Faster Core-Set Constructions and Data-Stream Algorithms in Fixed Dimensions*

Timothy M. Chan†

July 15, 2005

Abstract

We speed up previous $(1+\varepsilon)$-factor approximation algorithms for a number of geometric optimization problems in fixed dimensions: diameter, width, minimum-radius enclosing cylinder, minimum-width enclosing annulus, minimum-width enclosing cylindrical shell, etc. Linear time bounds were known before; we further improve the dependence of the "constants" in terms of ε.

We next consider the data-stream model and present new $(1+\varepsilon)$-factor approximation algorithms that need only constant space for all of the above problems in any fixed dimension. Previously, such a result was known only for diameter.

Both sets of results are obtained using the core-set framework recently proposed by Agarwal, Har-Peled, and Varadarajan.

Keywords: Approximation algorithms; Geometric optimization problems; Data streams

1 Introduction

1.1 Results

In the first part of the paper, we present results that lead to faster approximation algorithms for various fundamental geometric optimization problems in fixed dimensions, with applications to shape fitting, computational metrology, statistical analysis, and other areas. It has already been established that a $(1+\varepsilon)$-factor approximation for many such problems can be computed efficiently in $O(n+1/\varepsilon^c)$ time for some constant c depending on the problem and the dimension. Here, we are interested in obtaining the best time bound, i.e., determining the smallest c possible.

The optimization problems involve computing the objects listed below for a given set of n points in $\mathbb{R}^d$, where d is assumed to be a constant, unless stated otherwise. (Big-oh notation hides factors that depend on d but not ε. A model of computation that supports the floor function and square roots in constant time is assumed.) We summarize the old and new running times in each case.

1. The *diameter*, i.e., the maximum distance over all pairs of points. There are at least two "obvious" approximation algorithms, one running in $O(n/\varepsilon^{(d-1)/2})$ time [5] (by rounding directions) and one running in $O(n+1/\varepsilon^{2(d-1)})$ [9] (by rounding points). The author [14] has observed that a combination of the two yields an $O(n+1/\varepsilon^{3(d-1)/2})$ time bound. He has also given two

*This work was supported in part by an NSERC Research Grant.

†School of Computer Science, University of Waterloo, Waterloo, ON N2L 3G1, Canada, tmchan@uwaterloo.ca

simple algorithms running in $O(n+1/\varepsilon^{d-1/2})$ time; the time bound can be reduced slightly to $O((n+1/\varepsilon^{d-5/3})\,\text{polylog}\,(1/\varepsilon))$ in theory, using advanced data structures.

Here, we present a new (but similar) simple algorithm that runs in $O(n+1/\varepsilon^{d-3/2})$ time. The running time can be reduced to $O((n+1/\varepsilon^{d-2})\log(1/\varepsilon))$ if efficient planar-point-location data structures are available.

2. The *width*, i.e., the minimum width over all enclosing slabs (regions between two parallel hyperplanes). Duncan *et al.* [18] have described an $O(n/\varepsilon^{(d-1)/2})$-time approximation algorithm; the technique by Barequet and Har-Peled [9] can be used in combination to yield an $O(n+1/\varepsilon^{3(d-1)/2})$ time bound [14]. (Slight improvements were known in low dimensions using more complicated data structures; for example, for $d=3$, the author [14] and Agarwal *et al.* [2] have respectively obtained an $O((n+1/\varepsilon)\log(1/\varepsilon))$ and an $O(n+(1/\varepsilon)\log^2(1/\varepsilon))$ bound.)

   We give a new algorithm that runs in $O(n+1/\varepsilon^{d-1})$ time. With advanced data structures, the running time can in theory be reduced to $O((n+1/\varepsilon^2)\,\text{polylog}\,(1/\varepsilon))$ for $d=4$, and to $O((n+1/\varepsilon^{d-2})\log(1/\varepsilon))$ for $d>4$.

3. An enclosing *cylinder* of the minimum radius. Agarwal, Aronov, and Sharir [1] gave an $O(n/\varepsilon^2)$-time approximation algorithm for $d=3$. The author [14] gave an $O(n+1/\varepsilon^{3(d-1)/2})$-time algorithm for all fixed d (omitting minor theoretical improvements).

   We improve the time bound to $O(n+1/\varepsilon^{d-1})$ (again omitting minor theoretical improvements).

4. An enclosing *spherical shell/annulus* (region between two concentric spheres) of the minimum width. The author [14] was the first to obtain linear-time approximation algorithms in all fixed dimensions $d\geq 2$. Agarwal, Har-Peled, and Varadarajan [4] have given the best time bound for sufficiently large (but fixed) d: $O(n+1/\varepsilon^{3d})$.

   We improve their time bound to $O(n+1/\varepsilon^{2d})$, which beats previous results [14] for all $d\geq 4$.

5. Agarwal, Har-Peled, and Varadarajan [4] have also given linear-time approximation algorithms for finding an enclosing *cylindrical shell* (region between two co-axial cylinders) of the minimum width, and various *kinetic* versions of the above problems (approximating the width of a set of moving points).

   We obtain similar improvements for all these problems, by reducing time bounds of the form $O(n+1/\varepsilon^c)$ to $O(n+1/\varepsilon^{2c/3})$.

6. Agarwal, Har-Peled, and Varadarajan [4] have also presented dynamic data structures for the above approximation problems with polylogarithmic update time. Our results imply smaller degrees in the polylogarithmic factors.

In addition, Agarwal, Har-Peled, and Varadarajan [4] have given data structures that support insertions and use remarkably little space. The need to handle massive data has generated considerable attention recently to the *data-stream* model [24, 25, 29], where the input is too big to be stored, only one pass over the input is possible, and algorithms can maintain only a bounded amount of information at any time. Exact algorithms with sublinear space are not possible for any of the above problems, but allowing approximations, Agarwal *et al.*'s result has demonstrated that polylogarithmic space is sufficient for many geometric optimization problems. For example, for the width problem, there is a data stream algorithm that uses $O((1/\varepsilon^{(d-1)/2})\log^d n)$ space and spends $O(1/\varepsilon^{d-1})$ amortized time per data point.

The remaining question is whether some of the $\log n$ factors can be removed, or more boldly, whether constant space is possible. For diameter, one of the earlier algorithms [5] uses just $O(1/\varepsilon^{(d-1)/2})$ space in the data-stream setting (see also [19, 25]). For the width, no previous algorithms yield constant-space solutions; the problem is apparently open even in two dimensions, as Hershberger and Suri [25] only gave bounded-space results for the case when the diameter-to-width ratio is bounded. (For more related work, see also [3] for multiple-pass algorithms that exploit graphics hardware, and [15, 19] for algorithms under the stronger "sliding-window" model.)

In the second part of the paper, we answer the question in the affirmative by obtaining one-pass, data-stream algorithms for all of the above problems using an amount of storage independent of n. For the width and the minimum-radius cylinder problem, for example, in one of our methods, the space bound is $O([(1/\varepsilon)\log(1/\varepsilon)]^{d-1})$, the time to process each point is just $O(1)$, and an approximate answer can be reported at any time in $O(1/\varepsilon^{d-1})$ time. In a refinement of the method, the space bound is lowered by a factor of about $1/\sqrt{\varepsilon}$, at the expense of a slightly larger update time. (Note that our results even improve known data structures [27, 30] for approximate two-dimensional width in the traditional model; these structures required linear space and logarithmic insertion time.) For the spherical-shell problem, the space bound is $O([(1/\varepsilon)\log(1/\varepsilon)]^{2d})$, or smaller by a factor of about $1/\varepsilon$ in the refined version.

1.2 The Core-Set Framework

An approach to developing approximation algorithms, taken by Barequet and Har-Peled [9], Agarwal, Har-Peled, and Varadarjan [4] and others, is to identify a constant-size subset of the input that approximates the entire input (as stated in the definition below). Once such a subset is found, we can simply return a solution to the subset by a direct algorithm, exact or approximate. This approach has proved successful also for high-dimensional geometric problems [7, 8], but we focus only on the case of fixed dimensions.

Definition 1.1 Given a one-argument "measure" $\mu(\cdot)$ that maps every point set to a nonnegative real number, we say that a subset $R \subseteq P$ is an *ε-core-set of P* if $\mu(P) \geq \mu(R) \geq (1-\varepsilon)\mu(P)$.

The diameter is relatively easy to approximate in linear time, but problems like the width are less trivial. A key idea behind Agarwal, Har-Peled, and Varadarajan's approach [4] is to solve an apparently harder problem (formalized in the next definition), namely, to approximate the width along all directions at the same time. The advantage in considering this more general problem, as we will see later, is that we can "stretch" (i.e., nonuniformly scale) the point set without affecting the quality of the approximation.

Definition 1.2 Given a two-argument measure $\mu(\cdot,\cdot)$ that maps a point set and a vector to a nonnegative real number, we say that a subset $R \subseteq P$ is an *ε-core-set of P (over a domain S)* if $\mu(P,x) \geq \mu(R,x) \geq (1-\varepsilon)\mu(P,x)$ for all x (in S).

