Submitted to the 1997 *ACM Symposium on Computational Geometry*

Maintaining the Extent of a Moving Point Set

Pankaj K. Agarwal[*] Duke University
Leonidas J. Guibas[†] Stanford University
John Hershberger Mentor Graphics Corp.
Eric Veach Stanford University

Abstract

Let S be a set of n moving points in the plane. We give new efficient and compact kinetic data structures for maintaining the diameter, width, and smallest area or perimeter bounding rectangle of the points. When the points in S move with pseudo-algebraic motions, these structures process $O(n^{2+\epsilon})$ events. We also give constructions showing that $\Omega(n^2)$ combinatorial changes are possible in these extent functions even when the points move on straight lines with constant velocities. We give a similar construction and upper bound for the convex hull, improving known results.

1 Introduction

Suppose S is a set of n moving points in the plane. In this paper we investigate how to maintain various descriptors of the *extent* of the point set, such as diameter, width, smallest enclosing rectangle, etc. These extent measures give an indication of how spread out the point set S is and are useful in various virtual reality applications such as clipping, collision checking, etc. As the points move continuously, the extent measure of interest (e.g., diameter) changes continuously as well, through its combinatorial realization (e.g., the pair of points defining the diameter) only changes at certain discrete times. Our approach is to focus on these discrete changes or events and track through time the combinatorial description of the extent measure of interest.

We do so within the framework of *kinetic data structures* (KDSs for short), as developed by Basch, Guibas, and Hershberger [3] and further elaborated in Section 2. There are two notable and novel aspects of that framework. Firstly, while extensive work has been done on dynamic data structures in computational geometry [4], this is all focused on handling insertions/deletions of objects and not continuous change. Kinetic data structures by contrast gain their efficiency by exploiting the continuity or coherence in the way the system state changes. Secondly, unlike Atallah's dynamic computational geometry framework [2], which was introduced to estimate the maximum number of combinatorial changes in a geometric configuration under predetermined motions in a certain class, the KDS framework is fully *on-line* and allows each object to change its motion at will, due to interactions with other moving objects, the environment, etc.

Section 3 presents new kinetic algorithms for diameter, width, and smallest enclosing rectangle in both the area and perimeter senses. If we assume that the points of S follow pseudo-algebraic motions (defined below), then the number of events processed by each of our algorithms is $O(n^{2+\epsilon})$

[*]Support was provided by National Science Foundation research grant CCR–93–01259, by Army Research Office MURI grant DAAH04–96–1–0013, by a Sloan fellowship, by a National Science Foundation NYI award and matching funds from Xerox Corp, and by a grant from the U.S.-Israeli Binational Science Foundation.

[†]Support was provided in part by National Science Foundation grant CCR–9623851 and by US Army MURI grant 5–23542–A.

(for all $\epsilon > 0$). In particular these bounds prove that none of the extent measures mentioned can change combinatorially more than $O(n^{2+\epsilon})$ times. A quadratic bound is natural for diameter, as it is defined by two points of the set S, but it is somewhat surprising for the other measures, as width is defined by three points, and the minimum bounding rectangles by four or five of the points. The data structures we give are efficient and compact in the KDS sense, though not local.

Section 4 is devoted to giving lower bound constructions for these extent measures under *linear* point motions: we show that diameter, width, and the two flavors of smallest bounding rectangle can all change $\Omega(n^2)$ times as the n points of S move on straight line trajectories with constant velocities (possibly different for each point). Such lower bound constructions are much easier if we allow quadratic or other higher degree motions—the fact that the same bounds hold with linear motions is quite interesting. Our constructions employ a key component consisting of cocircular (or nearly cocircular) points that move on straight lines while maintaining their (near-) cocircularity. Finally in Section 5 we give a similar construction showing that the convex hull of n points moving linearly in the plane can also change $\Omega(n^2)$ times. We also prove a tighter upper bound than was previously known for the number of combinatorial changes to the convex hull. This bound is $O(n\lambda_s(n))$, where $\lambda_s(n)$ is the length of a Davenport-Schinzel sequence [6], and the parameter s bounds the number of times three points can become collinear. The bound specializes to $O(n^2)$ for linearly moving points—which is therefore tight.

