S_i moves rigidly (in a circular motion as $S_i = \{R_\theta a_i - b_j | b_j \in B\}$). First, however, we develop some new tools for bounding the number of such topological events.

Thus, in summary, we have the following plan. First we develop some general tools for analyzing the complexity (number of topological changes) of dynamic Voronoi diagrams of a set $T = \cup_i T_i$, where each set T_i moves rigidly. Then we apply these results in order to bound the number of edges that can ever become visible on all ∂Q. Finally we show that the minimum of $f_A(x,\theta)$ can be computed in constant time for each edge on ∂Q (over its epoch of visibility). By Fact 1, minimizing over all the edges that become visible on every ∂Q yields the global minimum of $f_A(x,\theta)$.

3. The Dynamic Voronoi Diagram of k Sets Moving Rigidly

The dynamic Voronoi diagram problem involves a set T of n points in the plane where each point moves according to some continuous function of time, and the motions are restricted in some fashion (e.g., to be fixed-degree polynomials). The problem is to bound the number of topological changes in the Voronoi diagram of T over all time. A topological change in the Voronoi diagram occurs when either a Voronoi edge ceases to exist or a new edge appears, due to changes in proximity relations among the points of T. We restrict ourselves to the case where $T = \cup_{i=1}^{k} T_i$, and each T_i is a set of n points in the plane. We assume the points of each T_i are in *general position* (i.e. no four points cocircular). The points of each set T_i move rigidly in a motion described by the function $\{f_i\}$, where $f_i : I \rightarrow \Re^2$ is a continuous function of time $t \in I \subset \Re^1$, and I is a finite closed interval. In what follows we take I to be $[0,1]$, without loss of generality. As in [FL, GMR] the motion is constrained by the following property,

Property 1 *Four points can become cocircular or collinear at most s times, where s is a constant.*

This property is achieved, for example, by motions that are low-degree polynomials.

The creation or destruction of a bounded edge of $\mathrm{Vor}(T)$, which we also call an *event*, corresponds to four points of T becoming cocircular on an empty circle (i.e., a circle that does not contain any other points of T). The creation or destruction of an unbounded edge corresponds to three points becoming collinear on the convex hull of T. As in [FL, GMR], we discuss only the changes in the bounded edges of the Voronoi diagram. The analysis of the unbounded edges follows analogously.

First we consider two special cases, where T consists of either two or three subsets, each of which moves rigidly as just described. In each of these cases we show that the complexity of the dynamic Voronoi diagram is $O(n^2)$. (The same bounds for these cases were obtained by [AIIT]; however, we present our proofs because they are helpful in understanding the general case of $k \geq 4$ moving sets.)

Claim 1 *Given two sets of points in the plane T_1 and T_2, $|T_1|, |T_2| = n$, each set moving rigidly in a motion that has Property 1, let $T = T_1 \cup T_2$. Then the complexity of the dynamic Voronoi diagram of T over time is $O(n^2)$.*

Proof. Denote the four points that become cocircular on an empty circle by p_1, p_2, p_3 and p_4. We are only interested in the cases where these four points do not all come from the same set. Thus there are two cases: three points come from one set and one from the other, or two points come from each set.

(i) Assume $p_1, p_2, p_3 \in T_1$, $p_4 \in T_2$. Since p_1, p_2 and p_3 are cocircular on an empty circle they define a vertex in $\mathrm{Vor}(T_1)$, and since $Vor(T_1)$ does not change over time this is true for all t. There are $O(n)$ vertices in $\mathrm{Vor}(T_1)$, and hence $O(n)$ such point triples. Each of these triples can become cocircular with each of the n points of T_2. By Property 1, each such cocircularity can only happen $O(1)$ times, and thus there are $O(n^2)$ events of this type.

(ii) Assume $p_1, p_2 \in T_1$, $p_3, p_4 \in T_2$. Since p_1 and p_2 are cocircular on an empty circle they determine an edge in $\mathrm{Vor}(T_1)$, and since $\mathrm{Vor}(T_1)$ does not change as a function of t this is true for all t. Similarly p_3 and p_4 define an edge in $\mathrm{Vor}(T_2)$. $\mathrm{Vor}(T_1)$ and $\mathrm{Vor}(T_2)$ have $O(n)$ edges each, hence there are $O(n^2)$ such sets of four points. By Property 1, a given set of four points can be cocircular $O(1)$ times, and thus there are $O(n^2)$ events of this type.

∎

Claim 2 *Given three sets T_1, T_2 and T_3, $|T_1|, |T_2|, |T_3| = n$, let $T = T_1 \cup T_2 \cup T_3$. The complexity of the dynamic Voronoi diagram of T over time is $O(n^2)$.*

Proof. As in Claim 1, we count the number of times four points $p_1, p_2, p_3, p_4 \in T$ become cocircular on an empty circle. Clearly two of these points must come from the same set, T_i. Assume without loss of generality that $p_1, p_2 \in T_1$. Then p_1 and p_2 determine a bounded edge e in $\mathrm{Vor}(T_1)$, with the endpoints of e due respectively to the points $p_k, p_l \in T_1$.

We define the *configuration* of p_1 and p_2 at time t to be the triple $((p_1, p_2), (q_1, q_2), (r_1, r_2))$ (see Figure 2). The points q_1 and q_2 are those points of $T_1 \cup T_2$ inducing the endpoints of the visible portion of e in $\mathrm{Vor}(T_1 \cup T_2)$ at time t. If e is not visible in $\mathrm{Vor}(T_1 \cup T_2)$, then we use the notation $(q_1, q_2) = (\phi, \phi)$. Thus $(q_1, q_2) \in ((T_2 \cup \{p_k\}) \times (T_2 \cup \{p_l\})) \cup \{(\phi, \phi)\}$. Similarly, r_1 and r_2 are the points causing the endpoints of e in $\mathrm{Vor}(T_1 \cup T_3)$, and $(r_1, r_2) \in ((T_3 \cup \{p_k\}) \times (T_3 \cup \{p_l\})) \cup \{(\phi, \phi)\}$.