For the width problem, the particular measure of interest is the following:

Definition 1.3 The *extent measure* is defined as $w(P,x) = \max_{p,q\in P}(p-q)\cdot x$ $(x \in \mathbb{R}^d)$.

As it turns out, considering the extent measure alone is enough to yield all the results mentioned: Agarwal, Har-Peled, and Varadarjan [4] have observed that a core-set for the extent measure is also a core-set for the width, the minimum-cylinder radius, the minimum bounding-box volume, and other problems determined solely by the convex hull. (Core-sets under the extent measure are also called ε-*kernels* in their paper.) They have also shown how core-sets for the extent measure lead to core-sets for more complicated measures like the minimum-spherical-shell and minimum-cylindrical-shell width, by lifting the given points to a higher dimension.

For some of the proofs to come, it is more convenient to work with a simpler "one-sided" function in place of the extent:

Observation 1.4 *Define $\overline{w}(P, x) = \max_{p \in P} p \cdot x$. If R is an ε-core-set of P for the extent measure, then $\overline{w}(R, x) \geq \overline{w}(P, x) - \varepsilon w(P, x)$ for all x. Conversely, if $\overline{w}(R, x) \geq \overline{w}(P, x) - \varepsilon w(P, x)$ for all x in a domain S that is symmetric (i.e., $S = -S$), then R is a (2ε)-core-set of P for the extent measure over S.*

Proof: Both parts follow from the fact that

$$w(P, x) - w(R, x) \;=\; [\overline{w}(P, x) - \overline{w}(R, x)] + [\overline{w}(P, -x) - \overline{w}(R, -x)].$$

(Thus, $w(P, x) - w(R, x) \leq \varepsilon w(P, x)$ implies $\overline{w}(P, x) - \overline{w}(R, x) \leq \varepsilon w(P, x)$. Conversely, $\overline{w}(P, x) - \overline{w}(R, x) \leq \varepsilon w(P, x)$ and $\overline{w}(P, -x) - \overline{w}(R, -x) \leq \varepsilon w(P, x)$ imply $w(P, x) - w(R, x) \leq 2\varepsilon w(P, x)$.) $\square$

1.3 The New Ideas

The first part of the paper (Section 2) follows closely the core-set methods by Agarwal, Har-Peled, and Varadarajan [4], although we keep our presentation mostly self-contained (except for the reference to a subroutine by Barequet and Har-Peled [9]). Modulo a simplification in one step of one of Agarwal *et al.*'s methods, the main new idea is the use of *discrete Voronoi diagrams* and related constructs, which can identify the (exact) nearest neighbors to all grid points for a given grid point set. This notion is different from "approximate Voronoi diagrams" [6, 21] and does not seem to have received as much attention in computational geometry, although the problem has been considered by Breu *et al.* [12] (see also [31]) and has applications in image processing.

The second part of the paper (Section 3) is mostly independent of the first and uses a new core-set construction that works for data streams. Unlike in previous data-stream algorithms such as Agarwal *et al.*'s [4], which use a hierarchical merging method (the so-called "logarithmic method" of Bentley and Saxe [10]), we suggest a more efficient, direct method based on a doubling strategy, which exploits the "monotonicity" of the insertion-only setting and allows the error to be bounded by a geometric series. The main idea can be illustrated by a very simple data-stream algorithm for the minimum-radius cylinder problem that achieves a constant approximation factor, even in non-constant dimensions.

2 Faster Core-Set Constructions

2.1 Discrete Upper Envelopes and Discrete Voronoi Diagrams

We begin with the main subroutine that accounts for all the improved results of this section. The subproblem is stated in the theorem below and involves computing a discrete version of an upper envelope of n hyperplanes. We use the following notation in this subsection: $[E]$ denotes the set of integers $\{1, \ldots, E\}$ (or more generally, E uniformly spaced reals), and p_i denotes the i-th coordinate of a given point p.

Theorem 2.1 *Fix $\delta > 0$ and suppose $E^\delta \leq F \leq E$. Given an n-point set $P \subseteq [E]^{d-1} \times \mathbb{R}$, we can compute*

$$q[\xi] = \text{a point } p \in P \text{ that maximizes } p_1\xi_1 + \cdots + p_{d-1}\xi_{d-1} + p_d$$

for every $\xi \in [F]^{d-1}$ in total time $O(n + E^{d-2}F)$ or $O((n + E^{d-3}F^2)\log E)$.

In other words, the above theorem lets us evaluate the upper envelope of the hyperplanes of the form $\{\xi \mid \xi_d = p_1\xi_1 + \cdots + p_{d-1}\xi_{d-1} + p_d\}$ at a set of grid points. Dually, this corresponds to computing the extreme points of the point set P along various "grid directions". The trivial algorithm to solve this problem requires time $O(nF^{d-1})$, which could be as big as $O(E^{d-1}F^{d-1})$.

Proof: The algorithm proceeds as follows:

1. for $i \in [E]$ do
2.   for $\xi_2, \ldots, \xi_{d-1} \in [F]$ do
3. $r[i, \xi_2, \ldots, \xi_{d-1}] =$ a point $p \in P$ with $p_1 = i$ that maximizes $p_2\xi_2 + \cdots + p_{d-1}\xi_{d-1} + p_d$
4. for $\xi_2, \ldots, \xi_{d-1} \in [F]$ do
5.   for $\xi_1 \in [F]$ do
6. $q[\xi_1, \ldots, \xi_{d-1}] =$ a point $p \in \{r[i, \xi_2, \ldots, \xi_{d-1}] \mid i \in [E]\}$ that maximizes $p_1\xi_1 + p_2\xi_2 + \cdots + p_{d-1}\xi_{d-1} + p_d$

Correctness is self-evident: for points $p \in P$ with the same first coordinate, maximizing $p_1\xi_1 + \cdots + p_{d-1}\xi_{d-1} + p_d$ is equivalent to maximizing $p_2\xi_2 + \cdots + p_{d-1}\xi_{d-1} + p_d$.

Observe that for a fixed $i \in [E]$, the loop at lines 2–3 is a $(d-1)$-dimensional subproblem of the same type, on a subset $\{p \in P \mid p_1 = i\}$, say, of size n_i. On the other hand, for fixed $\xi_2, \ldots, \xi_{d-1} \in [F]$, the loop at lines 5–6 is a 2-dimensional subproblem of the same type on a subset of size E—just think of $p_2\xi_2 + \cdots + p_{d-1}\xi_{d-1} + p_d$ collectively as the second coordinate. We can therefore implement the above algorithm with a running time upper-bounded by the solution to the following recurrence:

$$T_d(n) = \sum_{i=1}^{E} T_{d-1}(n_i) + F^{d-2}T_2(E) + O(EF^{d-2} + F^{d-1}),$$

where $\sum_{i=1}^{E} n_i = n$.

The base case $d = 2$ can be handled by explicitly constructing the upper envelope, which in the dual corresponds to a planar convex (lower) hull. The first coordinates are from $[E]$ and can be sorted in $O(n + E^\delta)$ time by using a radix-sort with $\lceil 1/\delta \rceil$ passes [16] (where in each pass, we use a counting sort to handle integers in the range $[E^\delta]$). We may remove all but the lowest point on each vertical grid line. Graham's scan [11] can then construct the lower hull in linear time. The answers $q[x]$ for all $x \in [F]$ can be computed by another scan. Therefore, $T_2(n) = O(n + F)$. By induction (and the assumption that $F \leq E$), one can show that the above recurrence solves to $T_d(n) = O(n + E^{d-2}F)$.

We can also make $d = 3$ a base case by a more complicated approach: Again, the upper envelope can be constructed explicitly, this time by an $O(n \log n)$-time 3-dimensional convex hull algorithm [11]. The answer $q[x]$ for each $x \in [F]^2$ can be obtained by a point location query [11]. This gives $T_3(n) = O((n + F^2)\log n) = O((n + F^2)\log E)$ (since a naive preprocessing can ensure that $n \leq E^2$, by removing the lowest point in each vertical grid line in the dual). The recurrence now solves to $T_d(n) = O((n + E^{d-3}F^2)\log E)$. □

By the standard lifting map [11], Voronoi diagrams of point sets transform to upper envelopes of hyperplanes. Hence, one application of Theorem 2.1 is the construction of a discrete Voronoi diagram:

Corollary 2.2 *Given n sites in $[E]^k \times \mathbb{R}^{d-k}$, we can compute the nearest neighbor to each grid point in $[F]^k \times \{0\}^{d-k}$ in total time $O(n + E^{k-1}F)$ or $O((n + E^{k-2}F^2)\log E)$.*

In particular, given n sites in $[E]^d$, we can compute the nearest neighbor to each grid point in $[E]^d$ in total time $O(E^d)$.