2 Kinetic data structure preliminaries

A kinetic data structure maintains a *configuration function* of continuously moving data (e.g., diameter, width, etc., of moving points). It does so by maintaining a set of *certificates* that jointly imply the correctness of the computed configuration function. Each certificate is a geometric predicate on a constant number of data elements, such as, for example, "points A and B are farther apart than points C and D." The certificates are typically derived from a static algorithm for computing the configuration function. For example, the certificates for maintaining the diameter might include a set of distance comparisons establishing a partial order on the relevant pairwise distances, with a single maximum element.

The certificates are stored in a priority queue, ordered by the next time at which a certificate will be violated. Each data element has a *flight plan* that gives full or partial information about the current motion of the element, and these flight plans are used to compute the next violation time for each certificate. When the next violation time is reached, the algorithm removes the violated certificate from the queue and computes certificates for the new data configuration. Some number of certificates may have to be removed from the queue, and some number of new certificates added.

When a data element changes its flight plan, all the certificates in the priority queue that depend on it must have their times of next violation recomputed, and their positions in the queue must be updated. A KDS is called *local* if the number of certificates that depend on a single data element is polylogarithmic in the total number of data elements.

The violation of a certificate is called an *event*. *External events* cause the configuration function to change. *Internal events* do not affect the configuration function, but must be processed for the integrity of the data structure. We evaluate a KDS by counting events under the assumption that the data motions are *pseudo-algebraic*, i.e., each certificate predicate changes sign a bounded number of times when applied to any fixed subset of data elements. A KDS is called *efficient* if the worst-case number of total events (internal plus external) is asymptotically the same as, or only slightly larger than, the worst-case number of external events, under the assumption of pseudo-algebraic motion.

A KDS is called *compact* if the number of certificates stored in the priority queue is roughly linear in the number of data elements.

Efficient, local, and compact kinetic data structures are known for maintaining the convex hull and closest pair of points moving in the plane, and for computing the maximum of points moving along a line [3]. The data structure for computing the maximum is known as a *kinetic tournament*.

3 Algorithms

In this section we present kinetic data structures for maintaining three different versions of the extent of a planar point set: diameter, width, and minimum enclosing box. Each of these data structures is based on the kinetic data structure of Basch, Guibas, and Hershberger for maintaining the convex hull of a point set in motion [3]. On top of that data structure we build a kinetization of the rotating calipers algorithm [5, 7], specialized to the desired version of the extent.

3.1 Diameter

The diameter of a point set is the maximum pairwise separation of two points in the set. It is realized by a pair of antipodal vertices of the convex hull. (By *antipodal* points we mean two points on opposite sides of the hull whose supporting lines are parallel.) A standard way to compute the diameter of a static point set is to compute the convex hull, find all pairs of antipodal vertices by a linear scan around the hull boundary, and then identify the pair with maximum separation [5]. In this section we show how to kinetize this algorithm.

Before we proceed, let us dualize the problem, because the computation of antipodal points is easier to describe in the dual setting. In the dual, each point (p, q) of the point set maps to the line $y = px + q$. The upper convex hull of the point set dualizes to the upper envelope of the set of dual lines. Likewise the lower convex hull dualizes to the lower envelope. Each convex hull vertex dualizes to a segment of the envelope, and the range of slopes of the vertex's supporting lines dualizes to the x-interval spanned by the segment.

If we consider the dual envelopes as x-ordered lists of intervals, such that each interval represents a convex hull vertex and its range of supporting slopes, then we can find all antipodal pairs simply by merging the lists for upper and lower dual envelopes in x-order. Any two convex hull vertices whose intervals overlap have a common supporting slope, i.e., they are antipodal.

