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

Timothy M. Chan†

December 11, 2003

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 annulus, minimum-volume bounding box, minimum-width 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.

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 throughout, unless stated otherwise. 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 [13] 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 simple algorithms running in $O(n+1/\varepsilon^{d-1/2})$ time; the time bound can be reduced slightly to $O((n+1/\varepsilon^{d-5/3}) \operatorname{polylog}(1/\varepsilon))$ in theory, using advanced data structures.

*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`

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.* [15] 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 [13]. (Slight improvements were known in low dimensions using more complicated data structures; for example, for $d = 3$, the author [13] 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 $O((n + 1/\varepsilon^2)\,\mathrm{polylog}\,(1/\varepsilon))$ 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 [13] 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. A *bounding box* of the minimum volume. Barequet and Har-Peled [9] described linear-time approximation algorithms, the best of which require $O(n \log n + 1/\varepsilon^3)$ time for $d = 3$.

   We give a slightly better time bound for $d = 3$: $O(n + 1/\varepsilon^3)$.

5. An enclosing *spherical shell/annulus* (region between two concentric spheres) of the minimum width. The author [13] 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 [13] for all $d \geq 4$.

6. 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, reducing time bounds of the form $O(n + 1/\varepsilon^c)$ to $O(n + 1/\varepsilon^{2c/3})$.

7. 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.

Agarwal, Har-Peled, and Varadarajan [4] have also given data structures that support insertion only but with remarkably little space. The need to handle massive data has generated considerable attention recently to the *data stream* model [21, 22, 25], 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 $O((1/\varepsilon^{(d-1)/2})\log^d n + 1/\varepsilon^{3(d-1)/2})$ 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 earlier algorithms [5] uses just $O(1/\varepsilon^{(d-1)/2})$ space in the data stream setting (see also [16, 22]). For the width, no previous algorithms yield constant-space solutions; the problem is apparently open even in the two-dimensional case, as Hershberger and Suri [22] mentioned only partial results under the assumption that the diameter-to-width ratio is bounded. (See also [3] for multiple-pass algorithms that exploit graphics hardware.)

In the second part of the paper, we answer the question in the affirmative by obtaining one-pass, data stream algorithms for all the above problems using an amount of storage independent of n. For the width and the cylinder problem, for example, the space and amortized insertion time are bounded by $O([(1/\varepsilon)\log(1/\varepsilon)]^{d-1})$. (This result even improves known data structures [23, 26] for approximating the two-dimensional width in the traditional model; these structures required linear space and logarithmic insertion time.) For the spherical-shell problem, the bound is $O([(1/\varepsilon)\log(1/\varepsilon)]^{2d})$.

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 will focus only on the case of fixed dimensions.

Definition 1.1 Given a single-argument measure $\mu(\cdot)$ that is monotone (i.e., $Q \subseteq P$ implies $\mu(Q) \leq \mu(P)$), we say that a subset $R \subseteq P$ is an *ε-core-set of P* if $\mu(R) \geq (1-\varepsilon)\mu(P)$.

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 to 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 double-argument measure $\mu(\cdot,\cdot)$ that is monotone in its first argument, we say that a subset $R \subseteq P$ is an *ε-core-set of P (over a set S)* if $\mu(R,x) \geq (1-\varepsilon)\mu(P,x)$ 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 $w(P,x) = \max_{p,q\in P}(p-q)\cdot x$ over all vector $x \in \mathbb{R}^d$.