For a given configuration, p_1 and p_2 can only become cocircular with q_1, q_2, r_1, r_2. We charge all of these cocircularities to this configuration; note that there are only $O(1)$ of these charges. Define a *change* to the configuration of p_1 and p_2 to be a change to one of the q_i or r_i $(i = 1, 2)$. Thus by bounding the number of changes to all configurations over time, we bound the number of cocircularities. Specifically, we show how to charge each configuration change for $p_1, p_2 \in T_1$ to a cocircularity event in $\mathrm{Vor}(T_1 \cup T_2)$ or in $\mathrm{Vor}(T_1 \cup T_3)$.

Initially (for $t = 0$) there are $O(n)$ configurations, one for each edge $e \in \mathrm{Vor}(T_1)$. Consider a given configuration $((p_1, p_2), (q_1, q_2), (r_1, r_2))$. The pair (q_1, q_2) can only change in three different ways (and analogously the pair (r_1, r_2) which we do not consider explicitly):

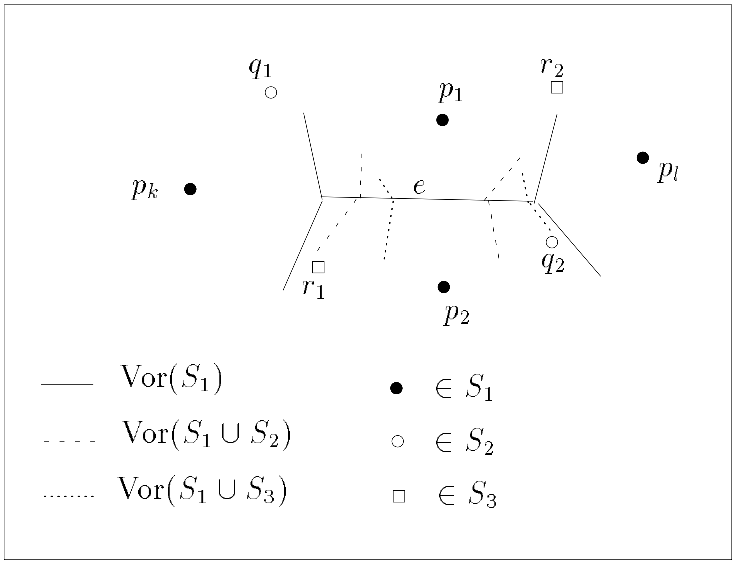

Figure 2: The configuration of p_1 and p_2

(i) The point $q_1 \in T_2$ (or alternatively q_2) is replaced by another point $q_1' \in T_2$ (alternatively $q_2' \in T_2$). In this case the four points p_1, p_2, q_1 and q_1' become cocircular. This is a cocircularity event in $T_1 \cup T_2$.

(ii) The point q_1 (or alternatively q_2) is replaced by p_k (or alternatively p_l). The reverse situation, of p_k being replaced by q_1 (or p_l by q_2), is analogous. This is a cocircularity of p_1, p_2, p_k and q_1 in $T_1 \cup T_2$, (in the alternative case the cocircularity involves q_2, p_l, p_1, p_2).

(iii) The pair (q_1, q_2) is replaced by (ϕ, ϕ) or vice versa, which corresponds to the edge e becoming invisible or visible in $\mathrm{Vor}(T_1 \cup T_2)$. This is a cocircularity involving p_1, p_2, q_1 and q_2 in $T_1 \cup T_2$.

We have thus shown that each way in which a configuration can change involves a cocircularity event in a pairwise diagram $\mathrm{Vor}(T_1 \cup T_2)$, (or analogously $\mathrm{Vor}(T_1 \cup T_3)$), and thus we can charge e configuration change to such an event. Note that there are only $O(1)$ charges to each event in a pairwise diagram, because each charge involves a specific four points becoming cocircular, which can only happen s times. There are $\binom{3}{2}$ such pairwise diagrams, each of complexity $O(n^2)$, for a total of $O(n^2)$. ∎

We now proceed to the case of k point sets moving rigidly for arbitrary k. We first prove a general lemma which is interesting in itself. We denote by $\mathrm{DS}(n,s)$ the (n,s) Davenport-Schinzel sequence with n symbols, $a_1, \ldots, a_n$, where each pair of symbols can interchange at most s times (see [ASS, HS] for an overview of these sequences including bounds on their lengths). We note that the length of a $\mathrm{DS}(n,s)$ sequence is $\lambda_s(n)$. Let μ be a $\mathrm{DS}(n,s)$ sequence and let $\mu(j)$ denote the

jth entry in the sequence μ. We call the symbol a_i *active* at entry j if $\mu(j') = a_i$ for some $j' \leq j$ and $\mu(j'') = a_i$ for some $j'' > j$. Define $\nu(j)$ to be the number of active symbols at entry j. We call μ a DSA(n,s,m) sequence if μ is a DS(n,s) sequence and for each j, $\nu(j) \leq m$. Let us denote by $\lambda_{s,m}(n)$ the maximum length of DSA(n,s,m) sequence.

