Maintaining Approximate Extent Measures of Moving Points*

Pankaj K. Agarwal[†] Sariel Har-Peled[‡]

26/05/2001 15:49

Abstract

We present approximation algorithms for maintaining various descriptors of the extent of moving points in $\mathbb{R}^d$. We first describe a data structure for maintaining the smallest orthogonal rectangle containing the point set. We then use this data structure to maintain the approximate diameter, and smallest enclosing disk of a set of moving points in $\mathbb{R}^2$ so that the number of events is only a constant. This contrasts with $\Omega(n^2)$ events that data structures for the maintenance of those exact properties have to handle.

1 Introduction

With the rapid advances in positioning systems, e.g., GPS, ad-hoc networks, and wireless communication, it is becoming increasingly feasible to track and record the changing position of continuously moving objects. These developments have raised a wide range of challenging geometric problems involving moving objects, including efficient data structures for answering proximity queries, for clustering, and for maintaining connectivity information. Many of these problems ask for maintaining a descriptor of the *extent* of a point set. For example, in R-trees, one of the most widely used spatial data structures in practice, each node is associated with a subset of points and the smallest orthogonal rectangle containing this subset. If the points are moving, then one has to maintain the smallest enclosing rectangle as the points move continuously. Other applications of maintaining an extent include collision detection, clustering, animation, and physical simulation.

*Work by P.A. was supported by Army Research Office MURI grant DAAH04-96-1-0013, by a Sloan fellowship, by NSF grants EIA–9870724 and CCR–9732787 and by a grant from the U.S.-Israeli Binational Science Foundation. Work by S.H.-P. was supported by Army Research Office MURI grant DAAH04-96-1-0013. Part of this work was done while he was at Duke University.

[†]Center for Geometric Computing, Department of Computer Science, Box 90129, Duke University, Durham, NC 27708-0129, USA. E-mail: `pankaj@cs.duke.edu`

[‡]Department of Computer Science, DCL 2111; University of Illinois; 1304 West Springfield Ave.; Urbana, IL 61801; USA; `sariel@uiuc.edu`; `http://www.uiuc.edu/~sariel/`

1.1 Problem statement

Let $P = \{p_1, \ldots, p_n\}$ be a set of n points moving in $\mathbb{R}^d$. For a given time t, let $p_i(t) = (x_i^1(t), \ldots, x_i^d(t))$ denote the position of p_i at time t. We will use $P(t)$ denote the set P at time t. We say that the motion of P has *degree* k if every $x_i^j(t)$ is a polynomial of degree at most k. We call a motion of degree 1 *linear*. In this case each point of P moves along a straight line with fixed speed. We say that the motion of P is *algebraic* if it is of degree d for some constant $k \geq 0$. For most of the discussion in this paper, we will assume that the motion of P is linear, i.e., $p_i(t) = \mathbf{a}_i + \mathbf{b}_i t$, where $\mathbf{a}_i, \mathbf{b}_i \in \mathbb{R}^d$. The values $\mathbf{a}_i, \mathbf{b}_i$ may change over time. We assume that the objects are responsible for updating the values of $\mathbf{a}_i, \mathbf{b}_i$.

In this paper we develop efficient approaches for maintaining various descriptors of the extent of P, including the smallest enclosing orthogonal rectangle, diameter, and smallest enclosing disk. These extent measures indicate how spread out the point set P is. As the points move continuously, the extent measure of interest changes continuously as well, though its combinatorial realization changes only at certain discrete times. For example, the smallest orthogonal rectangle containing P can be represented by a sequence of $2d$ points, each lying on one of the facets of the rectangle. As the points move, the rectangle also changes continuously. At certain discrete times, the points lying on the boundary of the rectangle change, and we have to update the sequence of points defining the rectangle. Similarly, in the case of diameter, the pair of points defining the diameter 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.

It turns out that maintaining the exact description of extent measures is quite expensive. For example, a result of Agarwal *et al.* [AGHV97] shows that the diameter of a point set under linear motion in the plane can change quadratic number of times. We therefore investigate the problem of maintaining extent measures approximately. More precisely, suppose we want to maintain the smallest enclosing orthogonal rectangle. Let $B(t)$ denote the smallest orthogonal rectangle containing $P(t)$. For a given parameter $\varepsilon > 0$, let $B^\varepsilon(t)$ denote a rectangle such that $B(t) \subseteq B^\varepsilon(t) \subseteq (1+\varepsilon)B(t)$, where $(1+\varepsilon)B(t)$ denotes the rectangle resulting from scaling $B(t)$ by a factor of $(1+\varepsilon)$ and with respect to the center of $B(t)$. We call $B^\varepsilon(t)$ an *(outer) ε-approximation* of $B(t)$. The intuition is that one has to change the combinatorial description of $B^\varepsilon(t)$ considerably fewer times than that of $B(t)$. We can define similar ε-approximations for other extent measures like diameter.

1.2 Previous work

Motivated by various applications, there has been a flurry of activity in computational geometry, databases, and networking on problems dealing with moving objects. In the computational geometry community, earlier work on moving points focused on bounding the number of changes in various geometric structures (e.g., convex hull, Delaunay triangulation) as the points move [Ata85, SA95]. Basch *et al.* [BGH97] introduced the notion of *kinetic data structures*. Their work has lead to several interesting results related to moving objects, including results on kinetic convex-hull, binary space partition trees, collision detection, and closest pair; see [AEG98, Gui98] and references therein. The main idea in the kinetic framework is that as the points move and their *configuration* changes, *kinetic updates* are performed on the data structure when certain events occur. Although the points move continuously, the combinatorial structure changes only at certain discrete times at which certain events occur, and therefore one does not have to update the data structure continuously. In contrast to fixed-time-step methods, in which the fastest moving object determines the update time step for the entire data structure, a kinetic data structure is based on events, which have a natural interpretation in terms of the underlying structure.

In the context of maintaining an extent measure of a point set P, a result by Basch *et al.*[BGH97] gives a kinetic data structure for maintaining the smallest orthogonal rectangle $B(t)$ containing $P(t)$. It processes $O(n \log n)$ events if the motion is linear, and each event requires $O(\log n)$ time to update the combinatorial description of $B(t)$. A point can be inserted or deleted in $O(\log^2 n)$ time. Saltenis *et al.* [SJLL00] describe a heuristic to maintain a small rectangle enclosing a set of points moving in the plane. Agarwal *et al.* [AGHV97] proposed a kinetic data structure for maintaining the diameter, width, and a smallest enclosing rectangle (of arbitrary orientation) of a point set in the plane. Their structure processes $O(n^{2+\varepsilon})$ events for algebraic motion of points, and each event requires $O(\log n)$ time. No efficient data structure is known for maintaining the diameter or width of a point set in higher dimensions. The best known data structure for maintaining the smallest enclosing disk of a point set in the plane is the same as the one that maintains the farthest point Voronoi diagram. This processes $O(n^3)$ events for linear motion.

1.3 Our results.

Most of the work on kinetic data-structures had focused on maintaining exact geometric structure which forces them to process many events. We develop efficient algorithms for maintaining various extent measures approximately. The most salient feature of our data-structures is that the number of events processed depends only on the approximation factor and not on the input size. In the following, let P be a set of n points in $\mathbb{R}^d$, and let $\varepsilon > 0$ be a parameter. This paper contains the following main results.

Extent of points in $1D$. Let P be a set of n points in $\mathbb{R}$. We present a data structure for maintaining $B^\varepsilon(t)$, each endpoint of B^ε follows a piecewise-linear trajectory. The combinatorial structure of B^ε is updated $O(\sqrt{1/\varepsilon})$ times, and at each such event the extent can be updated in $O(\log n)$ time after $O(n \log n)$ preprocessing. Note that the number of combinatorial changes depends only on ε, and not on the number of points. A point can be inserted into or deleted from the structure in $O(\log^2 n)$ time.

Actually, we define the notion of extent for a set $\mathcal{H}$ of hyperplanes. We show that there exists a small set $\mathcal{J}$ of hyperplanes whose extent approximates the extent of $\mathcal{H}$ and whose size is independent of $|\mathcal{H}|$. Maintaining B^ε is a special case of maintaining the extent of hyperplanes.

Smallest enclosing rectangle under linear motion. Next, we consider the problem of maintaining the smallest enclosing orthogonal rectangle of a point set P in $\mathbb{R}^d$. The problem of computing the smallest rectangle is decomposable. That is, for each $j = 1, \ldots, d$, we can project the points in P onto the x_j-axis, compute the extent of the projected points (the smallest interval containing the points), and combine the result to get the bounding box. If I_j is the extent of the points projected on the x_j-axis, then the smallest rectangle containing P is $I_1 \times \cdots \times I_d$. Thus our one-dimensional result implies that we can maintain B^ε whose combinatorial structure changes $O(\sqrt{1/\varepsilon})$ times, and at each such event the rectangle can be updated in $O(\log n)$ time after $O(n \log n)$ preprocessing. A point can be inserted into or deleted from in $O(\log^2 n)$ time.

We also describe a data structure for maintaining a rectangle $\beta^\varepsilon(t)$ such that $(1-\varepsilon)B(t) \subseteq \beta^\varepsilon(t) \subseteq B(t)$ and such that $\beta^\varepsilon(t)$ is defined by a sequence of $2d$ points $(p_1, \ldots p_{2d})$ of P in the sense that $\beta^\varepsilon(t) = \prod_{i=1}^{d}[x^i_{2i-1}(t), x^i_{2i}(t)]$. The data structure has the same performance bounds as the previous one.