To kinetize this static algorithm for computing the diameter, we use the merged x-order of the two envelope lists as certificates to guarantee the correctness of the current set of antipodal pairs. We store the merged list in a balanced binary tree so that updates to the list can be performed in time $O(\log n)$ plus time proportional to the number of antipodal pairs affected. When a certificate is violated, it means that two interval endpoints have exchanged places. A constant number of certificates involving those intervals need to be updated to restore the certificates to correctness. When the underlying convex hull changes combinatorially, intervals may be added to or deleted from one of the envelope lists, and we search the binary tree to find the location to modify in the merged list.

Theorem 3.1 *The data structure for maintaining the diameter is compact and efficient.*

Proof: The underlying convex hull data structure and the kinetic tournament are both compact and efficient. It is also easy to see that the merged list structure is compact: its size is $O(n)$.

To prove efficiency, we must bound the number of events in the merged list structure, under the assumption that the points move with pseudo-algebraic motion. The key quantity to bound is the number of pairs of points that become antipodal over the life of the algorithm, since the list changes only when antipodal pairs change.

We extend the 2-D upper envelope structure into 3-D by considering time as a static third dimension. Each line (dual to a point of the set) becomes a pseudo-algebraic surface when the third dimension is added. The upper envelope at any point in time is given by a 2-D slice through the 3-D upper envelope of surfaces. Because the convex hull of n points in pseudo-algebraic motion changes $O(n^{2+\epsilon})$ times, for any $\epsilon > 0$ [2], the upper envelope of surfaces has the same complexity.

At any instant in time, the antipodal pairs of the hull are determined by the overlay of two 2-D envelopes. Each pairwise overlap between the projections of two envelope edges, one from the upper envelope and one from the lower, corresponds to an antipodal pair of points. When we add time as a third dimension, we see that the total number of antipodal pairs created is equal to the number of envelope surface patches whose projections into the xt-plane overlap. This quantity is bounded by $O(n^{2+\epsilon})$ [1].

The separation of each antipodal pair is a pseudo-algebraic function of time, and hence the upper envelope of these functions has $O(n^{2+\epsilon})$ complexity; the kinetic tournament computes this upper envelope (the diameter) within essentially the same time bound. Theorem 4.1 of Section 4.1 shows that the total number of different diametral pairs is $\Omega(n^2)$, and hence our kinetic data structure is efficient. ∎

Note that the data structure is *not* local: one point may belong to $O(n)$ antipodal pairs. It may be possible to achieve locality by making the pairing relationship more sophisticated, but this would require some additional insight.

3.2 Width

The width of a point set is the minimum separation of two parallel lines that sandwich the point set between them. It is well known that one of the lines contains an edge of the convex hull and the other passes through a hull vertex.

In the dual, a convex hull edge maps to a vertex of the upper (or lower) envelope. An antipodal edge-vertex pair, therefore, corresponds to a vertex on the upper or lower envelope whose x-coordinate lies in the x-interval of a segment on the other envelope. To find all antipodal edge-vertex pairs, we merge the x-ordered intervals of the two envelopes, just as in the diameter algorithm, except we note overlapping interval-vertex pairs, instead of interval-interval pairs. To compute the width we simply find the minimum separation among all antipodal edge-vertex pairs.

The kinetic data structure for computing the width is almost identical to the one for the diameter. The only difference is that the basic antipodal pairs are edge-vertex pairs (both primally and dually), and the kinetic tournament computes the minimum separation, rather than the maximum.

Theorem 3.2 *The data structure for maintaining the width is compact and efficient.*

The similarity of the diameter and width data structures masks a rather surprising difference. We expect the number of combinatorial changes to the diameter to be $O(n^{2+\epsilon})$, because there are only $O(n^2)$ pairs of points. However, the width is determined by *triples* of points—two edge endpoints and an opposing vertex—and so the natural bound on the number of combinatorial changes to the width is $O(n^{3+\epsilon})$. However, our algorithm shows that the actual number of changes is only $O(n^{2+\epsilon})$, an order of magnitude smaller.

3.3 Minimum boxes

A common way of reducing the complexity of spatial algorithms is to approximate a complex geometric structure by a rectilinear box. Queries (e.g., intersection tests) are first performed on the box, then on the actual structure only if the approximate test shows it to be necessary. In this way many queries on the complex structure may be avoided. The box approximation is often chosen to be axis-aligned, but in situations in which a better approximation is desired, an arbitrarily oriented box may be computed.