Lemma 1 *$\lambda_{s,m}(n)$ is $O(\frac{n}{m}\lambda_s(m))$.*

Proof. Let μ be an arbitrary DSA(n,s,m) sequence. We partition it into disjoint blocks in the following manner. We scan μ from its beginning until we have the shortest subsequence of μ that has $2m$ distinct symbols. Call this block B_1, remove it from μ, and denote the remainder of μ by μ_1. We now repeat this process on μ_1: regardless of the symbols that have appeared in B_1 we scan μ_1 from its beginning for the shortest leading subsequence, B_2, that has $2m$ distinct symbols. We iterate this process. Since μ is of finite length, the process ends when we have produced L blocks $B_1, \ldots, B_L$ for some value of L.

We claim that each block B_i, $i < L$, contains at least m symbols that do not appear in μ_i. We prove this by contradiction; since there are $2m$ distinct symbols in B_i we assume that $m+1$ of these symbols appear in the remainder μ_i. Let us denote by c the index in μ of the last element of the block B_i. Then there are $m+1$ symbols that appear both at or before entry c, and after entry c in μ; i.e. $\nu(c) \geq m+1 > m$, which contradicts the property that $\nu(j) \leq m$ for each j.

It is immediate from this claim that $L \leq (\frac{n}{m}+1)$. Each block is a DS$(2m,s)$ sequence; therefore its length is $\lambda_s(2m)$. Multiplying by L we get that $\lambda_{s,m}(n)$ is $O(\frac{n}{m}\lambda_s(m))$. ∎

We now turn to the main result of this section. As above, we restrict the discussion to cases where the Voronoi edges are bounded. We define a k-configuration by analogy with the configuration of Claim 2. However, in this case, we form a configuration for each possible bisector between any pair of points in $T = \cup_{i=1}^{k} T_i$. For the moment we assume that the points come from two distinct sets, let $p \in T_i$, $q \in T_j$, $i \neq j$ be the two points. Denote the k-configuration of the bisector b_{pq} by $\kappa_{pq} = ((p,q), (l_{i_1}, r_{i_1}), \ldots, (l_{i_{k-2}}, r_{i_{k-2}}))$. A pair (l_m, r_m) in κ_{pq} is the the pair of points from the union $T_i \cup T_j \cup T_m$, $m \neq i,j$, that induce the *left* and *right* endpoints of the edge $e \subset b_{pq}$ in Vor$(T_i \cup T_j \cup T_m)$. (By *left* and *right* endpoints of e, we mean the leftmost endpoint on e and the rightmost one with respect to the vector pq.) Hence $(l_m, r_m) \in ((T_i \cup T_j \cup T_m) \times (T_i \cup T_j \cup T_m)) \cup \{(\phi,\phi)\}$, (where (ϕ,ϕ) indicates that the bisector b_{pq} does not show up as a Voronoi edge in Vor$(T_i \cup T_j \cup T_m)$).

A change to κ_{pq} is defined to be a change to any one of the l_m or r_m, e.g., a new $l'_m \in T_i \cup T_j \cup T_m \cup \{\phi\}$ replaces l_m in the k-configuration. By our definitions this is a topological event in Vor$(T_i \cup T_j \cup T_m)$: point l_m induced the endpoint on the edge e before the change, and point l'_m induces the endpoint on e after the change, and at the time of the change, points p, q, l_m and l'_m are cocircular. (In the case where l_m is replaced by ϕ, the points p, q, l_m and r_m become cocircular, which is analogous to Claim 2 case (iii).)

Let $e_{pq} \subset b_{pq}$ be the (possibly empty) Voronoi edge of p and q in Vor(T). Let N_{pq} denote the number of topological events over all time that involve e_{pq}. We define C_{pq} to be the total number of changes to κ_{pq}, plus $k-2$. The bulk of the proof consists of bounding the number of topological events, N_{pq}, by $O(\lambda_{s,(k-2)}(C_{pq}))$ (i.e. $O(\frac{C_{pq}}{k}\lambda_s(k))$).

Theorem 1 *The combinatorial complexity of the dynamic Voronoi diagram for k rigid sets of n points each is $O(k^2 n^2 \lambda_s(k))$.*

Proof. Consider κ_{pq} between two consecutive changes to it. Clearly if any of the (l_m, r_m) are equal to (ϕ, ϕ), then the edge e_{pq} is not visible in Vor(T). Call κ_{pq} *full* if no (l_m, r_m) is equal to (ϕ, ϕ). When κ_{pq} is full, e_{pq} will be visible if and only if all the left endpoints induced by the points l_m lie to the left of all the right endpoints induced by the points r_m. More specifically, for κ_{pq} full, call the rightmost left endpoint induced by one of the points l_m the *leading* left endpoint; analogously, call the leftmost right endpoint induced by one of the points r_m the *leading* right endpoint. Then e_{pq} is visible if and only if the leading left endpoint lies to the left of the leading right endpoint; these are in fact the endpoints of e_{pq} in this case. Thus, a topological event involving e_{pq} is precisely the crossing of a leading left endpoint and a leading right endpoint; i.e. a cocircularity of p, q, and the points l_m and $r_{m'}$ that induce these endpoints.

Since any four such points can become cocircular only s times, we can bound N_{pq} by bounding the number of distinct leading endpoints that appear on the bisector b_{pq} over all time. We show how to bound the length of the sequence of leading left endpoints; the bound for right leading endpoints is the same, and merging these two sequences yields a sequence of at most twice this length.