As mentioned above, this algorithm can be used as a building block for R-trees (or other hierarchical data structures) on moving points in $\mathbb{R}^d$. Namely, at each node v of the tree, we maintain an ε-approximation of the smallest rectangle B^ε_v enclosing the point subset associated with v. Since the representation of the kinetic bounding rectangle is small, we can store it at the node. In order to answer a range query — report all points that lie inside a rectangle R at time t — we check at each node v whether $B^\varepsilon_v(t) \subseteq R$. If so, we report all points in $S_v(t)$. If $R \cap B^\varepsilon(t) \neq \emptyset$, we recursively visit the children of v.

Smallest enclosing rectangle under algebraic motion. For degree k motion, we present a data structure for maintaining $B^\varepsilon(t)$, which processes $O(\log(1/\varepsilon)/\varepsilon)$ events. We currently use a naïve algorithm to update the box in $O(n)$ time at each event, but we believe that a more sophisticated algorithm can update the rectangle in $\log^{O(1)} n$ time at each event.

Diameter. For linear motion, we can maintain a pair of points (p, q) such that $d(p, q) \geq (1-\varepsilon)\,\mathrm{diam}(P)$. The pair is updated $O(\sqrt{1/\varepsilon^d})$ times. The total time spent in updating these events is $O((n/\varepsilon^{(d-1)/2})\log(1/\varepsilon))$. A similar approach can also maintain the smallest enclosing ball of a point set. In the plane, the number of events is roughly $O(1/\varepsilon^{5/2})$.

The paper is organized as follows. In Section 2, we define the extent for hyperplanes, review some techniques for handling the problem at hand, and prove bounds on maintaining the approximate extent of hyperplanes and the smallest enclosing rectangle. In Section 3, we present data-structures for maintaining the smallest enclosing rectangle. In Section 4, we extend these data-structures to maintaining the approximate diameter, and smallest enclosing ball. We then describe in Section 5 how to maintain B^ε for algebraic motion. We use the linearization technique to map the trajectories of points to linear functions in a higher dimension, compute an approximate extent in this parametric space, and then map the approximation back to the original plane.

2 Approximating the Extent

Let P be a set of n points in $\mathbb{R}$, each moving with fixed speed. That is $p_i(t) = a_i + b_i t$, where $a_i, b_i \in \mathbb{R}$. The *extent* $B(t)$ of $P(t)$ is the smallest interval containing $P(t)$. For a parameter ε, an *outer ε-extent* of $P(t)$ is an interval $B^\varepsilon(t) \subseteq (1+\varepsilon)B(t)$. (Here, and in the following, $(1+\varepsilon)I(t)$ denotes the set resulting by scaling $I(t)$ by a factor of $(1+\varepsilon)$, and centering it in the middle of $I(t)$.) Similarly, an *inner ε-extent* of $P(t)$ is an interval $\beta^\varepsilon(t) \supseteq (1-\varepsilon)B(t)$.

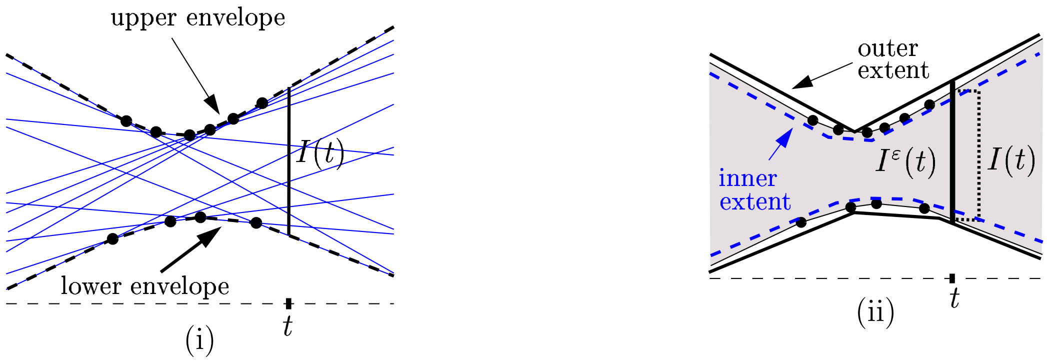

Figure 1: (i) The extent of the moving points, is no more than the vertical segment connecting the lower envelope to the upper envelope. The black dots mark where the movement description of $I(t)$ changes. (ii) The approximate extent.

It will be convenient to work in a parametric xt-plane in which a moving point $p(t) \in \mathbb{R}$ at time t is mapped to the point $(t, p(t))$. For $1 \le i \le n$, we map the point $p_i \in P$ to the line $\ell_i = \bigcup_t (t, p_i(t))$, for $i = 1, \ldots, n$. Let $L = \{\ell_1, \ldots, \ell_n\}$ be the resulting set of lines, and let $\mathcal{A}(L)$ be their arrangement. Clearly, the extent $B(t_0)$ of $P(t_0)$ is the vertical interval $I(t_0)$ in the arrangement $\mathcal{A}(L)$ connecting the upper and lower envelopes of L at $t = t_0$. See Figure 1(i). The combinatorial structure of $I(t)$ changes at the vertices of the two envelopes of L, and all the different combinatorial structures of $I(t)$ can be computed in $O(n \log n)$ time by computing the upper and lower envelopes of L.

We want to maintain a vertical interval $I^\varepsilon(t)$ so that $I(t) \subseteq I^\varepsilon(t) \subseteq (1+\varepsilon)I(t)$ for all t, so that the endpoints of $I^\varepsilon(t)$ follow piecewise-linear trajectories, and so that the number of combinatorial changes in $I^\varepsilon(t)$ is small. Geometricly, this has the following interpretation: We want to simplify the upper and lower envelopes of $\mathcal{A}(L)$ by convex and concave polygonal chains, respectively, so that the simplified upper (resp. lower) envelope lies above (resp. below) the original upper (resp. lower) envelope and so that for any t, the vertical segment connecting the simplified envelopes is contained in $(1+\varepsilon)I(t)$. See Figure 1 (ii). Actually, we generalize the notion of extent to arrangements of hyperplanes and prove a result on approximating the extent of hyperplanes.

2.1 Outer and inner approximations of extent

Let $h : x_d = a_1x_1 + \cdots a_{d-1}x_{d-1} + a_d$ be a hyperplane in $\mathbb{R}^d$. We will regard h as the graph of a linear function $h(x_1, \ldots, x_{d-1}) = a_1x_1 + \cdots + a_{d-1}x_{d-1} + a_d$ and will not distinguish between the function and its graph. A point $p = (p_1, \ldots, p_d)$ lies *above* (resp. *below*) h if $p_d > a_1p_1 + \cdots + a_{d-1}p_{d-1} + a_d$ (resp. $p_d < a_1p_1 + \cdots + a_{d-1}p_{d-1} + a_d$). For two parallel hyperplanes $h : x_d = a_d + \sum_{i=1}^{d-1} a_ix_i$ and $h' : x_d = a'_d + \sum_{i=1}^{d-1} a_ix_i$, the vertical distance between h and h' is defined as $\delta(h, h') = |a_d - a'_d|$. We will use $d(h, h')$ to denote the normal distance between h and h'. Note that $d(h, h') = \delta(h, h')/\sqrt{1 + \sum_{i=1}^{d-1} a_i^2}$.

Let $\mathcal{H} = \{h_1, \ldots, h_n\}$ be a set of hyperplanes in $\mathbb{R}^d$. The *lower envelope* of $\mathcal{H}$ is the graph of the linear function $\mathcal{L}_\mathcal{H} : \mathbb{R}^{d-1} \to \mathbb{R}$

$$\mathcal{L}_\mathcal{H}(\mathbf{x}) = \min_{h \in \mathcal{H}} h(\mathbf{x}).$$

Similarly, the *upper envelope* of $\mathcal{H}$ is the graph of the function

$$\mathcal{U}_{\mathcal{H}}(\mathbf{x}) = \max_{h\in\mathcal{H}} h(\mathbf{x}).$$

The *extent* $I_{\mathcal{H}} : \mathbb{R}^{d-1} \to \mathbb{R}$ of $\mathcal{H}$ is defined as

$$I_{\mathcal{H}}(\mathbf{x}) = \mathcal{U}_{\mathcal{H}}(\mathbf{x}) - \mathcal{L}_{\mathcal{H}}(\mathbf{x}).$$

With a slight abuse of notation, we will also use $\mathcal{L}_{\mathcal{H}}(\mathbf{x})$ and $\mathcal{U}_{\mathcal{H}}(\mathbf{x})$ to denote the points $(\mathbf{x}, \mathcal{L}_{\mathcal{H}}(\mathbf{x}))$ and $(\mathbf{x}, \mathcal{U}_{\mathcal{H}}(\mathbf{x}))$, respectively, in $\mathbb{R}^d$. Let $\varepsilon > 0$ a parameter. A set $\mathcal{J}$ of hyperplanes is an *outer ε-approximation* of $\mathcal{H}$ if the following two conditions hold for every point $\mathbf{x} \in \mathbb{R}^{d-1}$:

(i) $\mathcal{L}_{\mathcal{H}}(\mathbf{x}) - \frac{\varepsilon}{2} I_{\mathcal{H}}(\mathbf{x}) \le \mathcal{L}_{\mathcal{J}}(\mathbf{x}) \le \mathcal{L}_{\mathcal{H}}(\mathbf{x})$, and

(ii) $\mathcal{U}_{\mathcal{H}}(\mathbf{x}) \le \mathcal{U}_{\mathcal{J}}(\mathbf{x}) \le \mathcal{U}_{\mathcal{H}}(\mathbf{x}) + \frac{\varepsilon}{2} I_{\mathcal{H}}(\mathbf{x})$.