Proof: Given $\xi \in [F]^k \times \{0\}^{d-k}$, a point $p \in P$ minimizing $\|p - \xi\|$ maximizes $2p_1\xi_1 + \cdots + 2p_k\xi_k - \|p\|^2$, so the first part is an special case of the theorem in dimension $k + 1$. The second part is a further special case with $k = d$ and $F = E$ (where the simpler Graham-scan version suffices). □

The second part of the corollary was originally obtained by Breu *et al.* [12] for $d = 2$. The problem arises from image-processing applications, specifically in the computation of the Euclidean distance transform (the identification of the nearest neighbor to each pixel). Our algorithm is similar to Breu *et al.*'s second algorithm [12], although their description appears longer and does not explicitly invoke Graham's scan. They also mentioned that their algorithm can be generalized to any fixed dimension d, although details were not given.

The dimension-recursion approach behind our algorithm is also similar to some of the author's approximation algorithms for the diameter problem [14] (specifically, the fourth and the fifth algorithm).

2.2 Warm-Up: Diameter

We now demonstrate the usefulness of Theorem 2.1 in the design of geometric approximation algorithms by considering the diameter problem. The algorithm we use is essentially the same as the third diameter algorithm in [14]: the idea is to round both the input points and the space of all directions.

As is well-known, one can generate a small set of vectors that approximate all directions well. A particular, simple construction is described in the observation below; here, an *ε-grid* refers to a uniform grid, where the diameter of each grid cell is ε.

Observation 2.3 *Suppose we have a box B containing the origin o, where the boundary ∂B is of distance at least 1 from o. Given an ε-grid over ∂B, for any vector x, there exists a grid point ξ such that the angle $\angle(\xi, x)$ between $\overrightarrow{o\xi}$ and x is at most* $\arccos(1 - \varepsilon^2/8) = O(\varepsilon)$.

Proof: By scaling, we may assume that $x \in \partial B$. There exists a grid point ξ with $\|\xi - x\| \leq \varepsilon/2$, implying that $2\xi \cdot x \geq \|\xi\|^2 + \|x\|^2 - \varepsilon^2/4 \geq 2\|\xi\|\,\|x\| - \varepsilon^2/4 \geq 2\|\xi\|\,\|x\|(1 - \varepsilon^2/8)$. The observation follows because $\cos\angle(\xi, x) = \xi \cdot x/\|\xi\|\,\|x\|$. □

Recall that a number is a $(1+\varepsilon)$*-factor approximation* of another number if the ratio of the larger to the smaller number is at most $1+\varepsilon$.

Theorem 2.4 *The diameter of* n *points in* $\mathbb{R}^d$ *can be approximated to within a factor of* $1+\varepsilon$ *in* $O(n+1/\varepsilon^{d-3/2})$ *or* $O((n+1/\varepsilon^{d-2})\log(1/\varepsilon))$ *time.*

Proof: Let P be the given point set. We first compute a constant-factor approximation Δ of the diameter in linear time: for example, say $o \in P$ (by translation) and let v be the farthest point from o; then the diameter is between $\|v\|$ and $2\|v\|$.

The rest of the algorithm is as follows:

1. round each $p \in P$ to a point p' on an $(\varepsilon\Delta)$-grid
2. let Ξ be the points of a $\sqrt{\varepsilon}$-grid over $\partial[-1,1]^d$
3. return the farthest pair among $\{(p_\xi, q_\xi)\}_{\xi\in\Xi}$, where $p_\xi, q_\xi \in P$ maximize $(p'_\xi - q'_\xi)\cdot\xi$

The running time is dominated by that of line 3. Observe that this step (maximizing $p'\cdot\xi$ and minimizing $q'\cdot\xi$ for each point ξ on $2d$ grids of dimension $d-1$) reduces precisely to the discrete envelope problem in Theorem 2.1. Here, $E = O(1/\varepsilon)$ and $F = O(1/\sqrt{\varepsilon})$, so the claimed time bounds follow.

For the analysis, take any pair $p, q \in P$. By Observation 2.3, there is a grid point $\xi \in \Xi$ with $\angle(\xi, p'-q') \le \arccos(1-\varepsilon/8)$. Then

$$\begin{aligned}(p'_\xi - q'_\xi)\cdot\xi \ \ge\ (p'-q')\cdot\xi \quad &\Longrightarrow\quad \|p'_\xi - q'_\xi\|\,\|\xi\| \ \ge\ \|p'-q'\|\,\|\xi\|(1-\varepsilon/8)\\ &\Longrightarrow\quad \|p_\xi - q_\xi\| + \varepsilon\Delta \ \ge\ (\|p-q\| - \varepsilon\Delta)(1-\varepsilon/8).\end{aligned}$$

(The first implication is due to the Cauchy-Schwarz inequality. The last implication follows because by definition of rounding, $\|p-p'\| \le \varepsilon\Delta/2$ for every $p \in P$.)

Thus, $\max_{\xi\in\Xi}\|p_\xi - q_\xi\|$ approximates $\Delta = \max_{p,q\in P}\|p-q\|$ to within a factor of $1+O(\varepsilon)$. (By readjusting ε by a constant factor, we can ensure that the approximation factor is $1+\varepsilon$.) □

2.3 First Method

Naive rounding would not work for other problems like the width. In this subsection, we follow the general approach of Agarwal, Har-Peled, and Varadarajan [4] and discuss how to find core-sets of the given point set under the extent measure; this will automatically lead to approximation algorithms for a variety of geometric problems, including the width.

The algorithm below is essentially the same as one of Agarwal *et al.*; they have proved the following theorem with a weaker $O(n+1/\varepsilon^{2(d-1)})$ running time. The idea is to stretch the given point set to make it "fat"; afterwards, the grid-rounding strategy can be applied in the same manner as in our earlier diameter algorithm. We observe that Theorem 2.1 can again speed up the computation.

Theorem 2.5 *Consider the extent measure* $w(\cdot,\cdot)$*. Given an* n*-point set in* $\mathbb{R}^d$*, we can construct a partition of* $\mathbb{R}^d$ *into* $O(1/\varepsilon^{d-1})$ *regions and an* ε*-core-set of size* $O(1)$ *over each region, in total time* $O(n+1/\varepsilon^{d-1})$.

Proof: We first compute a constant-factor approximation B of the minimum-volume bounding box in linear time by a method of Barequet and Har-Peled [9]. In their construction, B contains the given point set P, and conversely, the convex hull of P contains a translated copy of cB for some constant $c > 0$ (depending on d). By applying a translation, we may assume that B is centered at o. By applying a linear transformation, we may assume that $B = [-1, 1]^d$. To see why core-sets are preserved under linear transformations, observe that if $M(R)$ is an ε-core-set of $M(P)$ over a region Δ for a non-singular matrix M, then R is an ε-core-set of P over $M^T(\Delta)$ (because $M(r) \cdot x \geq (1-\varepsilon)M(p) \cdot x$ is equivalent to $r \cdot M^T(x) \geq (1-\varepsilon)p \cdot M^T(x)$).

Thus, we may assume that $P \subseteq [-1, 1]^d$ and that the convex hull of P contains a hypercube of side length $2c$. In particular, $w(P, x) \geq 2c$ for all unit vectors x. The rest of the algorithm is as follows:

1. round each $p \in P$ to a point p' on an ε-grid over $[-1, 1]^d$
2. let Ξ be all cell centers of an ε-grid over $\partial[-1, 1]^d$
3. return regions $\{S_\xi\}_{\xi\in\Xi}$ and subsets of size 2 $\{\{p_\xi, q_\xi\}\}_{\xi\in\Xi}$
   where S_ξ is the cone with apex at o through the cell centered at ξ and
   $p_\xi, q_\xi \in P$ maximize $(p'_\xi - q'_\xi) \cdot \xi$

Line 3 (the computation of the p_ξ's and q_ξ's) can be handled by a discrete envelope (Theorem 2.1) as before, this time with both $E, F = O(1/\varepsilon)$, in $O(n + 1/\varepsilon^{d-1})$ time.

For the analysis, fix a vector $x \in \partial[-1, 1]^d$ lying in S_ξ. Take any pair $p, q \in P$. Since $\|\xi - x\| \leq \varepsilon/2$ and $\|p'_\xi - q'_\xi\|, \|p' - q'\| \leq 2\sqrt{d}$,

$$\begin{aligned}(p'_\xi - q'_\xi) \cdot \xi \geq (p' - q') \cdot \xi \quad &\Longrightarrow \quad (p'_\xi - q'_\xi) \cdot x + \sqrt{d}\varepsilon \geq (p' - q') \cdot x - \sqrt{d}\varepsilon \\ &\Longrightarrow \quad (p_\xi - q_\xi) \cdot x + 2\sqrt{d}\varepsilon \geq (p - q) \cdot x - 2\sqrt{d}\varepsilon.\end{aligned}$$

(The last inequality follows because by definition of rounding, $\|p - p'\| \leq \varepsilon/2$ for every $p \in P$, and because $\|x\| \leq \sqrt{d}$.)