For notational clarity, we actually bound the length of the sequence of l_m that induce leading left endpoints. We do this by representing it as an appropriate Davenport-Schinzel sequence μ_{pq}. The alphabet for our DS sequence is constructed as follows. At $t = 0$ we start with a symbol for each l_m $(1 \le m \le k-2)$ in κ_{pq} that is not equal to ϕ. Then, at each change to κ_{pq}, where l_m is replaced by a non$-\phi$ l'_m, we add l'_m to the alphabet. The following point is important: even if l'_m has appeared in κ_{pq} at some earlier time, we still add a *new* symbol to represent it.

μ_{pq} is defined to be the sequence of symbols that represent the leading left endpoints over all time. Observe the following: no two symbols of μ_{pq} can interchange more than s times, by Property 1. Also, the size of the alphabet of this sequence is bounded by $k-2$ plus the total number of changes to κ_{pq}; i.e. it is bounded by C_{pq}. Thus μ_{pq} is a DS(C_{pq}, s) sequence.

In fact, μ_{pq} is a DSA$(C_{pq}, s, k-2)$ sequence. This is clear from the construction of the alphabet. But as argued above, the number of topological events, N_{pq}, involving bisector b_{pq} is bounded by a constant multiple of the length of this sequence (and the strictly analogous one constructed for leading right endpoints). Thus we have shown that

$$N_{pq} = O(\lambda_{s,(k-2)}(C_{pq})) = O(\frac{C_{pq}}{k}\lambda_s(k)). \qquad (8)$$

For fixed i and j, the sum of C_{pq} is bounded by the total number of topological events in all Vor$(T_i \cup T_j \cup T_m)$, for all $m \neq i, j$, plus the original $k-2$ values at $t = 0$ for each p, q. The second of these terms is clearly bounded by $O(kn^2)$, since there are $O(n^2)$ pairs p, q. To bound the first of the terms, we note that the total number of k-configuration changes due to T_m for fixed m is $O(n^2)$ (Claim 2); summing for all $k-2$ possible values of m yields $O(kn^2)$. Thus we get

$$\sum_{p,q} C_{pq} = O(kn^2).$$

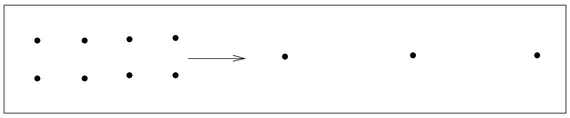

Figure 3: An $\Omega(n^2)$ construction

Summing up C_{pq} over all sets i, j and points in T_i and T_j we get

$$C = \sum_{i,j} \sum_{p,q} C_{pq} = O(k^3 n^2). \tag{9}$$

We substitute C instead of C_{pq} in equation 8 and get the complexity of the dynamic Voronoi diagram of k sets moving rigidly:

$$N = O(\frac{C}{k}\lambda_s(k)) = O(k^2 n^2 \lambda_s(k)).$$

We also need to consider the case where p and q come from the same set T_i. In this case the length of the k configuration increases by one, and crossings of leading endpoints correspond to four points from *three* sets becoming cocircular. It is easy to see that the number of distinct leading endpoints (and hence the number of topological events) on these bisectors b_{pq} is $O(k^3 n^2)$, and thus this case does not increase the above bound. (It is bounded by the total number of topological events in unions of three sets, which is at most $\binom{k}{3} O(n^2)$, by Claim 2.) ∎

This improves the bound of $O(n^3 k^4 \log^* n)$ given in [AIIT]. Moreover, if k is bounded by a constant then we get a bound of $O(n^2)$, whereas [AIIT] get $O(n^3)$. For constant k, this bound is tight, in the sense that a lower bound of $\Omega(n^2)$ can be demonstrated with just two rigidly moving point sets. See Figure 3: the vertically arranged pairs of points move rigidly to the right, so the Voronoi edge between each pair is destroyed and created n times. Summing over the n pairs gives the lower bound. Observe that we can slightly perturb the points in the figure so that they are in general position. Also, in our application, the point sets move in circular rather than linear trajectories; we note that the above construction can be modified without much difficulty to demonstrate the $\Omega(n^2)$ lower bound in this case as well.

4. Bounding the Changes to ∂Q

Now we return to the problem of bounding the number of edges that can appear on the boundaries of each $Q = \cup_j Q_j$. Each edge $e = (p, q, r, s, \theta)$ of ∂Q is a portion of an edge ϵ of some Q_j. (Recall, $q \in S_i$ is the common source of the Q_j's, p is the point in some S_j that induces ϵ, r and s induce the two endpoints of e on ∂Q, and θ is the orientation at which e first appeared on ∂Q.) In order to avoid confusion we reserve the term *edge* for e and call ϵ a Q_j-edge.

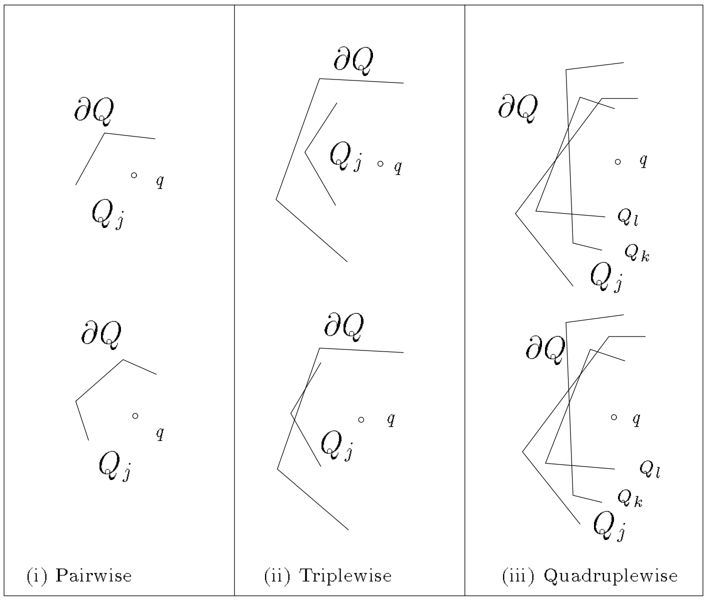

Figure 4: The types of changes on ∂Q

Clearly if a new Q_j-edge is created and is visible on ∂Q, then this is a new edge e on ∂Q. The appearance of any other edge e involves a portion of an already existing Q_j-edge becoming visible on ∂Q. Such a change must involve a Q_j-edge crossing a vertex of ∂Q. This vertex can either be convex (a vertex of another Q_k) or concave (the intersection of some Q_k and Q_l). Thus in the natural way we associate the appearance of each edge e on ∂Q with one of these three types of changes (see Figure 4 and also Figure 1):