This implies that $I_{\mathcal{H}}(\mathbf{x}) \le I_{\mathcal{J}}(\mathbf{x}) \le (1+\varepsilon) I_{\mathcal{H}}(\mathbf{x})$.

Similarly, a set $\mathcal{K}$ of hyperplanes in an *inner ε-approximation* of $\mathcal{H}$ if the following two conditions hold for every point $\mathbf{x} \in \mathbb{R}^{d-1}$:

(i) $\mathcal{L}_{\mathcal{H}}(\mathbf{x}) \le \mathcal{L}_{\mathcal{K}}(\mathbf{x}) \le \mathcal{L}_{\mathcal{H}}(\mathbf{x}) + \frac{\varepsilon}{2} I_{\mathcal{H}}(\mathbf{x})$, and

(ii) $\mathcal{U}_{\mathcal{H}}(\mathbf{x}) - \frac{\varepsilon}{2} I_{\mathcal{H}}(\mathbf{x}) \le \mathcal{U}_{\mathcal{K}}(\mathbf{x}) \le \mathcal{U}_{\mathcal{H}}(\mathbf{x})$.

Therefore $(1-\varepsilon) I_{\mathcal{H}}(\mathbf{x}) \le I_{\mathcal{K}}(\mathbf{x}) \le I_{\mathcal{H}}(\mathbf{x})$.

2.2 Duality and extent

The *dual* of a point $b = (b_1, \ldots, b_d)$ is a hyperplane $b^* : x_d = b_1 x_1 + \cdots b_{d-1} x_{d-1} - b_d$, and the dual of a hyperplane $h : x_d = a_1 x_1 + a_2 x_2 + \cdots + a_{d-1} x_{d-1} + a_d$ is the point $h^* = (a_1, \ldots, a_{d-1}, -a_d)$. Under this definition of duality, $b^{**} = b$ and the point b lies above (resp. below, on) the hyperplane h, if and only if the point h^* lies above (resp. below, on) the hyperplane b^*. The vertical distance between b and h is the same as that between b^* and h^*, and the vertical distance $\delta(h, h')$ between two parallel hyperplanes h and h' is the same as the length of the vertical segment $h^* h'^*$. It can be checked that the hyperplane dual to the point $\mathcal{L}_{\mathcal{H}}(\mathbf{x})$ (resp. $\mathcal{U}_{\mathcal{H}}(\mathbf{x})$) is normal to the vector $(\mathbf{x}, -1) \in \mathbb{R}^d$ and supports $\mathrm{conv}(\mathcal{H}^*)$ so that $\mathcal{H}^*$ lies below (resp. above) the hyperplane and so that $I_{\mathcal{H}}(\mathbf{x})$ is the vertical distance between these two parallel supporting planes.

2.3 Approximating the extent via duality

In this section we use duality to show the existence of inner and outer ε-approximations of $\mathcal{H}$ of small size.

Let S be a set of points in $\mathbb{R}^d$. For a point $\mathbf{x} \in \mathbb{R}^{d-1}$, let $\lambda_S(\mathbf{x})$ (resp. $\rho_S(\mathbf{x})$) be the supporting hyperplane of $\mathrm{conv}(S)$ in direction $(\mathbf{x}, -1)$ so that S lies below (resp. above) it. Set $W_S(\mathbf{x}) = d(\lambda_S(\mathbf{x}), \rho_S(\mathbf{x}))$ denote the *projection width* of S in the direction of $\mathbf{x}$. In view of the above discussion, $\lambda_{\mathcal{H}^*}(\mathbf{x}), \rho_{\mathcal{H}^*}(\mathbf{x})$ are the hyperplanes dual to the points $\mathcal{L}_{\mathcal{H}}(\mathbf{x})$ and $\mathcal{U}_{\mathcal{H}}(\mathbf{x})$, respectively, and $I_{\mathcal{H}}(\mathbf{x}) = \delta(\lambda_{\mathcal{H}^*}(\mathbf{x}), \rho_{\mathcal{H}^*}(\mathbf{x}))$. For a parameter $\varepsilon > 0$, a set R of points in $\mathbb{R}^d$ is called an *outer ε-approximation* of S if $\mathrm{conv}(S) \subseteq \mathrm{conv}(R)$ and for all $\mathbf{x} \in \mathbb{R}^{d-1}$

$$d(\lambda_R(\mathbf{x}), \lambda_S(\mathbf{x})),\ d(\rho_R(\mathbf{x}), \rho_S(\mathbf{x})) \leq (\varepsilon/2) W_S(\mathbf{x}).$$

An *inner ε-approximation* of S is defined similarly.

Lemma 2.1 *Let $\mathcal{H}$ be a set of hyperplanes in $\mathbb{R}^d$, and let $\varepsilon > 0$ be a parameter. A set $\mathcal{J}$ of hyperplanes is an outer (resp. inner) ε-approximation of $\mathcal{H}$ if and only if $\mathcal{J}^*$ is an outer (resp. inner) ε-approximation of $\mathcal{H}^*$.*

Proof: We prove the lemma for outer approximations. Suppose $\mathcal{J}$ is an outer ε-approximation of $\mathcal{H}$. Then

$$\begin{aligned} d(\rho_{\mathcal{H}^*}(\mathbf{x}), \rho_{\mathcal{J}^*}(\mathbf{x})) &= \frac{\delta(\rho_{\mathcal{H}^*}(\mathbf{x}), \rho_{\mathcal{J}^*}(\mathbf{x}))}{\|(\mathbf{x},1)\|} = \frac{\mathcal{L}_{\mathcal{H}}(\mathbf{x}) - \mathcal{L}_{\mathcal{J}}(\mathbf{x})}{\|(\mathbf{x},1)\|} \leq \frac{(\varepsilon/2) I_{\mathcal{H}}(\mathbf{x})}{\|(\mathbf{x},1)\|} \\ &= \frac{\varepsilon}{2} \frac{\delta(\rho_{\mathcal{H}^*}(\mathbf{x}), \lambda_{\mathcal{H}^*}(\mathbf{x}))}{\|(\mathbf{x},1)\|} = \frac{\varepsilon}{2} W_{\mathcal{H}^*}(\mathbf{x}). \end{aligned}$$

Similarly, we can prove that $d(\lambda_{\mathcal{H}^*}(\mathbf{x}), \lambda_{\mathcal{J}^*}(\mathbf{x})) \leq \varepsilon W_{\mathcal{H}^*}(\mathbf{x})/2$. Hence, $\mathcal{J}^*$ is an outer ε-approximation of $\mathcal{H}^*$. A similar argument shows that if $\mathcal{J}^*$ is an outer ε-approximation of $\mathcal{H}^*$, then $\mathcal{J}$ is an outer ε-approximation of $\mathcal{H}$. ∎

Corollary 2.2 *Let P, Q be two sets in $\mathbb{R}^d$, and let T be any invertible affine transformation. Then Q is an inner (resp. outer) ε-approximation of P* **iff** *$T(Q)$ is an inner (resp. outer) ε-approximation to $T(P)$.*

Proof: Clearly, we can ignore the translation that T performs, and consider it to be a linear transformation. In particular, let A be the matrix representing T. Then,

$$\langle \mathbf{x}, A\mathbf{y} \rangle = \mathbf{x}^T A \mathbf{y} = (A^T \mathbf{x})^T \mathbf{y} = \left\langle A^T \mathbf{x}, \mathbf{y} \right\rangle.$$

Thus, $W_{T(P)}(\mathbf{x}) = W_P(A^T\mathbf{x})$. But, by definition, $W_P(A^T\mathbf{x})$ is inner ε-approximated by $W_Q(A^T\mathbf{x}) = W_{T(Q)}(\mathbf{x})$. ∎

Definition 2.3 A convex-set $\mathcal{C}$ is α-fat, for $\alpha \geq 1$, if there exists two balls B_r, B_R with radiuses r, R, respectively. So that $B_r \subseteq \mathcal{C} \subseteq B_R$ and $R/r \leq \alpha$. A point-set P is *α-fat* if $\mathcal{CH}(P)$ is α-fat.

Lemma 2.4 *For a set P of n points in $\mathbb{R}^d$, one can compute in linear time an affine transformation T, such that $\mathcal{CH}(T(P))$ is a α_d-fat, and $\alpha_d \geq 1$ is a constant that depends only on the dimension.*

Proof: By Barequet and Har-Peled [BH01] (see also [Mac51]) there exists a box B, a constant $c_d > 0$ (which depend only on the dimension), and a vector $v \in \mathbb{R}^d$, such that

$$c_d B + v \subseteq \mathcal{CH}(P) \subseteq B.$$

The box B can be easily computed in linear time, by computing an approximate diameter pq of P, projecting the point-set of P in a perpendicular plane hyperplane π, computing recursively a tight-fitting bounding box on the projected point-set, and then lifting it back into a bounding box in $\mathbb{R}^d$, by using the bounding-box in π as a base for the minimum d-dimensional bounding box that contains P, and have a sides parallel to pq. See [BH01] for details.

It is easy to verify that the affine transformation T mapping B to the unit cube in $\mathbb{R}^d$ is the required transformation, and that $\mathcal{CH}(T(P))$ is $O(c_d)$-fat. ∎

Thus, by Corollary 2.2 and Lemma 2.4 to approximate the extent of a set of hyperplanes in $\mathbb{R}^d$, it is enough to approximate an appropriate fat point-set in the dual.