Surprisingly, considering just the extent measure 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, minimum-cylinder radius, and minimum-volume box. 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 in 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 symmetric domain S, then R is a (2ε)-core-set of P for the extent measure over S.*

Proof: Straightforward (e.g., for the second part, just note that $w(P,x) = \overline{w}(P,x) + \overline{w}(P,-x)$). $\square$

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 will 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 of a grid point set to all grid points. This notion is different from "approximate Voronoi diagrams" [6, 18] and does not seem to have receive as much attention in computational geometry, although the problem has been considered by Breu *et al.* [11] (see also [28]) 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 "corrects" itself as points are inserted via a doubling strategy. The analysis interestingly bounds the error in terms of a geometric series. The main idea can be illustrated by a very simple data stream algorithm for the minimum-radius enclosing cylinder that achieves a constant-factor approximation, 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, or more precisely, a sampling of the upper envelope at a set of grid points (or dually, computing extreme points along various "grid directions"). The trivial algorithm to solve this problem requires time $O(nF^{d-1})$, which could be as big as $O(E^{d-1}F^{d-1})$.

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 its 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*

$$q[x] = \text{a point } p \in P \text{ maximizing } p_1x_1 + \cdots + p_{d-1}x_{d-1} + p_d$$

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

Proof: The algorithm proceeds as follows:

1. for $i \in [E]$ do
2. for $x_2, \ldots, x_{d-1} \in [F]$ do
3. $r[i, x_2, \ldots, x_{d-1}] =$ a point $p \in P$ with $p_1 = i$ that maximizes $p_2x_2 + \cdots + p_{d-1}x_{d-1} + p_d$

```
4. for $x_2, \ldots, x_{d-1} \in [F]$ do
5.     for $x_1 \in [F]$ do
6.         $q[x_1, \ldots, x_{d-1}]$ = a point $p \in \{r[i, x_2, \ldots, x_{d-1}] \mid i \in [E]\}$ that
                maximizes $p_1x_1 + p_2x_2 + \cdots + p_{d-1}x_{d-1} + p_d$
```

Correctness is self-evident: for points $p \in P$ with the same first coordinate, maximizing $p_1x_1 + \cdots + p_{d-1}x_{d-1} + p_d$ is equivalent to maximizing $p_2x_2 + \cdots + p_{d-1}x_{d-1} + p_d$.

For the running time, 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 $x_2, \ldots, x_{d-1} \in [F]$, the loop at lines 5–6 is a 2-dimensional subproblem of the same type (think of $p_2x_2 + \cdots + p_{d-1}x_{d-1} + p_d$ as the second coordinate), on a subset of size E. We thus have the following recurrence, where $\sum_{i=1}^{E} n_i = n$:

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

The base case $d = 2$ can be handled by explicitly constructing the upper envelope, which in the dual corresponds to a planar convex hull. Since the first coordinates are from $[E]$ and can thus be radix-sorted in $O(n + E^{\delta})$ time, Graham's scan [10] requires $O(n + E^{\delta})$ time. The answers $q[x]$ for all $x \in [F]$ can be computed by another linear scan. Therefore, $T_2(n) = O(n + F)$. 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 [10]. The answer $q[x]$ for each $x \in [F]^2$ can be obtained by a point location query [10]. This gives $T_3(n) = O((n + F^2) \log n) = O((n + F^2) \log E)$ (since a naive preprocessing can ensure that $n \leq E^2$). The recurrence now solves to $T_d(n) = O((n + E^{d-3}F^2) \log E)$. □

By the standard lifting map, Voronoi diagrams of point sets become 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 $x \in [F]^k \times \{0\}^{d-k}$, a point $p \in P$ minimizing $\|p - x\|$ maximizes $2p_1x_1 + \cdots + 2p_kx_k - \|p\|^2$, so the first part is an instance of the theorem in dimension $k + 1$. The second part is a 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.* [11] 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, 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 author's approximation algorithms for the diameter problem [13] (specifically, the fourth and the fifth).

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 borrowed from the third diameter algorithm in [13]: the idea is to round both the input points and the space of all directions.