(i) A new Q_j-edge appears and is visible. When this occurs, three new edges are created: the new edge corresponding to the Q_j-edge, and the neighboring edges whose endpoints have changed. (Analogously, when a visible Q_j-edge disappears two new edges are created.)

(ii) A convex vertex of Q_j pierces the boundary to become visible. When this occurs, four new edges are created: the two that induce the convex vertex, and the one edge on ∂Q that is now split into two. (When a convex vertex becomes invisible one new edge is created.)

(iii) A hidden crossing of a Q_j-edge and Q_k-edge becomes visible. When this occurs, two new edges are created: the two edges that intersect at the crossing. (When a hidden crossing becomes invisible three new edges are created.)

We consider each case in turn. Recall that the boundaries of a given cell Q_j are edges of $\mathrm{Vor}(S_i \cup S_j)$, and involve a given point $q \in S_i$. Thus by definition a change of type (i), where some Q_j-edge changes, involves a change in the dynamic Voronoi diagram $\mathrm{Vor}(S_i \cup S_j)$.

A change of type (ii), where some vertex of Q_j becomes visible on ∂Q or vice versa, involves a crossing of a vertex of Q_j with a Q_k-edge. The two Q_j-edges that meet at the given vertex are due to $q \in S_i$ and two points in S_j; call them p_1 and p_2. The Q_k-edge that is being crossed is due to q and to a point of S_k; call it p_3. Thus at the time that the vertex pierces the edge, the four points q, p_1, p_2, and p_3 become cocircular. This is a topological event in the dynamic Voronoi diagram $\mathrm{Vor}(S_i \cup S_j \cup S_k)$.

A change of type (iii), where a crossing of a Q_j-edge and a Q_k-edge becomes visible or vice versa, involves a crossing of three edges from three different cells. Each of the three edges is due to q and a point $p_1 \in S_j$, $p_2 \in S_k$, and $p_3 \in S_l$, respectively. Thus at the time that the three edges cross, the four points q, p_1, p_2, and p_3 become cocircular. This is a topological event in the dynamic Voronoi diagram $\mathrm{Vor}(S_i \cup S_j \cup S_k \cup S_l)$.

Hence, we can bound the number of edges that ever appear on ∂Q (summed over all sources q) by bounding the number of topological changes to the pairwise, triplewise and quadruplewise dynamic Voronoi diagrams. This brings us to:

Claim 3 *The number of edges that can appear on all the boundaries of the cells Q is $O(m^4n^2)$.*

Proof. In the previous section we showed that for a dynamic Voronoi diagram with a total of n points that move on some constant number of different trajectories, the total number of changes to the Voronoi diagram is $O(n^2)$. We have just shown an equivalence, up to a constant factor, between the appearance of an edge on ∂Q and a topological event in a pairwise, triplewise, or quadruplewise Voronoi diagram involving the sets S_i, $1 \le i \le m$. There are $\binom{m}{4}$ quadruplewise diagrams (which dominates the number of pairwise and triplewise diagrams) each for $O(n)$ points, which yields $O(m^4n^2)$. ■

Remarks. Note that not all the changes to the pairwise, triplewise and quadruplewise dynamic Voronoi diagrams get to be visible on the boundary of some ∂Q. In fact many of these changes may occur within the cells Q and not be seen on any of the boundaries. Note also that one topological change to any of the above dynamic Voronoi diagrams can affect up to four different boundaries ∂Q, since only up to four different points take part in the cocircularity event causing this topological change, hence only the boundaries of ∂Q for each one of these points as sources might be affected. In each of the topological changes, detailed in (i), (ii) and (iii) above, some edges of ∂Q are destroyed and some are created. In type (i) two ∂Q edges are destroyed, in type (ii) one edge is destroyed and in (iii) three edges are destroyed. Summing up the number of ∂Q changes for each type (adding/deleting ∂Q edges) we get up to five such updates for each type.

Our plan for computing the minimum value of $f_A(x, \theta)$ was first to bound the number of edges that can appear on the ∂Q's, which we have just done, and then to compute the minimum over each edge of each boundary. Thus we now turn to the problem of computing the minimum value of $f_A(x, \theta)$ over the domain specified by an edge $e = (p, q, r, s, \theta_0)$ of ∂Q over its epoch $[\theta_0, \theta_0']$ (recall that the epoch is the continuous interval over which e is visible on ∂Q). We show that this

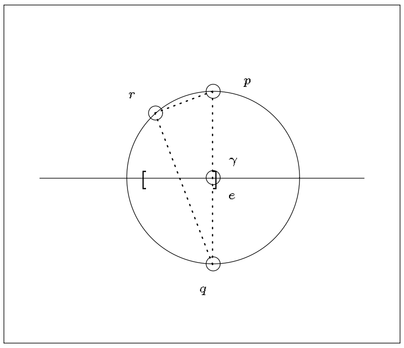

Figure 5: The minimum of $f_A(x,\theta)$ on e

minimum can be computed in constant time. Note that the points p, q, r and s are moving on circular trajectories; however as above we do not explicitly parameterize the points by θ.