We can maintain a box of minimum area or minimum perimeter with a single technique. The basic idea is the same as in previous sections. However, to maintain boxes, we need not just antipodal points, but sets of four points—two antipodal pairs with perpendicular supporting lines.

In the dual setting in which points map to lines, we compute four envelopes—upper, lower, left, and right. We merge the four envelope lists into one list. An interval in the merged list corresponds to a slope range in which the four convex hull vertices supported by lines parallel and perpendicular to the slope are constant. For this slope range and set of four vertices, the minimum area/perimeter rectangle is trivially computed. By minimizing over all intervals in the merged list, we find the global minimum rectangle.

The kinetic data structure for computing minimum boxes is essentially similar to those described above. The difference is that instead of maintaining the merge of two envelope lists, we maintain the merge of four lists. For each interval in the merged list we maintain the minimum rectangle enclosing the four extreme points, subject to the condition that the slope of one side of the rectangle must lie inside the range given by the interval. A kinetic tournament on the rectangles selects the smallest one.

Theorem 3.3 *The data structure for minimum boxes is compact and efficient.*

The preceding theorem shows that the number of combinatorially different minimum boxes for points in pseudo-algebraic motion is $O(n^{2+\epsilon})$.

4 Lower bounds with linear motion

In this section we give a collection of lower bounds on the number of combinatorial changes to the extent of a point set when each point moves linearly. Each of our constructions uses cocircular points whose linear motion maintains cocircularity. Note, however, that the lower bounds hold even if we perturb the points slightly to place them in general position.

Let $\bar{c}(\theta) = (\cos\theta, \sin\theta)$ be the point on the unit circle at angle θ from the origin. Suppose that a point p moves linearly along a chord of the unit circle:

$$p(t) = (1-t)\,\bar{c}(\alpha) + t\,\bar{c}(\alpha+\phi).$$

Then the position of $p(t)$ can be expressed in polar coordinates, $p(t) = (r_p(t), \theta_p(t))$, in terms of $\theta(t) = \tan^{-1}((2t-1)\tan\frac{\phi}{2})$, which varies in the range $[-\phi/2, \phi/2]$ as t varies in $[0, 1]$.

$$\begin{aligned} r_p(t) &= \frac{\cos\frac{\phi}{2}}{\cos\theta(t)} \\ \theta_p(t) &= \alpha + \frac{\phi}{2} + \theta(t) \end{aligned}$$

Note that the initial position $p(0)$ does not appear in the expression for $r_p(t)$. If multiple points start on the unit circle, then move at the same rate along chords of the same length, the points will remain cocircular through the whole motion. If all the motions are clockwise (or all counterclockwise), then the angular separation of each pair of points is constant: $\theta_p(t) - \theta_q(t)$ is just the difference of the initial angular positions of p and q. See Figure 1.

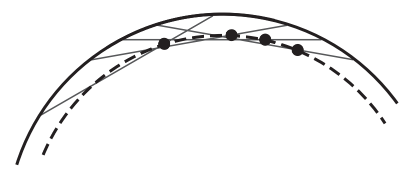

Figure 1: Cocircularity is preserved by linear motion along equal-length chords

4.1 Diameter

In this section we give an $\Omega(n^2)$ lower bound on the number of distinct diametral pairs that can appear in a set of n points moving linearly. We first discuss diametral pairs for points lying on two concentric circles, then specify a particular set of linearly moving points, and finally argue that our set has $\Omega(n^2)$ diametral pairs over time.

Consider points on two concentric circles C_1 and C_2 with radii r_1 and r_2. Suppose that the points on C_i lie on an arc of length at most $\pi r_i/4$, for each $i \in \{1, 2\}$. If a point on C_1 and another on C_2 are collinear with the circles' common center and on opposite sides of it, then their separation is $r_1 + r_2$, and this is the diameter of the set. If there is only one such pair, it is the unique diametral pair.