In the following, 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 whose boundary ∂B has distance at least 1 from the origin o. Given an ε-grid over ∂B, for any vector x, there exists a grid point a such that the angle $\angle(a, x)$ between $\overleftrightarrow{oa}$ 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 a with $\|a - x\| \le \varepsilon/2$, implying that $2a \cdot x \ \ge\ \|a\|^2 + \|x\|^2 - \varepsilon^2/4 \ \ge\ 2\|a\|\,\|x\| - \varepsilon^2/4 \ \ge\ 2\|a\|\,\|x\|(1 - \varepsilon^2/8)$. □

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 $o \in P$ and let v be its farthest neighbor; 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 A be the points of a $\sqrt{\varepsilon}$-grid over $\partial[-1, 1]^d$
3. return the farthest pair in $\{(p_a, q_a)\}_{a \in A}$, where $p_a, q_a \in P$ maximize $(p'_a - q'_a) \cdot a$

The running time is dominated by that of line 3. Observe that this step (maximizing $p' \cdot a$ and minimizing $q' \cdot a$ for each point a in $2d$ $(d-1)$-dimensional grids) reduces precisely to the 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, consider any pair $p, q \in P$. By Observation 2.3, there is a grid point $a \in A$ with $\angle(a, p' - q') \le \arccos(1 - \varepsilon/8)$. Then

$$
\begin{aligned}
(p'_a - q'_a) \cdot a \ \ge\ (p' - q') \cdot a \quad &\Longrightarrow \quad \|p'_a - q'_a\|\,\|a\| \ \ge\ \|p' - q'\|\,\|a\|(1 - \varepsilon/8) \\
&\Longrightarrow \quad \|p_a - q_a\| + \varepsilon\Delta \ \ge\ (\|p - q\| - \varepsilon\Delta)(1 - \varepsilon/8),
\end{aligned}
$$

so the approximation factor is $1 + O(\varepsilon)$. □

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 consider 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 borrowed from 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: Deferred to the appendix. □

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 idea is based on those of Dudley [14] and Bronshteyn and Ivanov [12]. 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/Corollary 2.2. In order to do 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: Deferred to the appendix. □

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 apply Duncan *et al.*'s approximation algorithm [15] to R. The author [13] 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 $O(1/\varepsilon^{d-2})$ if $d>4$, so the total time is within $O((n+1/\varepsilon^{d-2})\log(1/\varepsilon))$.

Since an ε-core-set under the extent measure is also an $O(\varepsilon)$-core-set under the minimum-cylinder radius or minimum-box volume, we can also solve these problems (items 3 and 4) by applying the second method (Theorem 2.6) to obtain a core set R and invoking known approximation algorithms on R. For the cylinder problem, we can use the author's $O((1/\varepsilon^{(d-1)/2}|R|)$-time algorithm [13] to get an $O(n+1/\varepsilon^{d-1})$ bound (which can also be marginally improved with advanced data structures). For the bounding box problem in dimension $d=3$, we can use O'Rourke's exact $O(|R|^3)$-time algorithm [27] to get an $O(n+1/\varepsilon^3)$ bound.

Other problems (items 5 and 6) can be solved by taking appropriate lifting maps (*linearization*), 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 a restriction of 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 [17] and Malandain and Boissonnat [24] 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 enclosing 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 [19]). The algorithm just picks an arbitrary input point, say o, finds the farthest point v from o, and returns the farthest point from 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 of the above algorithm.

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 $h_1|ov| = h_2|op| = h_3|pv|$ and $\mathrm{Rad}(\{o,v,p\}) = \min\{h_1,h_2,h_3\}/2$,

$$h_1 \;=\; 2\mathrm{Rad}(\{o,v,p\})\max\left\{1,\frac{h_1}{h_2},\frac{h_1}{h_3}\right\} \;\le\; 2\mathrm{Rad}(\{o,v,p\})\max\left\{1,\frac{|op|}{|ov|},\frac{|ov|+|op|}{|ov|}\right\}.$$

□

Unfortunately, the above algorithm requires two passes, one to find v and one to find the radius. Nevertheless, we show that a very simple variant of the algorithm, which maintains an approximate candidate for v on-line, can still work, albeit with a larger constant:

Theorem 3.2 *Given a stream of points in $\mathbb{R}^d$ (where d is not necessarily 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, say o and v are the first two points, and set $w=0$. Our algorithm for inserting a new point is this:

```
insert(p):
1.  w = max{w, 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 that $\|v_i\| > 2\|v_{i-1}\|$ 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. Since $\|q\| \leq 2\|v_j\|$, Observation 3.1 implies that $d(q, \overleftrightarrow{ov_j}) \leq 6w_f$.

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

$$d(q, \overleftrightarrow{ov_i}) \leq \begin{cases} 6w_f & \text{if } i = j \\ d(q, \overleftrightarrow{ov_{i-1}}) + (\|q\|/\|v_{i-1}\|)3w_f & \text{if } i > j \end{cases}$$

Expanding the recurrence, we get a geometric series:

$$d(q, \overleftrightarrow{ov_f}) \;\leq\; 6w_f + (\|q\|/\|v_j\|)(1 + 1/2 + 1/4 + \cdots)3w_f \;\leq\; 6w_f + (2)(2)3w_f \;=\; 18w_f.$$

So, $w_f \leq \mathrm{Rad}(P) \leq 18w_f$. □

3.2 The General Method

In this subsection, we not only improve the approximation factor for the cylinder problem (in fixed dimensions) but solve the more general problem of computing core-sets. Our new method is structurally based on our cylinder algorithm but involves more steps. The "easy" case is when the approximate diameter $\|v\|$ does not change by much. The "hard" case requires changing axes, but as in the previous proof, the error does not accumulated by much.

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

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 $\{h_i\}$, perpendicular to $\overleftrightarrow{ov}$, at distances $0, \varepsilon\|v\|, 2\varepsilon\|v\|, \ldots, 2\|v\|$ 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. We insert a new point as follows:

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 $R = \{o, v\} \cup \{r \mid r' \in R_i'\}$

To show that R is an $O(\varepsilon)$-core-set of the input point set P, observe that for every $p \in P$ and vector x, $|(p' - p) \cdot x| \leq \varepsilon|v \cdot x|/2 \leq \varepsilon w(P, x)/2$. Consider any $p \in P$. Say $p' \in P_i'$. Since R_i' is an ε-core-set of P_i', by Observation 1.4 there exists an $r' \in R_i'$ such that

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

so $\overline{w}(R, x)$ and $\overline{w}(P, x)$ differ by at most $O(\varepsilon)w(P, x)$. □

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 always choose v to be the farthest point from o to satisfy the assumption). This 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 closer to a core-set algorithm by Har-Peled and Wang [20] when dualized.)

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 move on to a new subset. To ensure that not too many subsets are in existence, we merge earlier subsets together; because that their points are close to the origin, it turns out that these points can be handled by a $(d-1)$-dimensional stream Q'. The pseudocode is as follows (where initially $i = 1$, and h_0 is an arbitrary hyperplane, which will contain Q'):

insert(p):
if $\|p\| < 2\|v_i\|$ then $R_i = P_i$.insert-special(p) else {
  $i = i + 1$, $v_i = p$, and initialize $P_i = \{o, v_i\}$
  for each $q \in R_{i-b}$ do {
    let h_i be the hyperplane through o perpendicular to $\overleftrightarrow{ov_i}$ and
      ϕ_i denote the projection to h_{i-1} in the direction $\overleftrightarrow{ov_i}$
    $q' = \phi_1 \circ \cdots \circ \phi_i(q)$
    $R' = Q'$.insert(q')
  }
}
return $R = R_{i-b+1} \cup \cdots \cup R_i \cup \{r \mid r' \in R'\}$