The function $f_A(x,\theta)$ is the upper envelope of a set of functions, and from Fact 1 we know that $d_i(x,\theta)$ is the maximum of these functions along ∂Q (that is $f_A(x,\theta) = d_i(x,\theta)$ for values of (x,θ) along ∂Q). Thus we minimize the value of $d_i(x,\theta)$ over each edge e of ∂Q (where $d_i(x,\theta)$ corresponds to the set $S_{i,\theta}$, $q \in S_i$, as in Figure 1).

Along the entire bisector b_{pq} (rather than just the edge e), the minimum value of $d_i(x,\theta)$ for fixed θ is clearly obtained where b_{pq} intersects the segment pq. We call this location γ. The value of d_i at γ is simply one half the distance between the points p and q. For a fixed value of θ, if e contains γ, the minimum value of $d_i(x,\theta)$ on e is attained at γ. If e does not contain γ, then the minimum is attained at the endpoint of e closer to γ. We therefore identify the values of θ in the epoch of e when γ enters or leaves the edge e, and divide the epoch into the subintervals between these values. (See Figure 5, where the endpoints of e are denoted by $[,]$ and γ is denoted by a dot).

We observe that it leads to simpler expressions to consider the task of minimizing the square of d_i. Consider a subinterval in which e contains γ. Over this subinterval, the minimum value of $d_i(x,\theta)^2$ is $\min_t(1/2\|p-q\|)^2$. This is a fixed degree polynomial in $\sin\theta$ and $\cos\theta$, as is its derivative, so its minimum can be computed in constant time, by virtue of the following fact:

Fact 2 *A degree d polynomial in $\sin x$ and $\cos x$ has at most $4d$ roots $x \in [0, 2\pi)$.*

Proof. Let $f(x)$ be a degree d polynomial in $\sin x$ and $\cos x$. $f(x) = g(x) + h(x)$, where the terms of g contain only odd powers of $\sin x$ and the terms of h contain only even powers of $\sin x$. If $\hat{x} \in [0, 2\pi)$ is a zero of f, it is also a zero of $g(x)^2 - h(x)^2$, which can be written as a degree d polynomial in $\cos^2(x)$. Thus $\cos^2(\hat{x})$ is one of d values.

Since the equation $\cos^2(x) = y$ has at most 4 solutions over $[0, 2\pi)$, we have the stated bound on the number of roots of f over $[0, 2\pi)$. ∎

Over a subinterval in which e does not contain γ, we observe that e lies always to one side of γ or the other; assume that r is the point inducing the endpoint of e closer to γ throughout the subinterval. Then the minimum value of $d_i(x,\theta)^2$ over this subinterval is the minimum squared radius of the circle through p, q, r. This (and its derivative) can be represented as the quotient of two fixed degree polynomials in $\sin\theta$ and $\cos\theta$, so its minimum can be computed in constant time.

Finally, we observe that we have divided the epoch into only a constant number of subintervals. The end of a subinterval occurs when γ enters or leaves e; this will happen when the center of the circle through p, q, r (or p, q, s) is equal to γ. But when this happens, p, q, r lie on the vertices of a right triangle; i.e. $(r-p)(r-q) = 0$. Since this is a fixed degree polynomial in $\sin\theta$ and $\cos\theta$, there are only a constant number of subintervals. Thus, we are performing a constant amount of work over each of a constant number of subintervals, in order to compute the minimum value of $d_i(x,\theta)$ over the epoch of e in constant time.

5. The Algorithm

In this section we give the algorithm and describe the data structures for computing the directed Hausdorff distance $h(R_\theta A, B \oplus x)$. The algorithm consists of two stages.

The first stage is an initialization step, in which we compute the boundaries of the cells Q for the initial orientation $\theta = 0$ and sort the edges of each boundary ∂Q in a clockwise order. This part of the initialization was done in [HKS] and we use their algorithm for it. For each source point q we maintain the boundary ∂Q as a clockwise ordered list of edges. Each edge is labeled by a quintuple $(p, q, r, s, 0)$ as described in Section 4. It is clear that each boundary ∂Q is star-shaped with respect to its source q, at any given orientation θ (cf. [HKS]). Hence, maintaining the edges of ∂Q in a clockwise order, gives also an angularly sorted list of the vertices of ∂Q around q. This is crucial for the updating stage of the algorithm where we will need a fast (logarithmic time) insertion and deletion of edges of ∂Q. As θ varies, before a topological change occurs, the vertices and edges of ∂Q move in the plane but the angular order of the vertices of ∂Q, with respect to q, does not change. The only time that the order changes is at a topological change that affects ∂Q.

We also prepare at this stage the topological changes that occur in all of the pairwise, triplewise and quadruplewise dynamic Voronoi diagrams as θ varies from 0 to 2π. We sort these events according to increasing θ and maintain, for each topological change, the orientation θ at which the change occurs, the four sources that become cocircular at this event and the indices of the sets to which these sources belong. Let us denote the event by another quintuple $(p_1, p_2, p_3, p_4, \theta)$, where p_1, p_2 induce the Voronoi edge just before the cocircularity event (for $\theta' < \theta$ and close to θ), and p_3 and p_4 induce the Voronoi edge just after the cocircularity (for $\theta' > \theta$), in the Voronoi diagram of the sets to whom these points belong.