Thus, $w(\{p_\xi, q_\xi\}, x)$ and $w(P, x)$ differ by at most an $O(\varepsilon)$ additive error. Since $w(P, x)$ is lower-bounded by a constant $2c$ for all $x \in \partial[-1, 1]^d$, the relative error is also $O(\varepsilon)$. Therefore, $\{p_\xi, q_\xi\}$ is an $O(\varepsilon)$-core-set over $S_\xi \cap \partial[-1, 1]^d$, and by scaling, over all of S_ξ. □

2.4 Second Method

Agarwal *et al.* [4] have proposed another way to approximate the extent measure, using a single core-set of a smaller total size—$O(1/\varepsilon^{(d-1)/2})$ instead of $O(1/\varepsilon^{d-1})$. This gives better results for some problems. The approach is based on the ideas of Dudley [17] and Bronshteyn and Ivanov [13]. The running time for the construction is $O(n + 1/\varepsilon^{3(d-1)/2})$. We again show how to improve the running time using Theorem 2.1 or Corollary 2.2. In order to accomplish this, we need to simplify one step of Agarwal *et al.*'s method; incidentally, our analysis is also simpler.

Theorem 2.6 *Given an n-point set in $\mathbb{R}^d$, we can construct an ε-core-set of size $O(1/\varepsilon^{(d-1)/2})$ for the extent measure in $O(n + 1/\varepsilon^{d-3/2})$ or $O((n + 1/\varepsilon^{d-2})\log(1/\varepsilon))$ time.*

Proof: Following the proof of Theorem 2.5, we invoke Barequet and Har-Peled's bounding box method and apply a linear transformation so that the point set P is inside $[-1, 1]^d$ and $w(P, x) \geq 2c$ for all unit vectors x. The algorithm then proceeds as follows:

1. round each $p \in P$ to a point p' on an ε-grid over $[-1,1]^d$
2. let Ξ be the points of a $\sqrt{\varepsilon}$-grid over $\partial[-2,2]^d$
3. return $R = \{p_\xi\}_{\xi\in\Xi}$, where $p_\xi \in P$ minimizes $\|p'_\xi - \xi\|$

Line 3 (the computation of nearest neighbors to the grid points in Ξ) can be handled by $2d$ discrete Voronoi diagrams (Corollary 2.2), with $k = d-1$, $E = O(1/\varepsilon)$, and $F = O(1/\sqrt{\varepsilon})$, yielding the claimed time bounds.

For the analysis, take any unit vector $x \in \mathbb{R}^d$ and point $p \in P$. By applying Observation 2.3 with the origin shifted to p' (which has distance at least 1 from $\partial[-2,2]^d$), there is a point $\xi \in \Xi$ such that $\angle(\xi - p', x) \le \arccos(1-\varepsilon/8)$. Then

$$\begin{aligned} \|\xi - p'_\xi\| \le \|\xi - p'\| &\implies (\xi - p'_\xi)\cdot x \le (\xi - p')\cdot x/(1-\varepsilon/8) \\ &\implies (\xi - p'_\xi)\cdot x - 3\sqrt{d}\varepsilon/8 \le (\xi - p')\cdot x \qquad (\text{since } \|\xi - p'_\xi\| \le 3\sqrt{d}) \\ &\implies p_\xi\cdot x + \varepsilon/2 + 3\sqrt{d}\varepsilon/8 \ge p\cdot x - \varepsilon/2. \end{aligned}$$

(The last inequality follows because by definition of rounding, $\|p - p'\| \le \varepsilon/2$ for every $p \in P$, and because $\|x\| = 1$.)

So, $\overline{w}(R,x) = \max_{\xi\in\Xi} p_\xi \cdot x$ and $\overline{w}(P,x) = \max_{p\in P} p\cdot x$ differ by at most an $O(\varepsilon)$ additive error, and so do $w(R,x)$ and $w(P,x)$, by Observation 1.4. Since $w(P,x)$ is lower-bounded by a constant for all unit vectors, R is an $O(\varepsilon)$-core-set over all unit vectors, and by scaling, over all of $\mathbb{R}^d$. □

Our simplification comes from line 3: in Agarwal *et al.*'s original algorithm [4], we find for each $\xi \in \Xi$ the nearest neighbor to the convex hull boundary of the rounded points. (Such a computation would require generalized linear programming.) In a very recent paper, Yu *et al.* [32] have independently made a similar observation.

2.5 Applications and Remarks

We now indicate how Theorems 2.5 and 2.6 can be used to solve the specific geometric optimization problems mentioned in the introduction.

The canonical example is the width problem (item 2 in Section 1.1). We can apply the first method (Theorem 2.5) to obtain an approximation of the function $w(P,x)$ in the variable x by a function of $O(1/\varepsilon^{d-1})$ overall complexity. Since the width of P is the minimum of $w(P,x)$ over all unit vectors x, an approximate answer can be computed in $O(n + 1/\varepsilon^{d-1})$ time. Alternatively, we can apply the second method (Theorem 2.6) to get a single ε-core set R of $O(1/\varepsilon^{(d-1)/2})$ size for the extent measure, and then invoke Duncan *et al.*'s approximation algorithm [18] to R. The author [14] noted that with known data structures for linear programming queries, Duncan *et al.*'s algorithm on $|R|$ points requires time $O(|R|\log(1/\varepsilon) + (|R|/\varepsilon^{(d-1)/2})^{1-1/(\lfloor d/2\rfloor+1)}\,\mathrm{polylog}\,(1/\varepsilon) + (1/\varepsilon^{(d-1)/2})\,\mathrm{polylog}\,(1/\varepsilon))$. This expression is less than $O(1/\varepsilon^{d-2})$ for $d > 4$, so the total time is within $O((n + 1/\varepsilon^{d-2})\log(1/\varepsilon))$ for $d > 4$.

Since an ε-core-set under the extent measure is also an $O(\varepsilon)$-core-set under the minimum-cylinder radius, we can also solve this problem (item 3) by applying the second method (Theorem 2.6) to obtain a core set R and invoking the author's approximate minimum-cylinder algorithm on R, which runs in $O((1/\varepsilon^{(d-1)/2}|R|)$ time [14]. The overall running time is $O(n + 1/\varepsilon^{d-1})$ (which can also be marginally improved with advanced data structures).

Other problems (items 4 and 5) can be solved by taking appropriate lifting maps (via the so-called *linearization* technique), as described by Agarwal, Har-Peled, and Varadarjan [4]. For example, for the spherical-shell problem, we want the minimum of the function $f(P, x) = \max_{p,q \in P}(\|x - p\| - \|x - q\|)$ over all $x \in \mathbb{R}^d$. The related function $g(P, x) = \max_{p,q \in P}(\|x - p\|^2 - \|x - q\|^2) = \max_{p,q \in P}(-2(p-q) \cdot x + \|p\|^2 + \|q\|^2)$ is essentially the extent measure of a $(d+1)$-dimensional point set. Agarwal *et al.* have shown that an $O(\varepsilon^2)$-core-set for g is an $O(\varepsilon)$-core-set for f. Thus, we can apply the first method (Theorem 2.5) to approximate the function $f(P, x)$ by a function of overall complexity $O((1/\varepsilon^2)^d)$, and thus solve the problem in $O(n + 1/\varepsilon^{2d})$ time.

We leave open the question of whether the running time in Theorem 2.6 can be improved further, or whether a bound of $O(n+1/\varepsilon^{d-c})$ for some absolute constant c is the best one can hope for. It would also be interesting to see how well our new diameter algorithm competes with the experimentally efficient algorithms by Har-Peled [20] and Malandain and Boissonnat [28] in practice.

3 Data-Stream Algorithms

3.1 Warm-Up: Constant Factor for Minimum Cylinder

We now consider algorithms under the data-stream model. To illustrate the main difficulties, we start with the minimum-radius cylinder problem (which includes two-dimensional width as a special case) and consider constant-factor approximation algorithms first.

There is a very simple constant-factor algorithm, noted by Agarwal, Aronov, and Sharir [1] (later generalized by Barequet and Har-Peled [9] and Har-Peled and Varadarajan [22]). The algorithm just picks an arbitrary input point, say, the origin o (without loss of generality), then finds the farthest point v to o, and returns the farthest distance to the line $\overleftrightarrow{ov}$.

Let $\mathrm{Rad}(P)$ denote the minimum radius of all cylinders enclosing P, and let $d(p, \ell)$ denote the distance between point p and line ℓ. The following observation immediately implies an upper bound of 4 on the approximation factor for the above algorithm (since $\|p\| \le \|v\|$ for all input points p).