For the space complexity, notice that at any time, only b streams $P_{i-b+1}, \ldots, P_i$ are active; earlier ones are discarded. Also, the projective maps ϕ_i are not stored individually; rather, a single matrix representing $\phi_1 \circ \cdots \circ \phi_i$ is stored. The space used therefore does not grow and is bounded by $O((1/\varepsilon)S_{d-1}(\varepsilon)\log(1/\varepsilon))$. For the time complexity, just note that each point is inserted to Q' at most once.

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

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

$$|(\pi_i(q) - q) \cdot x| = \frac{\|\hat{q}\|}{\|v_i\|}|v_i \cdot x| \le \frac{\|q\|}{\|v_i\|} w(P, x). \tag{1}$$

Now consider an arbitrary point $q' \in Q'$. Say $q \in R_{j-b}$, so that $\|q\| \le 2\|v_{j-b}\| \le 2^{1-b}\|v_j\|$. Observe that $M(q') = \pi_f \circ \cdots \circ \pi_j(q)$. Applying (1) repeatedly (and noting that $\|q\|$ does not increase through orthogonal projections), we get a geometric series:

$$|(M(q') - q) \cdot x| \;\le\; (\|q\|/\|v_j\|)(1 + 1/2 + 1/4 + \cdots) w(P, x) \;\le\; (2^{1-b})(2) w(P, x) \;\le\; 4\varepsilon w(P, x).$$

Consider any point $p \in P$. Say $p \in P_i$. If $i > f - b$, then since R_i is an ε-core set of P_i, there exists an $r \in R_i \subseteq R$ such that $r \cdot x \geq p \cdot x - \varepsilon w(P, x)$. Otherwise, there exists a $q \in R_i$ (with $q' \in Q'$) such that $q \cdot x \geq p \cdot x - \varepsilon w(P, x)$. Furthermore, since S' is an ε-core-set of Q', $M(R')$ is an ε-core set of $M(Q')$, so by Observation 1.4 there exists an $r' \in R'$ such that

$$M(r') \cdot x \geq M(q') \cdot x - \varepsilon w(M(Q'), x) \implies r \cdot x + 4\varepsilon w(P, x) \geq q \cdot x - 4\varepsilon w(P, x) - \varepsilon[w(P, x) + 8\varepsilon w(P, x)].$$

Thus, we have $r \cdot x \geq p \cdot x - O(\varepsilon) w(P, x)$ in this case also. So, $\overline{w}(R, x)$ and $\overline{w}(P, x)$ differ by at most $O(\varepsilon) w(P, x)$, and R is an $O(\varepsilon)$-core set of P. $\square$

3.3 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 and running our best approximation algorithms on the core-set, as in Section 2.5.

The size of the core-set in Theorem 3.4 is $O([(1/\varepsilon)\log(1/\varepsilon)]^{d-1})$, but if needed, it can be reduced to $O(1/\varepsilon^{(d-1)/2})$ by computing a core-set of the core-set with Theorem 2.6, although the space bound remains the same.

We leave open the question of whether $O(1/\varepsilon^{(d-1)/2})$ space is attainable. Currently, we are able to reduce space to $O(\sqrt{1/\varepsilon}\log(1/\varepsilon))$ in the two-dimensional case; with this newer base case, we get only a slightly improved bound of $O((1/\varepsilon^{d-3/2}\log^{d-1}(1/\varepsilon))$ in d dimensions.

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 hardward. In *Proc. 11th European Sympos. Algorithms*, Lect. Notes in Comput. Sci., vol. ??, Springer-Verlag, pages ??, 2003.

[4] P. K. Agarwal, S. Har-Peled, and K. R. Varadarajan. Approximating extent measures of points. `http://valis.cs.uiuc.edu/~sariel/research/papers/01/fitting/`, 2003. 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. `http://cm.bell-labs.com/who/clarkson/pubs.html`, 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] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Heidelberg, 1997.

[11] 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.

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

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

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

[15] 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.

[16] J. Feigenbaum, S. Kannan, and J. Zhang. Computing diameter in the streaming and sliding-window models. Technical report YALEU/DCS/TR–1245, Yale University, 2002.