In the second stage we update the boundaries of the cells by considering each topological change in order of increasing θ. We do it in the following manner. Assume we have maintained all the boundaries of the cells Q, for each point q in each one of the sets S_i, ordered clockwise. Let the next topological change occur at θ and assume that it occurred due to the cocircularity of the sources p_1, p_2, p_3, p_4. We check whether there is an update to the boundary ∂Q of the cells Q corresponding to each of the four points p_1, p_2, p_3, p_4. Consider the boundary ∂Q of the cell Q around p_1. We compute the center c of the circle on which the four points are cocircular and locate the point on the boundary ∂Q. Locating the point on ∂Q can now be done fast; we compute the angle that the line p_1c makes with the x-axis and start a binary search on the list of vertices of ∂Q. For each vertex $v \in \partial Q$ which is encountered during the search we compute in $O(1)$ time the angle between the line p_1v and the x-axis, thus locating c on ∂Q in logarithmic time. (Note that if c was not found on ∂Q then it is a topological change that occurs within the cell Q and does not affect ∂Q, hence no update is required.) We perform the updates to ∂Q and proceed to computing the minimum value of $f_A(x, \theta)$ for the edges that were destroyed in this update. For each such edge $e = (p, q, r, s, \theta')$, we compute the minimum value of $f_A(x, \theta)$ over e over its epoch $[\theta', \theta]$, as described in the previous section. We keep track of the minimum such value over all edges that appear on the boundary. We outline the algorithm schematically:

Stage 1: Initialization.

1. Compute the boundary of $Q = \cup_j Q_j$ for each source q in each set S_i at $\theta = 0$, using the method of [HKS]. For each source, store the boundary as a clockwise ordered list of edges and vertices. Each edge is stored as the quintuple (p, q, r, s, θ), where $\theta = 0$ for these initial edges.

2. Compute the topological changes (cocircularity events) to all the pairwise, triplewise, and quadruplewise dynamic Voronoi diagrams. For each orientation θ at which there is a cocircularity event, store the ordered cocircularity quintuple $(p_1, p_2, p_3, p_4, \theta)$, and keep pointers to the sets S_i to which the points belong.

3. Sort all the cocircularity events by increasing θ.

Stage 2: Update and Minimization.

1. For each cocircularity event $(p_1, p_2, p_3, p_4, \theta)$ in the ordered list of cocircularity events, update the at most five edges on the boundary ∂Q of the cell around each of each of the points p_i, $i = 1, \ldots, 4$. For each destroyed edge, get the beginning of its epoch, θ', from the labeling of the edge, compute the minimum value of $f_A(x, \theta)$ over its epoch $[\theta', \theta]$, and keep track of the minimum such value.

2. For each remaining edge on ∂Q (i.e., that has not been destroyed at $\theta = 2\pi$), compute the minimum value of $f_A(x, \theta)$ over the θ range that begins at the beginning of the epoch of the edge and ends at $\theta = 2\pi$; again keeping track of the minimum such value.

Data structures.

The first data structure maintains ∂Q for each source point q at the current θ. For each source q we keep a list of the edges of ∂Q in a clockwise order. The second data structure is the corresponding list of all the vertices of ∂Q, which are angularly sorted around q. We described above that each edge is labeled by a quintuple which points to all of the sources that induce it. In the same manner each vertex of ∂Q is labeled by the triple of points that induces the vertex and keeps pointers to the two edges it is adjacent to. The third data structure is the list of cocircularity events, ordered by θ.

Time and Space Complexity Analysis.

Recall that we are computing the directed Hausdorff distance $h(R_\theta A, B \oplus x)$, where $|A| = m$ and $|B| = n$. The initialization stage is straightforward to analyze. In step 1 we use the method of [HKS] to compute the initial cell boundaries ∂Q of the cells around all the points q (for all the points $q \in S_i$, where $S_i = a_i \ominus B$, $i = 1, \ldots, m$) in time $O(m^2 n \alpha(mn) \log(mn))$.

Step 2 involves computing dynamic Voronoi diagrams, for all the pairs, triples and quadruples of the sets S_i. Each diagram can be computed in $O(n^2 \log n)$ time by using the method of [GMR], which computes the dynamic Voronoi diagram of k points in time $O(k \log k)$ plus $O(\log k)$ time per a topological change. The topological changes (cocircularity events) are computed on the fly. Since the number of source points for each of the pairwise, triplewise and quadruplewise dynamic Voronoi diagrams that we compute here is less than, or equal, $4n$, and we know from the Section 3 that the number of topological changes in each dynamic Voronoi diagram is $O(n^2)$, computing one dynamic Voronoi diagram and reporting all its topological changes takes $O(n^2 \log n)$ time.

The number of the dynamic Voronoi diagrams that we compute in step 2 is dominated by the number of quadruples of sets S_i (clearly this dominates the number of pairs and triples of sets). There are $\binom{m}{4}$ quadruples of sets, hence the overall time complexity of this step is $O(m^4 n^2 \log n)$.

Step 3 simply sorts the $O(m^4 n^2)$ cocircularity events, which requires $O(m^4 n^2 \log(mn))$ time.

In the update and minimization stage we consider each cocircularity event $(p_1, p_2, p_3, p_4, \theta)$. As we discussed above, for each event we compute in constant time the center c of the circle on which p_1, p_2, p_3 and p_4 are cocircular. We then find, in time logarithmic in n, the location of c on the boundary ∂Q of the cell around each point p_i ($i = 1, \ldots, 4$). Finally, we perform in constant time, up to five updates on each o the four boundaries ∂Q. Then for each edge that we remove rom ∂Q we must compute the minimum of $f_A(x, \theta)$ over that edge in constant time by the method of Section 4.