Observation 3.1

$$d(p, \overleftrightarrow{ov}) \;\le\; 2\left(\frac{\|p\|}{\|v\|} + 1\right)\mathrm{Rad}(\{o, v, p\}).$$

Proof: In the triangle $\triangle ovp$, construct the altitudes h_1, h_2, h_3 at the respective bases ov, op, and pv. Since $h_1\|v\| = h_2\|p\| = h_3\|p - v\|$ and $h_1 = d(p, \overleftrightarrow{ov})$,

$$2\,\mathrm{Rad}(\{o, v, p\}) \;=\; \min\{h_1, h_2, h_3\} \;=\; h_1 \min\left\{1, \frac{\|v\|}{\|p\|}, \frac{\|v\|}{\|p - v\|}\right\} \;\ge\; d(p, \overleftrightarrow{ov})\left(\frac{\|v\|}{\|p\| + \|v\|}\right).$$

□

Unfortunately, the above algorithm requires two passes, one to find the farthest neighbor v to o, and one to find the farthest distance to $\overleftrightarrow{ov}$. Nevertheless, we show that a variant of this algorithm, which maintains an approximate candidate for v on-line, can be made to work in the data-stream setting, albeit with a larger approximation factor:

Theorem 3.2 *Given a stream of points in $\mathbb{R}^d$ (where d is not necessarily a constant), we can maintain a factor-18 approximation of the minimum radius over all enclosing cylinders, in a single pass, with $O(d)$ space and update time.*

Proof: Initially, set o and v to be the first two points, and $w = 0$. Our algorithm for processing a new point p is remarkably simple:

insert(p):
1. $w = \max\{w, \mathrm{Rad}(\{o, v, p\})\}$
2. if $\|p\| > 2\|v\|$ then $v = p$

In the following analysis, w_f and v_f refer to the final values of w and v, and v_i refers to the value of v after its i-th change. Note the crucial doubling property: $\|v_i\| > 2\|v_{i-1}\|$ for all i.

Fix a point $q \in P$, where P denotes the entire input point set. Suppose that $v = v_j$ just after q is inserted. Note that $\|q\| \le 2\|v_j\|$. By Observation 3.1, $d(q, \overleftrightarrow{ov_j}) \le 2(2+1)\mathrm{Rad}(\{o, v_j, q\}) \le 6w_f$.

For $i > j$, we have $d(q, \overleftrightarrow{ov_i}) \le d(q, \overleftrightarrow{ov_{i-1}}) + d(\hat{q}, \overleftrightarrow{ov_i})$, where $\hat{q}$ is the orthogonal projection of q to $\overleftrightarrow{ov_{i-1}}$. By similarity of triangles and Observation 3.1, $d(\hat{q}, \overleftrightarrow{ov_i}) = (\|\hat{q}\|/\|v_{i-1}\|)d(v_{i-1}, \overleftrightarrow{ov_i}) \le (\|\hat{q}\|/\|v_{i-1}\|)2(1/2+1)\mathrm{Rad}(\{o, v_{i-1}, v_i\}) \le (\|q\|/\|v_{i-1}\|)3w_f$. Therefore,

$$d(q, \overleftrightarrow{ov_i}) \le d(q, \overleftrightarrow{ov_{i-1}}) + \frac{\|q\|}{\|v_{i-1}\|}3w_f.$$

Expanding the recurrence, we get a geometric series due to the doubling property:

$$d(q, \overleftrightarrow{ov_f}) \le d(q, \overleftrightarrow{ov_j}) + \frac{\|q\|}{\|v_j\|}\left(1 + \frac{1}{2} + \frac{1}{4} + \cdots\right)3w_f \le 6w_f + 2(2)3w_f = 18w_f.$$

Thus, all points lie within a cylinder of radius $18w_f$, i.e., $w_f \le \mathrm{Rad}(P) \le 18w_f$. $\square$

In the appendix, we mention an improvement of the approximation factor, by a modified algorithm that also works in non-constant dimensions.

3.2 The General Method

In this subsection, we not only further improve the approximation factor to $1 + \varepsilon$ for the cylinder problem in fixed dimensions, but also solve the more general problem of computing core-sets. Our new method is "structurally" similar but involves more steps. The "easy" case is when $\|p\| \le 2\|v\|$. The $\|p\| > 2\|v\|$ case requires changing the reference line $\overleftrightarrow{ov}$, but as in the previous proof, we can show that somehow the error does not accumulate by much.

We describe the method inductively; this allows us to reduce the easy case directly to the problem in one dimension lower by a simple rounding idea.

Lemma 3.3 *Consider the extent measure. Suppose there is a data-stream algorithm for maintaining an ε-core-set in $\mathbb{R}^{d-1}$ using $S_{d-1}(\varepsilon)$ space and $O(1)$ amortized time. Then there is a data-stream algorithm for maintaining an ε-core-set of a point set P in $\mathbb{R}^d$ using $O((1/\varepsilon)S_{d-1}(\varepsilon))$ space and $O(1)$ amortized time, under the special case when the first two input points are o and v, and all other points are within distance $2\|v\|$ from o.*

Proof: We initialize by forming $O(1/\varepsilon)$ hyperplanes $\{h_i\}_i$, perpendicular to $\overleftrightarrow{ov}$, at distances $0, \pm\varepsilon\|v\|, \pm 2\varepsilon\|v\|, \ldots, \pm 2\|v\|$ from o. We recursively maintain an ε-core-set R_i' for a $(d-1)$-dimensional stream P_i' of points inside each h_i. Given a function P_i'.insert(p) that can insert a point p to the stream P_i and return its core-set R_i' in $d-1$ dimensions, we can design a function P.insert-special(p) that inserts a point p to the stream P and returns its core-set R as follows:

P.insert-special(p):
1. let p' be the orthogonal projection of p to its nearest hyperplane h_j
2. $R'_j = P'_j$.insert(p')
3. return $R = \{o, v\} \cup \{r \mid r' \in R'_i\}$

Note that in line 3, we do not need to recompute R from scratch every time; rather, we update R, knowing the changes that occur to R'_j in line 2.

For the analysis, observe that for every $p \in P$ and vector x,

$$|(p' - p) \cdot x| \leq \varepsilon |v \cdot x|/2 \leq \varepsilon w(P, x)/2. \tag{1}$$

Consider any point $p \in P$. Say $p' \in P'_i$. Since R'_i is an ε-core-set of P'_i, by Observation 1.4 there exists a point $r' \in R'_i$ such that

$$r' \cdot x \geq p' \cdot x - \varepsilon w(P'_i, x) \implies r \cdot x + \varepsilon w(P, x)/2 \geq p \cdot x - \varepsilon w(P, x)/2 - \varepsilon[w(P, x) + \varepsilon w(P, x)],$$

because of (1) (used four times). So, $\overline{w}(R, x)$ and $\overline{w}(P, x)$ differ by at most $O(\varepsilon)w(P, x)$. By Observation 1.4 again, R is an $O(\varepsilon)$-core-set of P. □

Note that repeated applications of the above lemma give us another static algorithm to construct a core-set, of size $O(1/\varepsilon^{d-1})$, under the traditional model (since we can choose v to be the global farthest point from o to satisfy the assumption). This core-set algorithm is more self-contained than Agarwal, Har-Peled, and Varadarajan's (see Section 2.3), as Barequet and Har-Peled's subroutine [9] is not explicitly called. (This dimension-recursion approach is perhaps more reminiscent of the core-set algorithm by Har-Peled and Wang [23] in dual space.)

We are now ready to present our main result for data streams:

Theorem 3.4 *Given a stream of points in $\mathbb{R}^d$, we can maintain an ε-core-set for the extent measure, in a single pass, using $O([(1/\varepsilon)\log(1/\varepsilon)]^{d-1})$ space and $O(1)$ amortized time.*

Proof: Set $b = \lceil \log_2(1/\varepsilon) \rceil$. The proof is by induction in the dimension (the base case $d = 1$ is trivial). The idea is to maintain an ε-core-set $R^{(i)}$ of the current subset $P^{(i)}$ by invoking Lemma 3.3, until the distance constraint is violated, in which case, we build a new subset. To ensure that not too many subsets are in existence, we merge old subsets together; it turns out that because the old points are close to the origin, they can be "rounded" to form a $(d-1)$-dimensional point set P'_{old}, whose core-set R'_{old} can be maintained recursively. The precise details are given in the pseudocode below (where initially $i = 1$, the first two input points are o and v_1, and η_0 is an arbitrary hyperplane, which will contain P'_{old}):

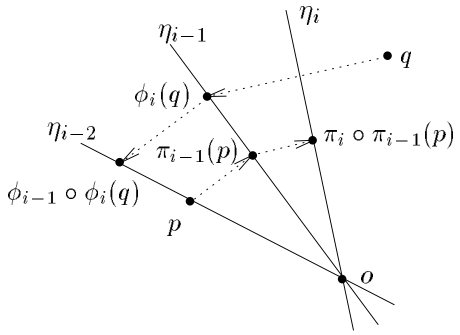

Figure 1: The projections ϕ_i and π_i.