Lemma 2.5 *For a set P of n points in $\mathbb{R}^d$, one can compute in linear time an inner (resp. outer) ε-approximation U to P, so that $|U| = O(1/\varepsilon^{d-1})$.*

Proof: We compute in linear time Lemma 2.4 the mapping T that transform P into a fat set $S = T(P)$.

Let B the axis parallel bounding box of S. Let G be the grid spanned by the box $B_\varepsilon = (\varepsilon/2)c_d B$ (c_d is an appropriately small constant); namely, each cell of G is a translated copy of B_ε. Next, let S_G be the set of points resulting by replacing each point of S by the vertices of the cell of G that contains this point. Note, that $|S_G| = O(1/\varepsilon^d)$, $\mathcal{CH}(S) \subseteq \mathcal{CH}(S_G)$, and that for any direction vector μ, the distance between the endpoints of $V_\mu(S)$ and $V_\mu(S_G)$ is smaller than $I_{B_\varepsilon}(\mu) \leq (\varepsilon/2)I_{c_d B}(\mu) \leq I_{\mathcal{CH}(S)}(\mu) \leq (\varepsilon/2)I_S(\mu)$. Thus, S_G ε-approximates S.

Since the measure $I_{S_G}(\mu)$ is sensitive only to the vertices of the convex-hull of S_G, it is possible to throw away points which are not extreme. In particular, for any grid line ℓ of G that contains more than two points of S_G, one can throw away the middle points, keeping only the two extreme points. Let $\mathcal{S}_G$ denote the resulting set. It is easy to verify that $|\mathcal{S}_G| = O(1/\varepsilon^{d-1})$, $\mathcal{S}_G$ can be computed in linear time in n (by using hashing/bucketing), and $\mathcal{S}_G$ is an ε-approximation to S. In particular, $U = T^{-1}(\mathcal{S}_G)$ is an outer ε-approximation to P.

An inner approximation can be computed by replacing each snapped point of $\mathcal{S}_G$ by the original point that was mapped to it. ∎

Lemma 2.6 *For a set P of n points in $\mathbb{R}^d$, one can compute in $O(n + 1/\varepsilon^{3(d-1)/2})$ time an inner (resp. outer) ε-approximation U to P, so that $|U| = O(1/\varepsilon^{(d-1)/2})$. Furthermore, the set realizing the inner approximation is a subset of P.*

Proof: As in the proof of Lemma 2.5 we map P into a fat point-set $S_1 = T(P)$. We compute an inner $\varepsilon/3$-approximate S_1 by the algorithm of Lemma 2.5. Let S_2 denote the resulting point-set (note that the points of S_2 are subset of the points of S_1).

We can now use the algorithm of [Dud74] to compute an $O(\sqrt{\varepsilon})$-dense set U of points on the boundary of $\mathcal{S} = \mathcal{CH}(S_2)$ (see [Dud74] for details). This requires projecting $O(1/\varepsilon^{(d-1)/2})$ points in $\mathbb{R}^d$ into their nearest point on $\partial\mathcal{S}$. Fortunately, each such projection can be done in linear time in $|S_2|$ by using the algorithm of [Gär95]. Thus, U can be computed in $O(1/\varepsilon^{3(d-1)/2})$ time.

Arguing as in [BI76], it is easy to verify that U is an inner $\varepsilon/3$-approximation to S_2. Furthermore, each point of U lies on a simplex induced by d-points of S_2. Thus, let U' be the set resulting by replacing each point of U by its d supporting points in S_2. Clearly, $|U'| = O(1/\varepsilon^{(d-1)/2})$, $U' \subseteq S_2 \subseteq S_1$, and it is an inner $\varepsilon/3$-approximation to S_2. However, S_2 is an $\varepsilon/3$-approximation to S_1, and we thus conclude that U' is an ε-approximation to S_1, and $T^{-1}(U)$ is an inner ε-approximation to P.

To get an outer ε-approximation to P, we repeat the same algorithm, with the following modification. We compute S_2 to be an outer ε-approximation to S_1. We compute U the $O(\sqrt{\varepsilon})$-dense set on $\partial\mathcal{S}$ as above, where $\mathcal{S} = \mathcal{CH}(S_2)$. We lift each point of U slightly in the direction of its normal to the $\partial\mathcal{S}$ (see [BI76] for details). One can then argue that the resulting set U' is an outer $\varepsilon/3$-approximation to S_1. Thus $T^{-1}(U')$ is the required outer ε-approximation to P. ∎

We conclude:

Theorem 2.7 *Given a set $\mathcal{H}$ of n hyperplanes in $\mathbb{R}^d$ and a parameter $\varepsilon > 0$, there exists a set $\mathcal{J}$ of $O(1/\varepsilon^{(d-1)/2})$ hyperplanes that is an outer (resp. inner) ε-approximation of $\mathcal{H}$. Furthermore, it can be computed in $O(n + 1/\varepsilon^{3(d-1)/2})$ time.*

Remark 2.8 The lower-bound construction in [BI76] implies that the bounds presented above in Theorem 2.7 are tight.

An immediate corollary of Theorem 2.7 is the following result on maintaining the smallest enclosing rectangle of a moving point set.

Corollary 2.9 *Given a set $P(t)$ of n linearly moving points in $\mathbb{R}^d$ and a parameter $\varepsilon > 0$, one can compute an ε-approximation $B^\varepsilon(t)$ of the smallest rectangle enclosing $P(t)$ such that the number of events is $O(\sqrt{1/\varepsilon})$.*

3 Maintaining the Extent

In this section, we describe how to maintain outer and inner ε-extents of a point set in $\mathbb{R}$. One can, of course, use the construction described in the previous section to maintain an outer ε-extent that changes $O(1/\sqrt{\varepsilon})$ times, but we describe a different algorithm that computes an outer ε-extent whose combinatorial structure changes at most $\mathrm{Opt}(\varepsilon)$ times, where $\mathrm{Opt}(\varepsilon)$ is the minimum size of an ε-extent. Again, we work in the parametric xt-plane.

3.1 Maintaining an outer ε-extent

Let L be the set of lines in the xt-plane as defined in the beginning of Section 2. Unlike the algorithm in the previous section, we do not compute an outer ε-approximation of L. Instead we compute the trajectories of the endpoints of the vertical segment I^ε. We describe the algorithm for maintaining the trajectory of the upper endpoint of I^ε; the lower endpoint can be maintained in a similar manner.

Let $f(t) = \mathcal{U}_L(t)$ be the linear function defining the upper envelope of L, and let

$$g(t) = f(t) + \varepsilon I_L(t)/2 = (1 + \varepsilon/2)f(t) - \varepsilon\mathcal{L}_L(t)/2.$$

Since $f(t)$ is a convex function and $\mathcal{L}_L(t)$ is a concave function, the graph of $g(t)$ is a convex polygonal chain.

We first compute in $O(n \log n)$ time the graphs of f and g. We then compute an almost minimum-link path in the "corridor" lying between $f(t)$ and $g(t)$. This path is the trajectory of the upper endpoint of ε-extent I^ε. Inductively, we maintain the invariant that the upper endpoint of I^ε moves along a line tangent to $g(t)$ and that it lies in the corridor. Initially, we choose a ray passing through the leftmost vertex of g and parallel to the leftmost edge of f. Suppose currently, the upper endpoint of $I^\varepsilon(t)$ is following a ray h_{i-1}. We compute the time t_i at which h_{i-1} intersects the upper envelope f (i.e., the time at which it tries to leave the corridor). We then compute in $O(\log n)$ time the rightward directed ray h_i emanating from $f(t_i)$ and tangent to g. The details are straightforward, and we omit them. If the minimum-link polygonal chain that lies between f and g consists of k vertices, then the above algorithm computes a convex polygonal chain with at most k vertices [ABO+89]. We summarize:

Lemma 3.1 *One can compute in $O(n \log n)$ time a convex polygonal chain C lying between f and g with k vertices, where k is the number of vertices in the minimum-link chain lying between f and g.*

We run the same algorithm for computing the trajectory of the lower endpoint of I^ε. For a given $\varepsilon > 0$, let $\mathrm{Opt}(\varepsilon)$ denote the minimum number of times the combinatorial structure of any ε-extent of P has to change, i.e., the number of times the (linear) trajectory of one of its endpoints changes. The number of events processed by our algorithm is bounded by $\mathrm{Opt}(\varepsilon)$ (the algorithm might compute one extra link for each of the two endpoints). We thus have the following result.

Theorem 3.2 *Given a set P of n points in $\mathbb{R}^d$ under linear motion and a parameter $\varepsilon > 0$, one can maintain an outer ε-approximation $B^\varepsilon(t)$ of the smallest rectangle $B(t)$ containing $P(t)$ in a total of $O(n \log n)$ time, whose combinatorial structure changes at most $\mathrm{Opt}(\varepsilon) = O(\sqrt{1/\varepsilon})$ times.*

3.1.1 Faster Algorithm

One can compute the outer ε-approximation even faster as described in this section.

Lemma 3.3 *Let L be a set of n lines, $f(t) = \mathcal{U}_L(t)$, $g(t) = f(t) + \varepsilon I_L(t)/2$, and p a point lying between the graphs of f and g. One can find the ray emanating for p that is tangent to g, to the right of p, in linear time. Similarly, one can answer ray-shooting queries on f and g in linear time.*

Proof: Let q^* be the requested tangency point on the graph of $g(t)$. Observe, that for a given coordinate t_0 we can compute in linear time the value of $f(t_0), g(t_0)$ and the slope of $g(t_0)$ at this point. In particular, since the graph of $g(t)$ is convex, we can decide whether $q*$ is to the right of left of t_0 in linear time.