We define a point set with m stationary points and m moving points, for a total of $n = 2m$ points. The set $P = \{p_0, \ldots, p_{m-1}\}$ of stationary points is defined by

$$p_i = \bar{c}\left(\frac{\pi}{8m^2} i\right).$$

The moving points are $Q = \{q_0, \ldots, q_{m-1}\}$. They move linearly along chords of the unit circle:

$$q_j(t) = (1-t)\,\bar{c}\left(\frac{7}{8}\pi + \frac{\pi}{8m} j\right) + t\,\bar{c}\left(\frac{9}{8}\pi + \frac{\pi}{8m} j\right).$$

Theorem 4.1 *The diameter of the set of n linearly moving points described above is defined by $\Omega(n^2)$ different pairs of points during the time interval $t \in [0, 1]$.*

Proof: As noted above, the points of Q lie on a common circle whose radius varies with time. The angular position of $q_j(t)$ is $\theta_j(t) = \pi + \frac{\pi}{8m} j + \theta(t)$, for $-\frac{\pi}{8} \le \theta(t) \le \frac{\pi}{8}$. Thus Q lies in a constant-size angular range $\theta_{m-1}(t) - \theta_0(t) = \frac{\pi(m-1)}{8m} < \frac{\pi}{8}$. The angular range of P is also less than $\frac{\pi}{8}$.

Points p_i, $q_j(t)$, and the origin are collinear iff

$$\frac{\pi}{8m^2} i + \pi = \theta_j(t) = \pi + \frac{\pi}{8m} j + \theta(t),$$

that is, iff

$$\theta(t) = \frac{\pi}{8m}\left(\frac{i}{m} - j\right).$$

Each (i, j) pair determines a unique value of $\theta(t)$ in the interval $[-\frac{\pi}{8}, \frac{\pi}{8}]$, which corresponds to a unique value of $t \in [0, 1]$; call this value t_{ij}. Thus there are $m^2 = \Theta(n^2)$ distinct values $t_{ij} \in [0, 1]$ such that $(p_i, q_j(t_{ij}))$ is the unique diametral pair of the point set at time t_{ij}. ▮

4.2 Width

In this section we give an $\Omega(n^2)$ lower bound on the number of distinct vertex triples that determine the width of a set of n linearly moving points. The construction uses two sets of points, one stationary and one moving, each set cocircular.

Let us define the *slab* of a line segment s to be the set of all points that project perpendicularly onto s. The basis of our construction is the following observation:

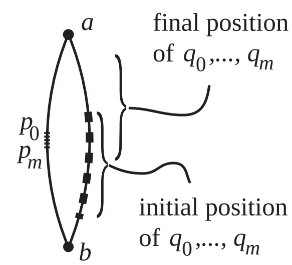

Figure 2: The width lower bound construction

Observation 4.2 *Suppose that the width of a point set is determined by a convex hull edge e and a hull vertex v. Then v lies in the slab of e.*

The stationary points of our set are

$$\begin{aligned} a &= \bar{c}\left(\frac{\pi}{8}\right) \\ b &= \bar{c}\left(-\frac{\pi}{8}\right) \\ p_i &= \bar{c}\left(-\pi + \frac{\pi(i - m/2)}{64m^2}\right) + (2\cos\frac{\pi}{8}, 0), \quad \text{for } i \in \{0, \ldots, m\}. \end{aligned}$$

The moving points are

$$q_j(t) = (1-t)\,\bar{c}\left(-\frac{3\pi}{32} + \frac{\pi}{8m}j\right) + t\,\bar{c}\left(-\frac{\pi}{32} + \frac{\pi}{8m}j\right), \quad \text{for } j \in \{0, \ldots, m\}.$$

See Figure 2. The total number of points is $n = 2m + 4$. These points lie on two unit circles that intersect in arcs of length $\pi/4$. The intersection points are a and b; the p_i lie in a tight clump at the center of the left arc; the q_j lie in a $\pi/8$ sector of the right arc.