P.insert(p):
1. if $\|p\| \leq 2\|v_i\|$ then
2. $\quad R^{(i)} = P^{(i)}$.insert-special(p)
3. else {
4. $\quad i = i+1$, $v_i = p$, and initialize $P^{(i)} = \{o, v_i\}$
5. $\quad$ let η_i be the hyperplane through o perpendicular to $\overleftrightarrow{ov_i}$ and
$\qquad \phi_i$ denote the projection to η_{i-1} parallel to the direction $\overleftrightarrow{ov_i}$
6. $\quad$ for each $q \in R^{(i-b)}$ do {
7. $\qquad q' = \phi_1 \circ \cdots \circ \phi_i(q)$
8. $\qquad R'_{\rm old} = P'_{\rm old}$.insert$(q')$
$\quad$ }
}
9. return $R = R^{(i-b+1)} \cup \cdots \cup R^{(i)} \cup \{r \mid r' \in R'_{\rm old}\}$

Notice that at any time, only b core-sets $R^{(i-b+1)}, \ldots, R^{(i)}$ are active; earlier ones can be discarded. Also, the projective maps ϕ_i need not be stored individually; rather, a single matrix representing $\phi_1 \circ \cdots \circ \phi_i$ can be maintained. The space usage therefore does not grow and is bounded by $S_d(\varepsilon) = O((1/\varepsilon) S_{d-1}(\varepsilon) \log(1/\varepsilon))$. Note that in line 9, we do not need to recompute R from scratch in line 3, but rather update R, knowing the changes that occur to $R^{(i)}$ in line 2 or $R'_{\rm old}$ in line 8. The amortized time complexity (and number of changes to R) is $O(1)$, because each point is inserted to one of the $P^{(i)}$'s once, and then to $P'_{\rm old}$ at most once.

In the following analysis, let f denote the final value of i, let π_i denote the orthogonal projection to η_i, and let $M = \pi_f \circ \cdots \circ \pi_1$. Note that ϕ_i and π_i are weak "inverses", in the sense that $\pi_i \circ \cdots \circ \pi_1 \circ \phi_1 \circ \cdots \phi_i = \pi_i$. (See Figure 1.)

For every point q and vector x, letting $\hat{q}$ denote the orthogonal projection of q to $\overleftrightarrow{ov_i}$, we have

$$|(\pi_i(q) - q) \cdot x| = \frac{\|\hat{q}\|}{\|v_i\|} |v_i \cdot x| \leq \frac{\|q\|}{\|v_i\|} w(P, x). \tag{2}$$

Now consider an arbitrary point $q' \in P'_{\rm old}$. Say $q \in R^{(j-b)}$. Then $\|q\| \leq 2\|v_{j-b}\| \leq 2^{1-b}\|v_j\|$, due to the doubling property. Since $q' = \phi_1 \circ \cdots \circ \phi_j(q)$, the weak-inverse relationship between the ϕ_i's and π_i's implies that $M(q') = \pi_f \circ \cdots \circ \pi_j(q)$. If we sum up (2) over $i = j, \ldots, f$ with q replaced by

$\pi_{i-1} \circ \cdots \circ \pi_j(q)$ (whose norm does not exceed $\|q\|$), the left-hand side telescopes and the right-hand side is upper-bounded by a geometric series due to the doubling property:

$$|(M(q') - q) \cdot x| \;\leq\; \frac{\|q\|}{\|v_j\|} \left(1 + \frac{1}{2} + \frac{1}{4} + \cdots\right) w(P, x) \;\leq\; 2^{1-b}(2) w(P, x) \;\leq\; 4\varepsilon w(P, x). \quad (3)$$

Consider any point $p \in P$. Say $p \in P^{(i)}$. Since $R^{(i)}$ is an ε-core set of $P^{(i)}$, by Observation 1.4, there exists a point $q \in R^{(i)}$ such that $q \cdot x \geq p \cdot x - \varepsilon w(P, x)$. If $i > f - b$, then set $r = q \in R$. Otherwise, $q' \in P'_{\text{old}}$; since R'_{old} is an ε-core-set of P'_{old}, $M(R'_{\text{old}})$ is an ε-core-set of $M(P'_{\text{old}})$, so by Observation 1.4, there exists a point $r \in R$, with $r' \in R'_{\text{old}}$, such that

$$M(r') \cdot x \;\geq\; M(q') \cdot x - \varepsilon w(M(P'_{\text{old}}), x) \;\Longrightarrow\; r \cdot x + 4\varepsilon w(P, x) \;\geq\; q \cdot x - 4\varepsilon w(P, x) - \varepsilon[w(P, x) + 8\varepsilon w(P, x)],$$

because of (3) (used four times). In any case, we have $r \cdot x \geq p \cdot x - O(\varepsilon) w(P, x)$. So, $\overline{w}(R, x)$ and $\overline{w}(P, x)$ differ by at most $O(\varepsilon) w(P, x)$, and by Observation 1.4, R is an $O(\varepsilon)$-core-set of P. □

An appealing feature of the above theorem is that the amortized insertion time is $O(1)$, regardless of ε. This time bound can be made worst-case by simple modifications: instead of transferring all points in $R^{(i-b)}$ to P'_{old} at the same time during a single insertion (lines 6–8), we spread out the work over all insertions. (The straightfoward details are omitted; roughly, we can maintain a superset S of $R^{(i-b+1)} \cup \cdots \cup R^{(i)}$, where the size of S is kept below or at $C(1/\varepsilon) S_{d-1}(\varepsilon) \log(1/\varepsilon)$ for a suitable constant C; each time the limit is exceeded, we can transfer the oldest point of S to P'_{old}.)

3.3 Refinement in Two Dimensions

In this subsection, we describe a slight improvement of the space bound in Theorem 3.4. We begin in dimension $d = 2$ with a very special case that can actually be solved by the core-set method from Section 2.4.

Lemma 3.5 *There is a data-stream algorithm for maintaining an ε-core-set of a point set P in $\mathbb{R}^2$ for the extent measure using $O(1/\sqrt{\varepsilon})$ space and time, under the special case when the first three input points are o, v, and u, all points are within distance $2\|v\|$ from o, and all points are within distance $2d(u, \overleftrightarrow{ov})$ from $\overleftrightarrow{ov}$.*

Proof: Let $L = \|v\|$ and $W = d(u, \overleftrightarrow{ov})$. We use the algorithm from Theorem 2.6. The main observation is that under the assumption of this lemma, Barequet and Har-Peled's bounding box [9] need not be changed as points are inserted. More precisely, fix B to be the box of length $4L$ and width $4W$ centered at o, with the longer sides parallel to $\overleftrightarrow{ov}$. Then B contains the entire point set P, and at the same time, the triangle $\triangle ovu$ (and thus the convex hull of P) contains a translated copy of $(1/16)B$ (see Figure 2).

Since the bounding box is fixed, so is the transformation used to make P fat. In the algorithm from Theorem 2.6, the nearest neighbor to each grid point in Ξ can be trivially maintained in $O(1)$ time as points are inserted (without discrete Voronoi diagrams). With $O(1/\sqrt{\varepsilon})$ grid points in Ξ, the claimed space and time bounds follow. □

We now give an improvement of Lemma 3.3 in the two-dimensional case, reducing the space bound from $O(1/\varepsilon)$ to $O((1/\sqrt{\varepsilon}) \log(1/\varepsilon))$.

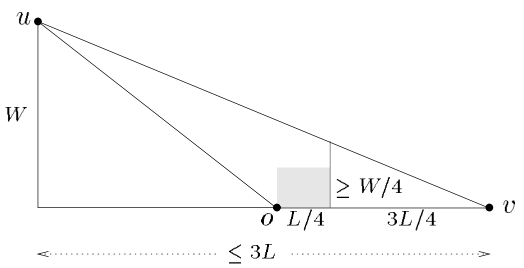

Figure 2: $\triangle ovu$ contains a box of length $L/4$ and width $W/4$. (The easier case when $\angle uov < \pi/2$ is omitted.)

Lemma 3.6 *There is a data-stream algorithm for maintaining an ε-core-set of a point set P in $\mathbb{R}^2$ for the extent measure using $O((1/\sqrt{\varepsilon})\log(1/\varepsilon))$ space and $O(1/\sqrt{\varepsilon})$ amortized time, under the special case when the first two input points are o and v, and all points are within distance $2\|v\|$ from o.*

Proof: Set $b = \lceil \log_2(1/\varepsilon) \rceil$. We use a method similar to (but a little simpler than) the method in Theorem 3.4. This time, we maintain an ε-core-set $R^{(i)}$ of the current subset $P^{(i)}$ by using Lemma 3.5, with a procedure named "insert-very-special()", until the distance-to-line constraint is violated, in which case we build a new subset. Points in old subsets are "rounded" to the line $\overleftrightarrow{ov}$ to form a one-dimensional point set P'_{old}. A 0-core-set R'_{old} of P'_{old} (consisting of just the minimum and maximum element) can be trivially maintained. The pseudocode is as follows (where initially $i = 1$, the first three input points are o, v, and u_1):

P.insert-special(p):
if $d(p, \overleftrightarrow{ov}) \leq 2d(u_i, \overleftrightarrow{ov})$ then
    $R^{(i)} = P^{(i)}$.insert-very-special(p)
else {
    $i = i + 1$, $u_i = p$, and initialize $P^{(i)} = \{o, v, u_i\}$
    for each $q \in R^{(i-b)}$ do {
        let q' denote the projection of q to $\overleftrightarrow{ov}$
        $R'_{\text{old}} = P'_{\text{old}}$.insert$(q')$
    }
}
return $R = R^{(i-b+1)} \cup \cdots \cup R^{(i)} \cup \{r \mid r' \in R'_{\text{old}}\}$

At any time, only the last b core-sets $R^{(i-b+1)}, \ldots, R^{(i)}$ need to be stored, so the space complexity is $O((1/\sqrt{\varepsilon})\log(1/\varepsilon))$.