If $|L| = O(1)$ we can solve the problem using brute force. Otherwise, let R be a random sample of L of size $O(r \log r)$, where r is a large enough constant. Compute in $O(r \log^2 r)$ the lower and upper envelope of $\mathcal{A}(R)$. By VC dimensions arguments [AS00, Theorem 4.4], with probability at least half, each vertical trapezoid of upper/lower envelopes of $\mathcal{A}(R)$ intersects at most n/r lines of L. Those conflict list can be computed in $O(nr \log r)$ time, and if this is false, we resample.

Let S be the t-coordinates of all the vertices in the upper and lower envelope of $\mathcal{A}(R)$. Since $|S| = O(r \log r)$, we can compute in $O(nr \log r)$ time the two consecutive values of t, that contain q^* between them. Let J denote this interval.

Note, that there are two vertical trapezoids Δ_1, Δ_2, on the upper and lower envelopes of $\mathcal{A}(R)$ respectively, that contain the vertical strip defined by J. Furthermore, the extent of L in the range defined by J is defined by the lines of $L' = \mathrm{cl}(\Delta_1) \cup \mathrm{cl}(\Delta_2)$, where $\mathrm{cl}(\Delta_1)$ is the conflict list of Δ_1.[1]

In particular, we can now compute q^* by rerunning this algorithm on L'. Since $|L'| = 2n/r$, the overall expected running time of the algorithm is $T(n) = O(nr \log r) + T(2n/r)$. Thus, picking r to be a large enough constant, results in expected linear running time. ∎

Lemma 3.4 *One can compute in $O(n \log k)$ expected time a convex polygonal chain C lying between f and g with at most k vertices, where k is the number of vertices in the minimum-link chain lying between f and g.*

Proof: We extent the algorithm of Lemma 3.3, as follows: We start the ray-shooting process left to right, and resolves the tangency/ray-shooting using the algorithm of Lemma 3.3. However, instead of starting each time we perform a ray-shooting queries from scratch, we instead compute two trees that represent the lower and upper envelope of L. Each node of those trees corresponds to a subset of L, and an associated vertical trapezoid. As in Lemma 3.3, we sample inside the conflict list of the node, and the vertical trapezoids forming the upper envelope of the sample (resp. lower envelope) are the children of this node. Continuing recursively, results in the two trees.

Observe, that only nodes along the k ray-shooting/tangency queries needed to be constructed. In particular, if we perform k queries, the overall expected time spent in constructing and answering the queries is $O(n \log k)$, as can be easily verified. ∎

Theorem 3.5 *Given a set P of n points in $\mathbb{R}^d$ under linear motion and a parameter $\varepsilon > 0$, one can maintain an outer ε-approximation $B^\varepsilon(t)$ of the smallest rectangle $B(t)$ containing $P(t)$ in a total of $O(n \log(1/\varepsilon))$ expected time, whose combinatorial structure changes at most $\mathrm{Opt}(\varepsilon) = O(\sqrt{1/\varepsilon})$ times.*

3.2 Maintaining an inner extent

The problem with the algorithm described above is that since $B(t) \subseteq B^\varepsilon(t)$, the endpoints of $B^\varepsilon(t)$ are not two of the input points. In some applications, we would like to maintain an interval $\beta(t) \supseteq (1-\varepsilon)B(t)$ whose endpoints are two of the input points; these endpoints act as a witness of the extent of P. Again, we focus on the one dimensional case. We call an inner ε-extent $\beta(t)$ of $P(t)$ *strong* if the endpoints of β are two of the input points. The algorithm described in Section 2.3 for computing the inner extent also does not compute a stronger inner ε-extent. We therefore describe a new algorithm.

[1]We consider the lines that define the floor and ceiling of Δ_1, Δ_2 to be in their trapezoid conflict list.

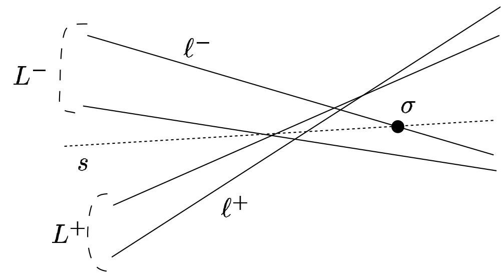

Figure 2: Illustration of the proof of Lemma 3.6.

Lemma 3.6 *let P be a set of points in $\mathbb{R}$. Given an ε-outer extent $B^\varepsilon(t)$ of $P(t)$, whose combinatorial structure changes k times, one can compute a strong 2ε-inner extent of $P(t)$ with at most $4k+4$ events, where $\varepsilon \le 1$.*

Proof: As in the previous subsections, we work in the parametric xt-plane. Let $I(t)$ (resp. $I^\varepsilon(t)$) denote the vertical segment in the xt-plane corresponding to the extent (resp. ε-outer extent) of $P(t)$. Define $I'(t) = (1-\varepsilon)I^\varepsilon(t)$. It is easily seen that $(1-2\varepsilon)I(t) \subseteq I'(t) \subseteq I(t)$. Note, that each event point of $I(t)$ becomes an event point for both the upper and lower envelope of $I'(t)$, and thus $I'(t)$ complexity is twice the complexity of $I(t)$. In particular, the number of segments in the upper/lower envelopes of $I'(t)$ is $\le k+1$. The endpoints of $I'(t)$ follow polygonal chains whose vertices lie at the values of t at which the combinatorial structure of $I^\varepsilon(t)$ changes. We construct a vertical interval $i^\varepsilon(t)$ whose endpoints lie on the lines of L and $I'(t) \subseteq i^\varepsilon(t) \subseteq I(t)$.

We will choose (described below) a subset $R^+ \subset L$ of at most $2k+2$ lines so that its upper envelope lies above $I'(t)$ for all t. We use the upper envelope of lines of R^+ as the trajectory of the upper endpoint of $i^\varepsilon(t)$. Let s be an edge of the polygonal chain followed by the upper endpoint of $I'(t)$. If s lies entirely below a line $\ell \in L$, we add ℓ to R^+. Otherwise, let L^- (resp. L^+) be the subset of lines in L whose slopes are less (resp. greater) than that of s, and let $\ell^- \in L^-$ be the line whose intersection with s has the maximum t-coordinate. Set $\sigma = \ell^- \cap s$. Since σ lies below the upper envelope of L, a line ℓ^+ of L lies above σ. One can verify that the upper envelope of $\{\ell^-, \ell^+\}$ lies above s, and we add them to R^+. See Figure 2.

Similarly, we can find another subset $R^- \subseteq L$ of at most $2k+2$ lines so that their lower envelope lies below $I'(t)$ for all values of t. Hence, we can maintain an 2ε-inner extent of P with at most $4k+4$ events. ■

Omitting all the algorithmic details, which are similar to the previous algorithm, we obtain the following.

Lemma 3.7 *We can maintain a strong inner ε-extent of $P(t) \subseteq \mathbb{R}$ in $O(n \log 1/\varepsilon)$ time, whose combinatorial structure changes at most $O(1/\sqrt{\varepsilon})$ times.*

Proof: We computing an $\varepsilon/2$-outer extent $J'(t)$ of $P(t)$ using the algorithm of Theorem 3.5. Let $J(t) = (1-\varepsilon)J'(t)$, and let $\mathfrak{U}, \mathfrak{L}$ denote the upper and lower chains of $J(t)$, respectively. Since $\mathfrak{U}, \mathfrak{L}$ are convex polygons, as can be easily verified, and their complexity is $O(1/\sqrt{\varepsilon})$, one can preprocess them in linear time, so that given a line $\ell \in L$, one can compute, in $O(\log 1/\varepsilon)$ time, what edges of $\mathfrak{U}, \mathfrak{L}$ are crossed by ℓ. Performing this for all the lines of L takes $O(n \log 1/\varepsilon)$ time, and once this is done, one can compute for each edge e of $\mathfrak{U}$ two lines of L that their upper envelope lie above e. Let L^+ be this set of lines. One can compute in a similar fashion a corresponding set L^- for $\mathfrak{L}$. Clearly, the upper/lower envelopes of $L^- \cup L^+$ is the required approximation. Overall, the running time of the algorithm is $O(n \log 1/\varepsilon)$. ∎

Theorem 3.8 *Given a set P of n points in $\mathbb{R}^d$ under linear motion and a parameter $\varepsilon > 0$, one can maintain, in a total of $O(n \log n)$ time, an inner ε-approximation $\beta^\varepsilon(t)$ of $B(t)$ whose facets contain input points and whose combinatorial structure changes at most $O(1/\sqrt{\varepsilon})$ times.*

3.3 Maintaining the extent dynamically

Recall that our algorithm for maintaining the extent performs the following two queries: (i) compute the intersection point of a line with the upper envelope f, and (ii) given a point p below g, find the ray emanating from p in the rightward direction and tangent to g. We can use the dynamic data structure by Overmars and van Leeuwen [OvL81], which inserts or deletes a line in $O(\log^2 n)$ time and answers a query of type (i) or (ii) in $O(\log n)$ time. However, we have an additional difficulty because the convex chain g depends on both $\mathcal{U}_L$ and $\mathcal{L}_L$. We therefore store both the envelopes of L separately, using the above data structure. For a given value t, one can then compute, in $O(\log n)$ time, $g(t)$ and its derivative at t (if $g(t)$ is a vertex of the chain, then we compute the slope of the two edges incident upon the vertex). Using this operation as the primitive, a type (ii) query can be answered in $O(\log^2 n)$ time. Our overall algorithm remains the same. We now define an event to be the time at which a point is inserted or deleted or at which the combinatorial structure of the extent changes. Hence, we obtain the following.