Theorem 4.3 *The width of the set of n linearly moving points described above is defined by $\Omega(n^2)$ different triples of points during the time interval $t \in [0, 1]$.*

Proof: As in the previous subsection, $q_j(t)$ can be expressed in polar form as

$$\begin{aligned} r_j(t) &= \frac{\cos\frac{\pi}{32}}{\cos\theta(t)} \\ \theta_j(t) &= -\frac{\pi}{16} + \frac{\pi}{8m}j + \theta(t) \end{aligned}$$

for $\theta(t) \in [-\frac{\pi}{32}, \frac{\pi}{32}]$.

In their initial positions, all the points appear on the convex hull, in the counterclockwise order $a, p_0, \ldots, p_m, b, q_0, \ldots, q_m$. In fact, we show in the full paper that all the points appear on the convex hull for all $t \in [0,1]$, in the same order.

For each j such that $\lceil m/4 \rceil \leq j < \lfloor 3m/4 \rfloor$, the convex hull edge (q_j, q_{j+1}) intersects the line $y = 0$ during the angular interval

$$\frac{\pi}{16} - \frac{\pi}{8m}\left(j + \frac{3}{4}\right) \leq \theta(t) \leq \frac{\pi}{16} - \frac{\pi}{8m}\left(j + \frac{1}{4}\right). \tag{1}$$

(The restriction on j ensures that this is a valid interval of $\theta(t)$.) In the full paper, we show that during this $\theta(t)$ interval, only the edge (q_j, q_{j+1}) that intersects $y = 0$ satisfies Observation 4.2, and so only it can determine the width. As $\theta(t)$ varies in the range given by (1), each point p_i becomes antipodal to (q_j, q_{j+1}) in turn, hence determining the width.

We have exhibited roughly $m/2$ convex hull edges, each of which in its turn determines the width with $m+1$ different hull vertices. Thus the triple of hull vertices determining the width changes $\Omega(n^2)$ times. ∎

4.3 Minimum boxes

This section exhibits a configuration of n points in linear motion such that the minimum-area (or minimum-perimeter) enclosing rectangle undergoes $\Omega(n^2)$ combinatorial changes.

The construction involves $n/2$ closely spaced points $p_1, \ldots, p_{n/2}$ that always lie on a circular arc of large radius, rotating counterclockwise around the origin. There are an additional $n/2$ points $q_1, \ldots, q_{n/2}$ near the origin, whose convex hull forms a sequence of squares. In particular, at integer times $t = j$ (for $j = 1, \ldots, n/8$), the convex hull of the q_i will be a square Q_j defined by the four points $q_{4j-3}, \ldots, q_{4j}$. All squares have a side length between 2 and 3, although each Q_j is slightly bigger than Q_{j-1}. The squares also have different orientations: the base of Q_j makes an angle of $j\theta$ with respect to the x-axis (where θ is a function of n). See Figure 3.

The idea is that at time $t = j$, the p_i will be just below the line L_j through the origin with angle $j\theta$. We will show that the bounding box B has sides parallel to Q_j, and thus its combinatorial description depends on which of the p_i is farthest from the origin in the direction of L_j. Each of the p_i will become the farthest point in turn, as the points rotate through L_j, thereby producing $n/2$ combinatorial changes to the bounding box. This is repeated at times $t = 1, \ldots, n/8$, yielding $\Theta(n^2)$ changes in total.

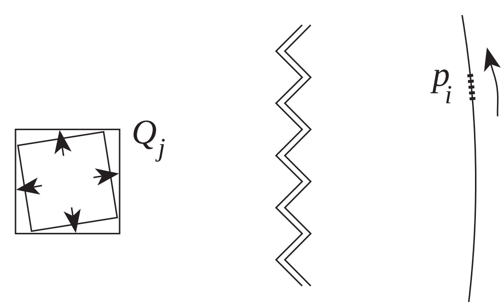

Figure 3: The lower bound for minimum boxes

The following arguments make this construction more precise.