Consider an arbitrary point $q' \in P'_{\text{old}}$ and a unit vector x. Say $q \in R^{(j-b)}$. Then $d(q, \overleftrightarrow{ov}) \leq 2d(u_{j-b}, \overleftrightarrow{ov}) \leq 2^{1-b} d(u_j, \overleftrightarrow{ov})$. By Observation 3.1, $d(u_j, \overleftrightarrow{ov}) \leq 2(2+1)\mathrm{Rad}(\{o, v, u_j\}) \leq 3\mathrm{Width}(P)$, where $\mathrm{Width}(P)$ denote the width of the two-dimensional point set P. So,

$$|(q - q') \cdot x| \;\leq\; \|q - q'\| \;=\; d(q, \overleftrightarrow{ov}) \;\leq\; 2^{1-b}(3)w(P, x) \;\leq\; 6\varepsilon w(P, x). \tag{4}$$

Consider any point $p \in P$. Say $p \in P^{(i)}$. Since $R^{(i)}$ is an ε-core set of $P^{(i)}$, by Observation 1.4, there exists a point $q \in R^{(i)}$ such that $q \cdot x \geq p \cdot x - \varepsilon w(P, x)$. If $i > f - b$ (where f denotes the last index), then set $r = q \in R$. Otherwise, $q' \in P'_{\rm old}$, so $r' \cdot x \geq q' \cdot x$ for one of the two points $r \in R$ with $r' \in R'_{\rm old}$. Thus, by (4), $r \cdot x + 6\varepsilon w(P, x) \;\geq\; q \cdot x - 6\varepsilon w(P, x)$. In any case, we have $r \cdot x \geq p \cdot x - O(\varepsilon) w(P, x)$. So, $\overline{w}(R, x)$ and $\overline{w}(P, x)$ differ by at most $O(\varepsilon) w(P, x)$, and by Observation 1.4, R is an $O(\varepsilon)$-core-set of P. □

By replacing Lemma 3.3 with the new lemma in the proof of Theorem 3.4, we can now reduce the space bound for the general two-dimensional problem from $O((1/\varepsilon)\log(1/\varepsilon))$ to $O((1/\sqrt{\varepsilon})\log^2(1/\varepsilon))$. Using this better base case in Theorem 3.4, we get the following result:

Theorem 3.7 *Given a stream of points in $\mathbb{R}^d$, we can maintain an ε-core-set for the extent measure, in a single pass, using $O((1/\varepsilon^{d-3/2})\log^d(1/\varepsilon))$ space and $O(1/\sqrt{\varepsilon})$ amortized time.*

3.4 Applications and Remarks

The data-stream results mentioned in the introduction, on the width, minimum-radius cylinder, minimum-width annulus, etc. follow immediately by applying Theorem 3.4 or Theorem 3.7 and running our best approximation algorithms on the core-set, as in Section 2.5.

The size of the core-set is $O([(1/\varepsilon)\log(1/\varepsilon)]^{d-1})$ in Theorem 3.4 or $O((1/\varepsilon^{d-3/2})\log^d(1/\varepsilon))$ in Theorem 3.7, but if needed, it can be reduced to $O(1/\varepsilon^{(d-1)/2})$ by computing a core-set of the core-set by Theorem 2.6, at the expense of increasing the time bound to $O([(1/\varepsilon)\log(1/\varepsilon)]^{d-1})$ or $O(1/\varepsilon^{d-3/2}\log^d(1/\varepsilon))$.

We leave open the question of whether further refinements can bring the space bound down to near $O(1/\varepsilon^{(d-1)/2})$.

A Appendix: High-Dimensional Cylinder Revisited

We reconsider the constant-factor data-stream algorithm for the minimum-radius cylinder problem from Section 3.1 and describe a modification that reduces the approximation factor from 18 to around 5. Despite the stronger results from Section 3.2 in fixed dimensions, these constant-factor algorithms are worthy of study because they work well even in high dimensions (in contrast to all other algorithms from this paper, which have exponential dependence on d regardless of ε).

We begin with some observations:

Observation A.1 *Given n vectors in Euclidean space of any dimension, if the angle between every pair of vectors exceeds $\pi/2 + \delta$, then $n = O(1/\delta)$.*

Proof: Without loss of generality, assume that the given vectors $v_1, \ldots, v_n$ are unit vectors. Then

$$0 \;\leq\; \left\| \sum_{i=1}^{n} v_i \right\|^2 \;=\; n + 2 \sum_{1 \leq i < j \leq n} v_i \cdot v_j \;\leq\; n + n(n-1)\cos(\pi/2 + \delta) \;=\; n[1 - (n-1)\sin\delta],$$

implying that $n \leq 1 + 1/\sin\delta$. □

Observation A.2 *Let $d(p, \overrightarrow{ov})$ denote the distance of p to the ray $\overrightarrow{ov}$. If $\|p\| \leq (1+\delta)\|v\|$ and $\angle pov \leq \pi/2 + \delta$, then*

$$d(p, \overrightarrow{ov}) \;\leq\; (2 + O(\delta))\mathrm{Rad}(\{o, v, p\}) \;\text{ or }\; \|p\| \leq (2\sqrt{2} + O(\delta))\mathrm{Rad}(\{o, v, p\}).$$

Proof: Consider triangle $\triangle ovp$ and its altitudes h_1, h_2, and h_3 at bases ov, op, and pv. If op or ov is the longest side, then

$$2\,\mathrm{Rad}(\{o,v,p\}) \;=\; \begin{cases} h_1 \;=\; d(p,\overrightarrow{ov}) & \text{if } \|v\| \geq \|p\| \\ h_2 \;=\; \frac{\|v\|}{\|p\|}h_1 \;\geq\; \frac{1}{1+\delta}d(p,\overrightarrow{ov}) & \text{if } \|p\| \geq \|v\|. \end{cases}$$

If pv is the longest side, then because $\angle pov \leq \pi/2+\delta$, we have only two subcases:

$$2\,\mathrm{Rad}(\{o,v,p\}) \;=\; h_3 \;=\; \begin{cases} \|p\|\sin\angle p \;\geq\; \frac{\|p\|}{\sqrt{2}+O(\delta)} & \text{if } \angle p \geq \pi/4-\delta/2 \\ \|v\|\sin\angle v \;\geq\; \frac{\|p\|}{(1+\delta)(\sqrt{2}+O(\delta))} & \text{if } \angle v \geq \pi/4-\delta/2, \end{cases}$$

since $\sin(\pi/4-\delta/2) \geq 1/\sqrt{2}-O(\delta)$. □

We now improve the algorithm from Theorem 3.2. The main new ideas are to (i) use a different multiplier rather than doubling and (ii) maintain a larger subset V of points rather than just two points.

Theorem A.3 *Fix a constant $\delta > 0$. Given a stream of points in $\mathbb{R}^d$ (where d is not necessarily a constant), we can maintain a factor-$(5+\delta)$ approximation of the minimum radius over all enclosing cylinders, in a single pass, with $O(d)$ space.*

Proof: The modified algorithm is as follows (initially, $V=\emptyset$, $w=0$, and the first input point is o):

insert(p):
1. $w \;=\; \max\{w, \mathrm{Rad}(V\cup\{o,p\})\}$
2. if for all $v\in V$, $\|p\| > (1+\delta)\|v\|$ or $\angle pov > \pi/2+\delta$ then {
3. insert p to V
4. remove all points v from V such that $\|v\| \leq \delta\|p\|$ and $\angle pov \leq \pi/2+\delta$

}

To analyze the space complexity, define the *index* of a point v to be the number $\lfloor \log_{1+\delta}\|v\|\rfloor$. Note that if two points $u,v\in V$ have the same index, then $\angle uov > \pi/2+\delta$ (if not, one of the points would not be inserted to V). Also note that if two points $u,v\in V$ have indices differing by more than $b := \lceil \log_{1+\delta}(1/\delta)\rceil$, then $\angle uov > \pi/2+\delta$ (if not, one of the points would not be inserted or would be removed from V). By Observation A.1, there are at most $O(1/\delta)$ points in V with indices from each equivalent class modulo $b+1$. Thus, the total number of points in V at any time is bounded by $O(b/\delta)$, a constant independent of the dimension.

To analyze the approximation factor, let w_f and V_f denote the final values of w and V. Fix a point $q\in P$, where P denotes the entire point set. Construct a sequence of points $v_0, v_1, \ldots$ as follows.