Theorem 3.9 *Given a set P of n points in $\mathbb{R}$ under linear motion, one can maintain an outer or a strong inner ε-extent of P, so that it can be updated in $O(\log^2 n)$ at each event.*

Unfortunately, the data structure of Overmars and van Leeuwen is too complicated to be of practical use. We can use a simpler hierarchical data structure to maintain $B^\varepsilon(t)$. This data structure is especially suitable for applications (such as R-trees) in which we store a set of points in a tree T, each of whose node v is associated with the subset S_v of points stored in the subtree rooted at that node, and in which we wish to maintain a bounding rectangle $B_v^\varepsilon \subseteq (1+\varepsilon)B(S_v)$ at each node v. For simplicity, we describe the algorithm for one dimensional case. For higher dimensions, we repeat the same algorithm for each axis.

We set $\delta = \varepsilon/(c \log_r n)$, where c is a sufficiently large constant. For each node v of T, let L_v be the set of lines corresponding to the trajectories of the points in S_v. If the height of v is i, we maintain an outer $(2i\delta)$-approximation $\mathcal{J}_v$ of L_v and also the upper and lower envelopes of L_v. (Note that we do not maintain an outer $2i\delta$-extent using the on-line algorithm described in Section 3.1.) The insertion/deletion procedure visits a path Π from the root to a leaf of the tree. After having inserted or deleted a point at a leaf, we recompute the extent information at each node v on Π in a bottom-up manner, as follows. Let $J = \bigcup_w \mathcal{J}_w$, where w is a child of v; $|J| = O(r/\sqrt{\delta})$. We compute in $O(r/\sqrt{\delta}\log(1/\delta))$ time an outer δ-approximation $\mathcal{J}_v$ of $\mathcal{J}$ and the upper and lower envelopes of $\mathcal{J}_v$. It is easily seen that $\mathcal{J}_v$ is an outer $(2i\delta)$-approximation of L_v. Since Π has $O(\log_r n)$ nodes, an update operation takes

$$O\left((\log_r n) \cdot \frac{r}{\sqrt{\delta}} \log \frac{1}{\sqrt{\delta}}\right) = O\left(\frac{r\,(\log_r n)^{3/2}}{\sqrt{\varepsilon}} \log\left(\frac{\log_r n}{\varepsilon}\right)\right)$$

time. We thus obtain the following.

Theorem 3.10 *For a tree T storing n points $P(t)$, with a maximal out-degree r and depth $O(\log_r(n))$, one can perform insertion/deletion operations so that the time spent on updating the inner/outer ε-extent of the nodes is $O((r/\sqrt{\varepsilon})(\log_r n \log\log n)^{3/2})$.*

A similar idea can be used to maintain the outer and inner ε-approximations of a set of hyperplanes under insertions and deletions. Omitting all the details we conclude the following.

Theorem 3.11 *For a tree T storing n hyperplanes in $\mathbb{R}^d$, with a maximal out-degree r, and depth $O(\log_r(n))$, one can perform insertion/deletion operations so that the time spent on updating the inner/outer ε-approximate extent of the nodes is*

$$O\left(\frac{r}{\varepsilon^{d-1}} \log^{d-1} n \log_r n\right)$$

time. Each node stores a set of $O(1/\varepsilon^{d-1})$ hyperplanes that ε-approximates the extent of the hyperplanes in this subtree.

4 Maintaining Other Extent Measures

In this section, we show how the algorithms described in the previous section can be used to maintain the diameter, and the smallest enclosing disk of a set of points moving linearly in $\mathbb{R}^2$.

Diameter. Let P be a set of n linearly moving points in the plane, and let $\varepsilon > 0$ be a parameter. We choose a sequence $\mathbf{n}_1, \ldots, \mathbf{n}_k \in \mathbb{S}^1$ of $k = O(1/\sqrt{\varepsilon})$ directions so that the angle between two consecutive directions is at most $\sqrt{\varepsilon}/2$. Let $P^i(t)$ denote the projection of P on a line ℓ_i in direction $\mathbf{n}_i$. Choose $\delta = \varepsilon/2$. We maintain a strong inner δ-extent $\beta_i^\delta(t)$ of $P^i(t)$.

Lemma 4.1 $\max_{1\le i\le k} |\beta_i^\delta(t)| \geq (1-\varepsilon)\operatorname{diam}(P(t))$.

Proof: Suppose $\operatorname{diam}(P(t)) = d(p(t), q(t))$. Let $\mathbf{n}_i$ be the direction closest to the line $p(t)q(t)$, and let p^i, q^i be the projections of p and q on ℓ_i. The angle between ℓ_i and $p(t)q(t)$ is at most $\sqrt{\varepsilon}/2$. Therefore

$$\begin{aligned} |\beta_i^\delta(t)| &\geq |p^i(t)q^i(t)| \geq d(p(t), q(t))\cos(\sqrt{\varepsilon}/2) = \operatorname{diam}(P(t))(1 - 2\sin^2(\sqrt{\varepsilon}/4)) \\ &\geq (1-\varepsilon)\operatorname{diam}(P(t)). \end{aligned}$$

■

Hence, it suffices to maintain the maximum of the set $\mathcal{B}(t) = \{|\beta_i^\delta(t)| \mid 1 \le i \le k\}$. Recall that $|\beta_i^\delta|$ is a piecewise-linear function with a total of $O(\sqrt{1/\varepsilon})$ vertices. By Theorem 3.8, the total time spent in computing the set $\mathcal{B}$ is $O((n/\sqrt{\varepsilon})\log(1/\varepsilon))$.

We can use a kinetic data structure by Basch *et al.* [BGH97] to maintain the maximum of $\mathcal{B}$. This structure processes at most

$$O\left(\frac{1}{\sqrt{\varepsilon}}\frac{1}{\sqrt{\varepsilon}}\log\left(\frac{1}{\varepsilon}\right)\alpha\left(\frac{1}{\varepsilon}\right)\right) = O\left(\frac{\log(1/\varepsilon)\alpha(1/\varepsilon)}{\varepsilon}\right)$$

events, and it spends $O(\log(1/\varepsilon))$ time at each of these events. Hence, we obtain the following.

Theorem 4.2 *Given a set P of n linearly moving points in the plane and parameter $\varepsilon > 0$, there is a kinetic data structure for maintaining an ε-approximation of the diameter of S that processes $O(\log(1/\varepsilon)\alpha(1/\varepsilon)/\varepsilon)$ events and spends a total of $O((n/\sqrt{\varepsilon})\log(1/\varepsilon))$ time on these events.*

Remark 4.3 We can also insert or delete a point in $O(\log^2(n)/\sqrt{\varepsilon})$ time. Moreover the above algorithm can be extended to $\mathbb{R}^d$. We now have to choose a set of $O(1/\varepsilon^{(d-1)/2})$ directions, so the number of events processed by the data structure is $O(\log^2(1/\varepsilon)/\varepsilon^{d/2})$.

Smallest enclosing disk. Let $\mathcal{B}$ be the set as defined above. For each i, let $p_i(t)$ and $q_i(t)$ be the endpoints of the δ-inner extent $\beta_i^\delta(t)$. Set

$$W(t) = \{p_i(t), q_i(t) \mid 1 \le i \le k\}.$$

For a set X in the plane, let $r(X)$ be the radius of the smallest disk enclosing X. We can prove the following.

Lemma 4.4 $r(W(t)) \ge (1-\varepsilon)r(P(t))$.

We can therefore maintain the smallest enclosing disk of $W(t)$. Unfortunately, the best known kinetic data structure to maintain the smallest enclosing disk of a point set, which basically maintains the farthest-point Voronoi diagram of the point set, processes cubic number of events. Whenever the set W changes, we restart from the beginning, spending $O(\log(1/\varepsilon)/\sqrt{\varepsilon})$ time reconstructing the data structure. Hence, the total number of events processed by the structure is $O((1/\sqrt{\varepsilon})^3 \cdot (1/\sqrt{\varepsilon}) \cdot 1/\sqrt{\varepsilon}) = O(1/\varepsilon^{5/2})$. We thus obtain the following.

Theorem 4.5 *Given a set P of n linearly moving points in the plane and parameter $\varepsilon > 0$, there is a kinetic data structure for maintaining an ε-approximation of the smallest enclosing disk of S that processes $O(1/\varepsilon^{5/2})$ events and spends a total of $O((n/\sqrt{\varepsilon})\log(1/\varepsilon))$ time at these events.*

5 Handling Higher Degree Motion

In this section, we consider the case in which the degree of motion of P is k for some constant $k \geq 1$. As in Section 2, we focus on points in $\mathbb{R}$, that is, $p_i(t) = a_i^0 + \sum_{j=1}^{k} a_i^j t^j$. The goal is to compute an outer ε-extent of $B(t)$. Instead of working with a set of curves (graphs of trajectories of points) in the parametric xt-plane, we use the so-called *linearization* technique (see e.g. [YY85, AM94]). We map each point $p_i \in P$ to a hyperplane h_i in $\mathbb{R}^{k+1}$ $h_i : x_{k+1} = a_i^0 + \sum_{j=1}^{k} a_i^j x_j$. Set $\mathcal{H} = \{h_1, \ldots, h_n\}$. Let $\mu : \mathbb{R} \to \mathbb{R}^k$ be the curve $\mu(t) = (t, t^2, \ldots, t^k)$, then $p_i(t) = h_i(\mu(t))$. For brevity, let $\mathcal{U}(\mathbf{x}) = \mathcal{U}_{\mathcal{H}}(\mathbf{x})$, $\mathcal{L}(\mathbf{x}) = \mathcal{L}_{\mathcal{H}}(\mathbf{x})$, and $I(\mathbf{x}) = I_{\mathcal{H}}(\mathbf{x})$. The lower (resp. upper) endpoint of $B(t)$ is $L(\mu(t))$ (resp. $U(\mu(t))$, and $|B(t)| = I(\mu(t))$. Using Theorem 2.7, we can compute an outer ε-approximation $\mathcal{J}$ of $\mathcal{H}$ of size $O(1/\varepsilon^{k/2})$. The restriction of $I_{\mathcal{J}}(x)$ to the curve μ gives an outer ε-extent of P, whose combinatorial structure changes roughly $O(1/\varepsilon^{k/2})$ times.