Lemma 4.4 *Let $p_1, \ldots, p_n$ be points on a circle with radius $r \gg 1$, centered at the origin. Assume that these points satisfy $p_{i,x} > 0$ and $p_{i,y} \in [-1,1]$ for all i, i.e., they lie on a short arc near the positive x-axis. Also let Q be the square whose vertices $q_1, \ldots, q_4$ are the points $(\pm 1, \pm 1)$. Then the minimum-area (or minimum-perimeter) bounding box that encloses the p_i and Q is given by $B = [-1, x_{\max}] \times [-1, 1]$, where $x_{\max} = \max_i p_{i,x}$.*

The construction now proceeds as follows. For each $i = 1, \ldots, n/8$, the vertices of Q_i are obtained by taking the square Q whose vertices are $(\pm 1, \pm 1)$, and rotating it by an angle of $i\theta$ counterclockwise around the origin. The size of Q_i varies with time, according to the scale factor

$$s_i(t) = 1 + 4\theta(2it - i^2).$$

In other words, each vertex of Q_i is on a linear trajectory through the origin, such that its distance from the origin at time t is $\sqrt{2}s_i(t)$. We let $\theta = 8/n^2$, which ensures that all $s_i(t)$ lie in the range $[1/2, 3/2]$ for $0 \leq t \leq n/8$.

Note that Q_j is clearly the largest square at time $t = j$, since we can rewrite $s_i(t)$ as

$$s_i(t) = 1 + 4\theta[t^2 - (i - t)^2].$$

The following lemma is a slightly stronger version of this.

Lemma 4.5 *When $j - 1/4 \leq t \leq j + 1/4$, the square Q_j contains all other squares Q_i.*

Finally, the points $p_1, \ldots, p_n$ are placed on a circle C of radius $r = 4/\theta$, equally spaced along an arc of length 1/2. At time $t = 0$, they all have y-coordinates in the range $[-1/4, 1/4]$. All points move counterclockwise along chords that subtend an angle of $\theta^* = (n/8)\theta$, such that they intersect C at times $t = 0$ and $t = n/8$. The approximate chord length is $r\theta^* = n/2$, so that the speed of each p_i is approximately 4, and the y-coordinates of the points lie in the range $[4t - 1/4, 4t + 1/4]$ (noting that the chords are all nearly vertical for large n).

Now, for each $j = 1, \ldots, n/8$, consider the points near time $t = j$. The bounding box is determined by Q_j and the point p_i that is farthest from the origin along the line L_j that makes an angle of $j\theta$ with the x-axis. Now, at time $t = j - 1/8$, all p_i have y-coordinates in the range $[4j - 3/4, 4j - 1/4]$, and lie below the intersection of L_j with C (at $y \approx 4j$). As each p_i crosses L_j, it is clearly the farthest point in direction L_j. By time $t = j + 1/8$, all the p_i have crossed, and we are done.

We have established the following theorem.

Theorem 4.6 *The combinatorial description of the minimum-area (or minimum-perimeter) bounding box of n points moving linearly in the plane can change $\Omega(n^2)$ times.*

5 Tight bounds for kinetic convex hulls

In this section we give tight bounds on the number of combinatorial changes that may occur in the convex hull of points moving linearly in the plane. The lower bound construction is an easy application of the linear-motion-on-circles technique of Section4. The upper bound is an improvement on the known bounds for points in general pseudo-algebraic motion; when specialized to the case of linear motion, it shows that the convex hull may undergo $\Theta(n^2)$ combinatorial changes.

5.1 Lower bound

We exhibit a configuration of $2n$ points in linear motion for which the convex hull undergoes $\Omega(n^2)$ combinatorial changes. This improves the lower bound example given by Sharir and Agarwal [6], which uses quadratic motions.

We define two convoys of oppositely moving points. The points always lie on a common circle (which varies in size), so all are on the convex hull, but their order along the circle changes.