Hence the overall algorithm for computing the directed Hausdorff distance is dominated by the sorting in step 3 of the Initialization stage and is $O(m^4 n^2 \log(mn))$.

The storage requirements are as follows. For a fixed θ the total space needed to store the boundaries of ∂Q for all the mn points in $\cup S_i$ is $O(m^2 n \alpha(mn))$. Hence this is the storage requirement for the first and the second data structures. Trivially, the data structure for the topological changes needs $O(m^4 n^2)$ space, and this size dominates the space needed for the algorithm.

As noted above, to simplify the discussion, we have been considering the case of minimizing the directed distance as given by $f_A(x, \theta)$ in equation (6), rather than the Hausdorff distance as given by equation (7). The function f_A is the maximum of m functions d_i, each of which is in turn the minimum of n functions $\delta_{i,j}$. Similarly f_B is the maximum of n functions d'_j, each of which is the minimum of m functions $\delta_{i,j}$. Equation (7), which maximizes f_A and f_B, is then just the maximum of $m + n$ functions.

Therefore in order to modify the above algorithm to compute the *bidirectional* distance, we seek the maximum of $m+n$ functions rather than m functions. That is, we consider the m sets S_i each with n points, and the n sets S'_j each with m points. The cell $Q = \cup_j Q_j$ is then computed for each of the points in each S_i and for each of the points in each S'_j (rather than just for the points in each S_i). Thus all the steps of the algorithm remain the same, only now there are $O(m+n)$ sets each with $O(m+n)$ points, rather than m sets each with n points, yielding $O((m+n)^6)$ topological events (as opposed to $O(m^4n^2)$). Hence we get to the main result,

Theorem 2 *The minimum Hausdorff distance under Euclidean motion, $D(A,B) = \min_{x,\theta} H(r_\theta(A), B \oplus x)$, can be computed in time $O((m+n)^6 \log(mn))$, where the sets A and B contain m and n points respectively.*

6. Concluding remarks and open problems

We have shown that the minimum Hausdorff distance between point sets in the plane can be computed in time $O((m+n)^6 \log(mn))$ for sets with m and n points respectively. The algorithm that computes the distance is based on computing the topological changes in certain dynamic Voronoi diagrams. Thus in order to obtain our bounds for computing the Hausdorff distance we investigated the complexity of the dynamic Voronoi diagram of k sets of points in the plane, where each set consists of n points moving rigidly. We showed that the number of topological changes to such a diagram is $O(n^2k^2\lambda_s(k))$ for some fixed s, improving on the previous known bound of $O(n^3k^4 \log^* k)$ [AIIT]. This involved proving a Lemma that is interesting in itself, on the complexity of Davenport-Schinzel sequences in which the alphabet is divided into 'active' and 'inactive' subsets.

Our method for computing the minimum Hausdorff distance under rigid motion does not generalize easily to sets of *segments* in the plane. One interesting question is thus whether the parametric search methods of [AST] that were used for solving the minimum Hausdorff distance for segments under translation will be useful for solving this problem as well.

References

[AIIT] Aonuma, H., Imai, H., Imai, K., and Tokuyama, T., "Maximin location of convex objects in a polygon and related dynamic Voronoi diagrams", Sixth ACM Symposium on Computational Geometry, 1990, pp. 225–234.

[ASS] Agarwal, P.K., Sharir, M., and Shor, P., "Sharp upper and lower bounds for the length of general Davenport-Schinzel sequences", *J. Combinatorial Theory, Series A*, 52(1989), pp. 228–274.

[AST] Agarwal, P.K., Sharir, M., and Toledo, S., "Applications of parametric searching in geometric optimization", to appear in Third ACM-SIAM Symposium on Discrete Algorithms, 1992.

[ABB] Alt, H., Behrends, B. and Blomer, J., "Measuring the resemblance of polygonal shapes", manuscript.

[AMWW] Alt, H., Mehlhorn, K., Wagener, H., and Welzl, E., "Congruence, similarity, and symmetries of geometric objects", *Discrete and Computational Geometry*, 3(1988), pp. 237–256.

[ACHKM] 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.

[FL] Fu, J-J., and Lee, R.C.T., "Voronoi Diagrams of Moving Points in the Plane", *Int. Journal of Computational Geometry & Applications*, Vol 1(1), March 1991, pp. 23–32.

[GMR] Guibas, L.J., Mitchell, J.S.B., and Roos, T., "Voronoi diagrams of moving points in the plane", *17th International Workshop on Graph-Theoretic Concepts in Computer Science*, June 17-19, 1991, Richterheim/Fischbachau (Germany), Lecture Notes in Computer Science, Springer-Verlag.

[GS] Guibas, L.J.and Stolfi, J., "Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams", *ACM Transactions on Graphics*, Vol 4(2), April 1985, pp. 74 – 123.

[HS] Hart, S., and Sharir, M., "Non-linearity of Davenport-Schinzel sequences and of generalized path compression schemes", *Combinatorica* 6(1986) pp. 151–177.

[HK] Huttenlocher, D.P., and Kedem, K., "Efficiently computing the Hausdorff distance for point sets under translation", Sixth ACM Symposium on Computational Geometry, 1990, pp. 340–349.

[HKS] Huttenlocher, D.P., Kedem, K. and Sharir, M. "The upper envelope of Voronoi surfaces and its applications", *Proceedings of the Seventh ACM Symposium on Computational Geometry*, 1991, pp 194–203.

[Mu] Mumford, D., "The problem of robust shape descriptors", First Intl. Conf. on Computer Vision, IEEE Computer Soc. Press, 1987, pp. 602–606.