A drawback of this approach is that the bound depends on k, the degree of motion. In this section we describe a different approach that can maintain an ε-extent of P whose combinatorial structure changes $O((1/\varepsilon)\log 1/\varepsilon)$ times. In particular, we compute two piecewise-linear functions $\mathcal{L}^\varepsilon, \mathcal{U}^\varepsilon : \mathbb{R}^k \to \mathbb{R}$ so that

(C1) $\mathcal{L}(\mathbf{x}) - (\varepsilon/2)I(\mathbf{x}) \leq \mathcal{L}^\varepsilon(\mathbf{x}) \leq \mathcal{L}(\mathbf{x})$, and

(C2) $\mathcal{U}(\mathbf{x}) \leq \mathcal{U}^\varepsilon(\mathbf{x}) \leq \mathcal{U}(\mathbf{x}) + (\varepsilon/2)I(\mathbf{x})$.

We will show that the restriction of $I^\varepsilon(\mathbf{x}) = \mathcal{U}^\varepsilon(\mathbf{x}) - \mathcal{L}^\varepsilon(\mathbf{x})$ to the curve μ has $O((1/\varepsilon)\log 1/\varepsilon)$ breakpoints. We first prove the following result.

Theorem 5.1 *Given a set of n hyperplanes in $\mathbb{R}^{k+1}$ and a parameter $\varepsilon > 0$, one can compute in time $O(n + 1/\varepsilon^{3k/2}\log 1/\varepsilon)$ two piecewise-linear functions $\mathcal{L}^\varepsilon, \mathcal{U}^\varepsilon : \mathbb{R}^k \to \mathbb{R}$ that satisfy conditions (C1) and (C2) and that consist of $O(1/\varepsilon^k \log(1/\varepsilon))$ k-simplices.*

Note that, unlike Section 2, we do not compute a small set of hyperplanes whose lower and upper envelopes approximate the extent of $\mathcal{H}$.

This theorem is proved in three stages. In the first stage, we apply a transformation on $\mathcal{H}$ so that the resulting set of hyperplanes are "well behaved". We then draw a d-dimensional box $\mathcal{C}$ on the hyperplane $x_{k+1} = 0$ and construct $\mathcal{U}^\varepsilon(\mathbf{x})$ and $\mathcal{L}^\varepsilon(\mathbf{x})$ for $\mathbf{x} \in \mathcal{C}$. We then compute the approximate envelopes on the boundary $\partial\mathcal{C}$, by invoking the algorithm recursively in one lower dimension, and then extend them to the exterior of $\mathcal{C}$.

Definition 5.2 For positive constants α, β, a set $\mathcal{H}$ of hyperplanes in $\mathbb{R}^{k+1}$ is (α, β)-*normalized* if the following conditions hold:

(i) $I(\mathbf{x})$ is minimum at the origin o;

(ii) the hyperplane $x_{k+1} = 0$ is in $\mathcal{H}$;

(iii) for any point $v \in \mathbb{R}^k$, there exists a hyperplane $h \in \mathcal{H}$, such that $|h(v) - h(o)| \geq \alpha||v||$, where o denotes the origin; and

(iv) for any vector $v \in \mathbb{R}^k$ and for any $h \in \mathcal{H}$, $|h(v) - h(o)| \leq \beta||v||$.

Let $\mathcal{H}$ be a set of hyperplane in $\mathbb{R}^{k+1}$. We represent a hyperplane in $\mathcal{H}$ as $h(\mathbf{x}) \equiv x_{k+1} = \mathbf{a} \cdot \mathbf{x} + b$, where $\mathbf{a} \in \mathbb{R}^k$ and $b \in \mathbb{R}$ are constants, and $\mathbf{x} \in \mathbb{R}^k$. For a hyperplane $h'(\mathbf{x}) \equiv x_{k+1} = \mathbf{a}' \cdot \mathbf{x} + b'$, let $h - h' \equiv x_{k+1} = (\mathbf{a} - \mathbf{a}') \cdot \mathbf{x} + (b - b')$. For a non-singular affine transformation, $A : \mathbb{R}^k \to \mathbb{R}^k$, let $A \circ h$ denote the hyperplane $x_{k+1} = \mathbf{a} \cdot (A \cdot \mathbf{x}) + b$.

Definition 5.3 Two sets of hyperplane $\mathcal{H}, \mathcal{H}'$ are called *equivalent* if one can map an ε-approximation to $\mathcal{H}$ into an ε-approximation to $\mathcal{H}'$, and vice versa, and the running time is linear in the complexity of the approximation.

The following claim can be easily verified:

Claim 5.4 *A set of hyperplanes $\mathcal{H}$ is equivalent to $A \circ \mathcal{H} - h$, where h is a hyperplane in $\mathbb{R}^{k+1}$ and $A : \mathbb{R}^k \to \mathbb{R}^k$ is a non-singular affine transformation.*

Lemma 5.5 *Given a set $\mathcal{H}$ of hyperplanes in $\mathbb{R}^{k+1}$ one can compute in linear time an equivalent set of hyperplanes $\mathcal{H}''$ which $(c_k, 1)$-normalized, where c_k is a positive constant depending only on k.*

Proof: Let $U = \{\mathbf{a}_1, \ldots, \mathbf{a}_n\} \subseteq \mathbb{R}^k$ be the coefficients of the hyperplanes of $\mathcal{H}$. Let A be the matrix of a *linear* transformation so that $Q = UA$ is a c_k-fat set of points so that $\mathcal{Q} = \mathcal{CH}(Q)$ have the following properties: (i) there is a ball b of radius $> c_k$ centered in a point $\mathbf{c} \in \mathbb{R}^k$, so that $b \subseteq \mathcal{Q}$, where c_k are constants that depends only on the dimension k, (ii) There is a ball B centered in $\mathbf{c}$ of radius 1 that contains $\mathcal{Q}$. Such a transformation can be computed in $O(n)$ using the algorithm of Lemma 2.4.

Let $\mathcal{H}' = A \circ H - h$, where $h \equiv x_{k+1} = \mathbf{c} \cdot \mathbf{x}$. Using linear programming, one can compute in linear time the point $\mathbf{x}_0$, where $I_{H'}(\mathbf{x})$ is minimized. Finally, let $T(\mathbf{x}) = \mathbf{x} - \mathbf{x}_0$, and let $\mathcal{H}'' = T \circ \mathcal{H}'$.

It is now straightforward to verify that $\mathcal{H}''$ is an $(c_k, 1)$ normalized set of hyperplanes. Indeed, since the coefficient vectors of the hyperplanes of $\mathcal{H}''$ are centered at origin, and their convex hull contain a ball of radius $> c_k$, and it is contained inside a ball of radius 1, it follows that for any vector $\mathbf{x} \in \mathbb{R}^k$, we know that there exists an $hin\mathcal{H}''$ such that $\mathbf{x} \cdot \mathbf{a} > c_k |\mathbf{x}|$, where $h \equiv x_{k+1} = \mathbf{a} \cdot \mathbf{x} + b$. Furthermore, $|h(\mathbf{x}) - h(0)| = |\mathbf{x} \cdot \mathbf{a}| \leq |\mathbf{x}||\mathbf{a}| \leq |\mathbf{x}|$. ■

Lemma 5.6 *Let $\mathcal{H}$ be a set of (α, β)-normalized hyperplanes in $\mathbb{R}^{k+1}$, and let $\mathcal{C}$ be the hypercube of side length $12M/(\varepsilon\alpha)$ centered at the origin, where $M = I(0)$ is the minimal extent of $\mathcal{H}$. We compute in time $O(n + 1/\varepsilon^{3k/2} \log 1/\varepsilon)$ two piecewise-linear functions $\mathcal{U}^\varepsilon, \mathcal{L}^\varepsilon$ (as defined above) of complexity $O(1/\varepsilon^k \log 1/\varepsilon)$ that satisfy (C1) and (C2) for points inside $\mathcal{C}$.*

Proof: Let $\mathcal{C}_0$ be the cube of side $M_0 = 12M/\alpha$ centered at the origin, and let $\mathcal{G}_0$ be a uniform grid in $\mathcal{C}_0$ with distance $\delta_0 = \varepsilon M/(12(k+1)\beta)$ between grid points. For each point z of $\mathcal{G}_0$, we compute $z^+ = \mathcal{U}(z)$ and $z^- = \mathcal{L}(z)$. Next, each cell of G_0 (which is a subcube of side length δ_0) is partitioned into simplices in some canonical way. This results into a simplicial decomposition Ξ of $\mathcal{C}_0$ into $O(1/\varepsilon^k)$ simplices. We lift each such simplex $\Delta = \mathrm{conv}(z_1, \ldots, z_{k+1})$ to two k-simplices $\Delta^+ = \mathrm{conv}(z_1^+, \ldots, z_{k+1}^+)$ and $\Delta^- = \mathrm{conv}(z_1^-, \ldots, z_{k+1}^-)$ in $\mathbb{R}^{k+1}$. We claim that Δ^+, Δ^- approximate $\mathcal{U}$ and $\mathcal{L}$ inside Δ.