During the insertion of q, if q is inserted to V, set $v_0 = q$. Otherwise, let $v_0\in V$ be such that $\|q\| \leq (1+\delta)\|v_0\|$ and $\angle qov_0 \leq \pi/2+\delta$. By Observation A.2, we have $d(q,\overrightarrow{ov_0}) \leq (2+O(\delta))w_f$ or $\|q\| \leq (2\sqrt{2}+O(\delta))w_f$.

For $i>0$, define v_i to be the point that is inserted to V during the iteration when v_{i-1} is removed. Note that $\|v_{i-1}\| \leq \delta\|v_i\|$ and $\angle v_{i-1}ov_i \leq \pi/2+\delta$. As in the previous analysis, we have $d(q,\overrightarrow{ov_i}) \leq d(q,\overrightarrow{ov_{i-1}}) + d(\hat{q},\overrightarrow{ov_i})$, where $\hat{q}$ is the closest point on $\overrightarrow{ov_{i-1}}$ from q. By similarity of triangles and Observation A.2, $d(\hat{q},\overrightarrow{ov_i}) = (\|\hat{q}\|/\|v_{i-1}\|)d(v_{i-1},\overrightarrow{ov_i}) \leq (\|q\|/\|v_{i-1}\|)(2+O(\delta))w_f$ or $\|q\| \leq (1+\delta)\|v_{i-1}\| \leq (2\sqrt{2}+O(\delta))w_f$. Therefore,

$$d(q,\overrightarrow{ov_i}) \leq d(q,\overrightarrow{ov_{i-1}}) + \frac{\|q\|}{\|v_{i-1}\|}(2+O(\delta))w_f,$$

or $\|q\| \leq (2\sqrt{2}+O(\delta))w_f$. Expanding the recurrence, we get a geometric series: either $\|q\| \leq (2\sqrt{2}+O(\delta))w_f$, or for all i,

$$\begin{aligned} d(q,\overrightarrow{ov_i}) &\leq d(q,\overrightarrow{ov_0}) + \frac{\|q\|}{\|v_0\|}(1+\delta+\delta^2+\cdots)(2+O(\delta))w_f \\ &\leq (2+O(\delta))w_f + (1+\delta)(1+O(\delta))(2+O(\delta))w_f \;=\; (4+O(\delta))w_f. \end{aligned}$$

Since $\|q\| \leq (1+\delta)\|v_i\|$, in either case, q is of distance at most $(4+O(\delta))w_f$ from the scaled line segment $(1+\delta)\overline{ov_i}$.

We conclude that all points of P lie within a distance of $(4+O(\delta))w_f$ from a $(1+\delta)$-factor scaled copy of the convex hull of $V_f \cup \{o\}$. So, $w_f \leq \mathrm{Rad}(P) \leq (1+\delta)\mathrm{Rad}(V_f \cup \{o\}) + (4+O(\delta))w_f \leq (5+O(\delta))w_f$. □

Note that in the above algorithm, instead of computing $\mathrm{Rad}(V \cup \{o,p\})$ exactly, we can apply Har-Peled and Varadarajan's $(1+\delta)$-factor cylinder algorithm [22] in high dimensions.

It would be interesting to see what is the smallest constant factor achievable in the data-stream setting for the cylinder problem in high dimensions. A similar question can asked for related problems, like the smallest enclosing j-flat. (For example, Indyk [26] has given a data-stream algorithm for the diameter problem in high dimensions, with sublinear space and factor arbitrarily close to $\sqrt{2}$.)

References

[1] P. K. Agarwal, B. Aronov, and M. Sharir. Line transversals of balls and smallest enclosing cylinders in three dimensions. *Discrete Comput. Geom.*, 21:373–388, 1999.

[2] P. K. Agarwal, L. J. Guibas, S. Har-Peled, A. Rabinovitch, and M. Sharir. Computing the penetration depth of two convex polytopes in 3D. *Nordic J. Comput.*, 7:227–240, 2000.

[3] P. K. Agarwal, S. Krishnan, N. H. Mustafa, and S. Venkatasubramanian. Streaming geometric optimization using graphics hardware. In *Proc. 11th European Sympos. Algorithms*, Lect. Notes Comput. Sci., vol. 2832, Springer-Verlag, pages 544–555, 2003.

[4] P. K. Agarwal, S. Har-Peled, and K. R. Varadarajan. Approximating extent measures of points. *J. ACM*, to appear. Preliminary versions appeared in *Proc. 12th ACM-SIAM Sympos. Discrete Algorithms*, pages 148–157, 2001; *Proc. 42nd IEEE Sympos. Found. Comput. Sci.*, pages 66–73, 2001.

[5] P. K. Agarwal, J. Matoušek, and S. Suri. Farthest neighbors, maximum spanning trees and related problems in higher dimensions. *Comput. Geom. Theory Appl.*, 1:189–201, 1992.

[6] S. Arya, T. Malamatos, and D. M. Mount. Space-efficient approximate Voronoi diagrams. In *Proc. 34th ACM Sympos. Theory Comput.*, pages 721–730, 2002.

[7] M. Bădoiu and K. L. Clarkson. Optimal core-sets for balls. Manuscript, 2003. Preliminary version in *Proc. 14th ACM-SIAM Sympos. Discrete Algorithms*, pages 801–802, 2003.

[8] M. Bădoiu, S. Har-Peled, and P. Indyk. Approximate clustering via core-sets. In *Proc. 34th ACM Sympos. Theory Comput.*, pages 250–257, 2002.

[9] 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.

[10] J. Bentley and J. Saxe. Decomposable searching problems I: static-to-dynamic transformation. *J. Algorithms*, 1:301–358, 1980.

[11] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Heidelberg, 1997.

[12] H. Breu, J. Gil, D. Kirkpatrick, and M. Werman. Linear time Euclidean distance transform algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:529-533, 1995.

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

[14] T. M. Chan. Approximating the diameter, width, smallest enclosing cylinder, and minimum-width annulus. *Int. J. Comput. Geom. Appl.*, 12:67–85, 2002.

[15] T. M. Chan and B. S. Sadjad. Geometric optimization problems over sliding windows. In *Proc. 15th Int. Sympos. Algorithms and Computation*, Lect. Notes Comput. Sci., vol. 3341, Springer-Verlag, pages 246–258, 2004.

[16] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. McGraw-Hill, 2nd ed., 2001.

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

[18] C. A. Duncan, M. T. Goodrich, and E. A. Ramos. Efficient approximation and optimization algorithms for computational metrology. In *Proc. 8th ACM-SIAM Sympos. Discrete Algorithms*, pages 121–130, 1997.

[19] J. Feigenbaum, S. Kannan, and J. Zhang. Computing diameter in the streaming and sliding-window models. *Algorithmica*, 41:25–41, 2004.

[20] S. Har-Peled. A practical approach for computing the diameter of a point-set. In *Proc. 17th ACM Sympos. Comput. Geom.*, pages 177–186, 2001.

[21] S. Har-Peled. A replacement for Voronoi diagrams of near linear size. In *Proc. 42nd IEEE Sympos. Found. Comput. Sci.*, pages 94–103, 2001.

[22] S. Har-Peled and K. Varadarajan. High-dimensional shape fitting in linear time. In *Proc. 19th ACM Sympos. Comput. Geom.*, pages 39–47, 2003.

[23] S. Har-Peled and Y. Wang. Shape fitting with outliers. *SIAM J. Comput.*, 33:269–285, 2004.

[24] M. Henzinger, P. Raghavan, and S. Rajagopalan. Computing on data streams. Technical Note 1998–011, Digital Systems Research Center, Palo Alto, CA, 1998.

[25] J. Hershberger and S. Suri. Adaptive sampling for geometric problems over data streams. In *Proc. 23rd Sympos. Principles on Database Systems*, pages 252–262, 2004.

[26] P. Indyk. Better algorithms for high-dimensional proximity problems via asymmetric embeddings. In *Proc. 14th ACM-SIAM Sympos. Discrete Algorithms*, pages 539–545, 2003.

[27] R. Janardan. On maintaining the width and diameter of a planar point-set online. *Int. J. Comput. Geom. Appl.*, 3:331–344, 1993.

[28] G. Malandain and J.-D. Boissonnat. Computing the diameter of a point set. *Int. J. Comput. Geom. Appl.*, 12:489–510, 2002.

[29] S. Muthukrishnan. Data streams: algorithms and applications. Manuscript, 2003.

[30] G. Rote, C. Schwarz, and J. Snoeyink. Maintaining the approximate width of a set of points in the plane. In *Proc. 5th Canad. Conf. Comput. Geom.*, pages 258–263, 1993.

[31] O. Schwarzkopf. Parallel computation of distance transforms. *Algorithmica*, 6:685–697, 1991.

[32] H. Yu, P. K. Agarwal, R. Poreddy, and K. R. Varadarajan. Practical methods for shape fitting and kinetic data structures using core sets. In *Proc. 20th ACM Sympos. Comput. Geom.*, pages 263–272, 2004.