[17] 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.

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

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

[20] S. Har-Peled and Y. Wang. Shape fitting with outliers. In *Proc. 19th ACM Sympos. Comput. Geom.*, pages 29–38, 2003.

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

[22] J. Hershberger and S. Suri. Convex hulls and related problems in data streams. *ACM SIGMOD/PODS Workshop on Management and Processing of Data Streams (MPDS)*, `http://www.research.att.com/conf/mpds2003/schedule/`, 2003.

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

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

[25] S. Muthukrishnan. Data streams: algorithms and applications. `http://www.cs.rutgers.edu/~muthu/`, 2003.

[26] 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.

[27] J. O'Rourke. Finding minimal enclosing boxes. *Internat. J. Comput. Info. Sci.*, 14:183–199, 1986.

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

A Proof of Theorem 2.5

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 an affine transformation, we may assume that $B = [-1, 1]^d$; this step is justified, because 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 $M^T(\Delta)$.

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
2. let A be all cell centers of an ε-grid over $\partial[-1, 1]^d$
3. return regions $\{S_a\}_{a \in A}$ and subsets of size 2 $\{\{p_a, q_a\}\}_{a \in A}$
   where S_a is the cone with apex at o through the cell centered at a and
   $p_a, q_a \in P$ maximize $(p'_a - q'_a) \cdot a$

Line 3 is handled by Theorem 2.1 as before, this time with both $E, F = O(1/\varepsilon)$.

To analyze the approximation factor, fix a vector $x \in \partial[-1, 1]^d$ lying in S_a. Since $\|a - x\| \leq \varepsilon/2$ and $P \subseteq [-1, 1]^d$, for every $p, q \in P$,

$$\begin{aligned} (p'_a - q'_a) \cdot a \geq (p' - q') \cdot a \quad &\Longrightarrow \quad (p'_a - q'_a) \cdot x + \sqrt{d}\varepsilon \geq (p' - q') \cdot x - \sqrt{d}\varepsilon \\ &\Longrightarrow \quad (p_a - q_a) \cdot x + 2\sqrt{d}\varepsilon \geq (p - q) \cdot x - 2\sqrt{d}\varepsilon. \end{aligned}$$

Thus, $w(\{p_a, q_a\}, x)$ 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, $\{p_a, q_a\}$ is an $O(\varepsilon)$-core-set over $S_a \cap \partial[-1, 1]^d$, and by scaling, over all of S_a. □

B Proof of Theorem 2.6

Following the proof of Theorem 2.5, we invoke Barequet and Har-Peled's bounding box method and apply an affine 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
2. let A be the points of a $\sqrt{\varepsilon}$-grid over $\partial[-2, 2]^d$
3. return $R = \{p_a\}_{a \in A}$, where $p_a \in P$ minimizes $\|a - p'_a\|$

Line 3 is handled by ($2d$ instances of) Corollary 2.2, with $k = d - 1$, $E = O(1/\varepsilon)$, and $F = O(1/\sqrt{\varepsilon})$.

For the analysis, consider 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 an $a \in A$ such that $\angle(a - p', x) \leq \arccos(1 - \varepsilon/8)$. Then

$$\begin{aligned} \|a - p'_a\| \leq \|a - p'\| \quad &\Longrightarrow \quad (a - p'_a) \cdot x \leq (a - p') \cdot x/(1 - \varepsilon/8) \\ &\Longrightarrow \quad (a - p'_a) \cdot x - 3\varepsilon/8 \leq (a - p') \cdot x \\ &\Longrightarrow \quad p_a \cdot x \geq p \cdot x - 11\varepsilon/8. \end{aligned}$$

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

Remark: Our simplification comes from line 3: in Agarwal *et al.*'s original algorithm [4], we find for each $a \in A$ the nearest neighbor to the convex hull boundary of the rounded points. (Such a computation would require generalized linear programming.)