By the convexity of the upper (reps. lower) envelope of $\mathcal{H}$, $\Delta^+(\mathbf{x}) \geq \mathcal{U}(\mathbf{x})$ and $\Delta^-(\mathbf{x}) \leq \mathcal{L}(\mathbf{x})$ for all $\mathbf{x} \in \Delta$. Thus, it remains to bound the error incurred by this approximation. The error for each of Δ^+ andΔ^- is clearly bounded by $\beta\,\mathrm{diam}(\Delta)$ because $\mathcal{H}$ is (α,β)-normalized. Thus, for any $\mathbf{x} \in \Delta$,

$$\Delta^+(\mathbf{x}) \;\leq\; \mathcal{U}(\mathbf{x}) + \beta\,\mathrm{diam}(\Delta) \leq \mathcal{U}(\mathbf{x}) + \beta\frac{\sqrt{k+1}\varepsilon M}{12(k+1)\beta} \leq \mathcal{U}(\mathbf{x}) + \frac{\varepsilon M}{12\sqrt{k+1}} \leq \mathcal{U}(\mathbf{x}) + \frac{\varepsilon}{2}I(\mathbf{x})$$

since $I(\mathbf{x}) \geq M$. Similarly, we can prove that $\Delta^-(\mathbf{x}) \geq \mathcal{L}(\mathbf{x}) - \varepsilon I(\mathbf{x})/2$.

We continue in the same way as follows: Let $\mathcal{C}_i$ be the cube centered at the origin of side $M_i = 12 \cdot 2^{i-1}M/\alpha$, for $i = 1,\ldots,\lceil \log 1/\varepsilon \rceil$. Note, that for $\mathbf{x} \in \mathcal{C}_i \setminus \mathcal{C}_{i-1}$, we have $I(\mathbf{x}) \geq M_i\alpha/2$, since $\mathcal{H}$ is (α,β)-normalized. Let $\mathcal{G}_i$ be a uniform grid in $\mathcal{C}_i$ with distance $\delta_i = \varepsilon 2^{i-1}M/(12(k+1)\beta)$ between the grid points. For each point $z \in G_i \cap (\mathcal{C}_i \setminus \mathcal{C}_{i-1})$, we compute the $\mathcal{U}(z)$ and $\mathcal{L}(z)$ and compute approximations of $\mathcal{U}$ and $\mathcal{L}$ inside $\mathcal{C}_i \setminus \mathcal{C}_{i-1}$ as earlier. Arguing as above, we conclude that the resulting functions satisfy (C1) and (C2) inside $\mathcal{C}_i \setminus \mathcal{C}_{i-1}$.

Thus, the resulting piecewise-linear functions (defined by the lifted simplices of the resulting simplicial decomposition of $\mathcal{C}$), ε-approximate the extent of $\mathcal{H}$, and they can be computed in $O(n/\varepsilon^k \log 1/\varepsilon)$ time, since $\mathcal{U}(z), \mathcal{L}(z)$, for any point $z \in \mathbb{R}^k$ can be computed in $O(n)$ time.

The running time can be further improved, by first applying the algorithm of Theorem 3.11 to compute an $\varepsilon/3$ approximation $\mathcal{H}'$ to $\mathcal{H}$, and the applying the above algorithm to compute an $\varepsilon/3$ approximation to $\mathcal{H}'$. The resulting set is clearly an ε-approximation to $\mathcal{H}$, and the overall running time is $O(n + 1/\varepsilon^{3k/2} \log 1/\varepsilon)$. ■

We omit the straightforward details of computing $\mathcal{U}^\varepsilon$ and $\mathcal{L}^\varepsilon$ in the exterior of $\mathcal{C}$. This completes the proof of Theorem 5.1.

Finally, we observe the the curve μ intersects only $O(1/\varepsilon)$ cells in $\mathcal{C}_0$ and in $\mathcal{C}_i \setminus \mathcal{C}_{i-1}$ for $i \geq 1$. Therefore, the restriction of $\mathcal{U}^\varepsilon$ and $\mathcal{L}^\varepsilon$ to μ has complexity only $O(\log(1/\varepsilon)/\varepsilon)$. Putting everything together and omitting all further details, we obtain the following.

Theorem 5.7 *Given a set of P of n points in $\mathbb{R}^d$ with motion of degree k, one can maintain an ε-approximate smallest bounding rectangle of P, in a total of $O((n/\varepsilon)\log(1/\varepsilon))$ time, whose combinatorial structure changes $O(\log(1/\varepsilon)/\varepsilon)$ times; the constant of proportionality depends on k.*

Using the above theorem in conjunction with Theorem 3.11, we can expedite the time spent in updating the rectangle.

Theorem 5.8 *Given a set of P of n points in $\mathbb{R}^d$ with motion of degree k, one can maintain an ε-approximation $B^\varepsilon(t)$ of the smallest enclosing rectangle of P, whose combinatorial structure changes $O(\log(1/\varepsilon)/\varepsilon)$ times; the constant of proportionality depends on k. We can compute the new rectangle after insertion/deletion in time*

$$O\left(\frac{\log^{k+1} n}{\varepsilon^k} + \frac{1}{\varepsilon^{k+1}}\log\frac{1}{\varepsilon}\right).$$

Proof: We use the algorithm of Theorem 3.11 to maintain an $\varepsilon/2$-approximate extent to the given set of hyperplanes (with $r = 2$). For the extent stored in the root of our tree, we use the above grid approximation, to compute the extent of the points along the curve μ, this results in $O(1/\varepsilon \log(1/\varepsilon))$ events, and since the number of hyperplanes stored in the root of the tree is $O(1/\varepsilon^k)$, the bound follows. ■

Remark 5.9 The above construction be simplified using duality. See [HV00] for details. However, their construction those not yield a result similar to Theorem 5.7, as their regions of approximation are not as well behaved as the regions of the exponential grid of Lemma 5.6.

Acknowledgments

The authors thank Imre Bárány, Jeff Erickson, Cecilia Magdalena Procopiuc, and Micha Sharir for helpful discussions.

References

[ABO$^+$89] A. Aggarwal, H. Booth, J. O'Rourke, S. Suri, and C.K. Yap. Finding minimal convex nested polygons. *Inform. Comput.*, 83(1):98–110, October 1989.

[AEG98] P.K. Agarwal, J. Erickson, and L.J. Guibas. Kinetic BSPs for intersecting segments and disjoint triangles. In *Proc. 9th ACM-SIAM Sympos. Discrete Algorithms*, pages 107–116, 1998.

[AGHV97] P.K. Agarwal, L.J. Guibas, J. Hershberger, and E. Veach. Maintaining the extent of a moving point set. In *Proc. 5th Workshop Algorithms Data Struct.*, volume 1272 of *Lecture Notes Comput. Sci.*, pages 31–44. Springer-Verlag, 1997.

[AM94] P.K. Agarwal and J. Matoušek. On range searching with semialgebraic sets. *Discrete Comput. Geom.*, 11:393–418, 1994.

[AS00] N. Alon and J.H. Spencer. *The probabilistic method.* Wiley Inter-Science, 2nd edition, 2000.

[Ata85] M. J. Atallah. Some dynamic computational geometry problems. *Comput. Math. Appl.*, 11(12):1171–1181, 1985.

[BGH97] J. Basch, L.J. Guibas, and J. Hershberger. Data structures for mobile data. In *Proc. 8th ACM-SIAM Sympos. Discrete Algorithms*, pages 747–756, 1997.

[BH01] G. Barequet and S. Har-Peled. Efficiently approximating the minimum-volume bounding box of a point set in three dimensions. *J. Algorithms*, 38:91–109, 2001.

[BI76] E.M. Bronshteyn and L.D. Ivanov. The approximation of convex sets by polyhedra. *Siberian Math. J.*, 16:852–853, 1976.

[Dud74] R.M. Dudley. Metric entropy of some classes of sets with differentiable boundaries. *J. Approx. Theory*, 10(3):227–236, 1974.

[Gär95] B. Gärtner. A subexponential algorithm for abstract optimization problems. *SIAM J. Comput.*, 24:1018–1035, 1995.

[Gui98] L.J. Guibas. Kinetic data structures — a state of the art report. In P.K. Agarwal, L. E. Kavraki, and M. Mason, editors, *Proc. 5th Workshop Algorithmic Found. Robot.*, pages 191–209. A. K. Peters, Wellesley, MA, 1998.

[HV00] S. Har-Peled and K.R. Varadarajan. Approximate shape fitting via linearization. submitted, 2000.

[Mac51] A.M. Macbeath. A compactness theorem for affine equivalence-classes of convex regions. *Canad. J. Math.*, 3:54–61, 1951.

[OvL81] M. H. Overmars and J. van Leeuwen. Maintenance of configurations in the plane. *J. Comput. Syst. Sci.*, 23:166–204, 1981.

[SA95] M. Sharir and P.K. Agarwal. *Davenport-Schinzel Sequences and Their Geometric Applications.* Cambridge University Press, New York, 1995.

[SJLL00] S. Saltenis, C.S. Jensen, S. Leutenegger, and M. Lopez. Indexing the positions of continuously moving objects. In *ACM SIGMOD Int. Conf. on Manag. of Data*, pages 331–342, 2000.

[YY85] A. C. Yao and F. F. Yao. A general approach to D-dimensional geometric queries. In *Proc. 17th Annu. ACM Sympos. Theory Comput.*, pages 163–168, 1985.