Let

$$\begin{aligned} p_i(t) &= (1-t)\,\bar{c}\left(\frac{\pi}{4n}i\right) + t\,\bar{c}\left(\frac{\pi}{4}+\frac{\pi}{4n}i\right) \\ q_j(t) &= (1-t)\,\bar{c}\left(\frac{\pi}{4}+\frac{\pi}{8n^2}j\right) + t\,\bar{c}\left(\frac{\pi}{8n^2}j\right), \end{aligned}$$

for $i, j \in \{1, \ldots, n\}$. At any time $t \in [0,1]$, all the p_i and q_j lie on a common circle with radius $r(t) = \cos\frac{\pi}{8}/\cos\theta(t)$, for $\theta(t) = \tan^{-1}((2t-1)\tan\frac{\pi}{8})$. The angular position of $p_i(t)$ is $\theta(p_i,t) = \pi i/4n + \pi/8 + \theta(t)$ and the angular position of $q_j(t)$ is $\theta(q_j,t) = \pi j/8n^2 + \pi/8 - \theta(t)$. Point p_i coincides with q_j at

$$\theta(t) = \frac{\pi}{8n}\left(\frac{j}{2n} - i\right).$$

Thus each (i,j) pair determines a unique $\theta(t) \in [-\frac{\pi}{8}, 0]$ at which p_i and q_j exchange on the convex hull. We have established the following theorem.

Theorem 5.1 *There is a set of n linearly moving points whose convex hull undergoes $\Omega(n^2)$ combinatorial changes as the points move.*

5.2 Upper bound

We bound the number of combinatorial changes to the convex hull in terms of the number of times any three points become collinear. It is well known that if the point trajectories are algebraic of degree k, then three points become collinear at most $s = 2k$ times. The theorem below shows that in this case there are $O(n\lambda_{2k}(n))$ changes to the convex hull. This improves the bound of $O(n\lambda_{2k+2}(n))$ given in [6]. In particular, it implies that for linear motion the number of changes is $O(n^2)$, matching the lower bound of the preceding section.

Theorem 5.2 *Given n points moving in the plane such that no three points become collinear more than s times, the combinatorial description of their convex hull changes at most $O(n\lambda_s(n))$ times.*

Proof: Let the points be identified by integers, $P = \{1, \ldots, n\}$, and define the *left-neighbor* function $l_i(t)$ as follows. If i does not belong to the convex hull at time t, then $l_i(t) = \epsilon$. Otherwise, $l_i(t)$ is the point j on the convex hull that is adjacent to i in the counterclockwise direction.

For each i, let L_i be the sequence of values assumed by $l_i(t)$ as t ranges from $-\infty$ to ∞. We remove all occurrences of ϵ from L_i, and replace any strings of identical symbols by a single occurrence, to yield a reduced sequence L_i^*.

In the full paper, we show that $\sum |L_i^*|$ is an upper bound on the number of changes to the convex hull (where $|S|$ denotes the length of a sequence S). Furthermore, each L_i^* is a $(n-1, s)$ Davenport-Schinzel sequence. If the alternation $j \ldots k$ appears in L_i^*, then the signed area of triangle ijk is zero at some intermediate time, implying a collinearity of i, j, and k. Given that any three points are collinear at most s times, there are at most s alternations between any two symbols j and k. Thus, each L_i^* is a $(n-1, s)$ Davenport-Schinzel sequence, and we have $\sum |L_i^*| \leq n\lambda_s(n)$. ∎

References

[1] P. Agarwal, O. Schwarzkopf, and M. Sharir. The overlay of lower envelopes and its applications. *Discr. Comput. Geom.*, 15:1–13, 1996.

[2] M. J. Atallah. Some dynamic computational geometry problems. *Comput. Math. Appl.*, 11:1171–1181, 1985.

[3] J. Basch, L. Guibas, and J. Hershberger. Data structures for mobile data. In *Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, 1997.

[4] Y.-J. Chiang and R. Tamassia. Dynamic algorithms in computational geometry. *Proc. IEEE*, 80(9):1412–1434, September 1992.

[5] F. P. Preparata and M. I. Shamos. *Computational Geometry.* Springer-Verlag, New York, 1985.

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

[7] G. Toussaint. Solving geometric problems with the “rotating calipers”. In *Proceedings of IEEE MELECON ’83*, 1983.