Approximation and Exact Algorithms for Minimum-Width Annuli and Shells*

Pankaj K. Agarwal† Boris Aronov‡ Sariel Har-Peled§ Micha Sharir¶

Abstract

Let S be a set of n points in $\mathbb{R}^d$. The "roundness" of S can be measured by computing the width $\omega^*(S)$ of the thinnest spherical shell (or annulus in $\mathbb{R}^2$) that contains S. This paper contains four main results related to computing $\omega^*(S)$: (i) For $d = 2$, we can compute in $O(n \log n)$ time an annulus containing S whose width is at most $2\omega^*(S)$. (ii) For $d = 2$ we can compute, for any given parameter $\varepsilon > 0$, an annulus containing S whose width is at most $(1+\varepsilon)\omega^*(S)$, in time $O(n \log n + n/\varepsilon^2)$. (iii) For $d \geq 3$, given a parameter $\varepsilon > 0$, we can compute a shell containing S of width at most $(1+\varepsilon)\omega^*(S)$ in time $O\left(\frac{n}{\varepsilon^d} \log\left(\frac{\mathrm{diam}(S)}{\omega^*(S)\varepsilon}\right)\right)$ or $O\left(\left(\frac{n \log n}{\varepsilon^{d-2}} + \frac{n}{\varepsilon^{d-1}}\right) \log\left(\frac{\mathrm{diam}(S)}{\omega^*(S)\varepsilon}\right)\right)$. (iv) For $d = 3$, we present an $O(n^{3-1/19+\varepsilon})$-time algorithm to compute a minimum-width shell containing S.

*Work by P.A. was supported by Army Research Office MURI grant DAAH04-96-1-0013, by a Sloan fellowship, by NSF grants EIA–9870724, and CCR–9732787, by an NYI award, and by a grant from the U.S.-Israeli Binational Science Foundation. Work by B.A. was supported by a Sloan Research Fellowship and by a grant from the U.S.-Israeli Binational Science Foundation. Work by M.S. was supported by NSF Grants CCR-97-32101, CCR-94-24398, by grants from the U.S.-Israeli Binational Science Foundation, the G.I.F., the German-Israeli Foundation for Scientific Research and Development, and the ESPRIT IV LTR project No. 21957 (CGAL), and by the Hermann Minkowski–MINERVA Center for Geometry at Tel Aviv University. Part of the work by P.A. and B.A. on the paper was done when they visited Tel Aviv University in May 1998.

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

‡Department of Computer and Information Science, Polytechnic University, Brooklyn, NY 11201-3840, USA. E-mail: aronov@ziggy.poly.edu

§School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel. E-mail: sariel@math.tau.ac.il

¶School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel; and Courant Institute of Mathematical Sciences, New York University, New York, NY 10012, USA. E-mail: sharir@math.tau.ac.il

1 Introduction

Let S be a set of n points in $\mathbb{R}^d$. The *roundness* of S can be measured by approximating S with a sphere Γ so that the maximum distance between a point of S and Γ is minimized, i.e., by computing

$$\min_{c \in \mathbb{R}^d, r \in \mathbb{R}} \max_{p \in S} |d(p,c) - r|.$$

For $c \in \mathbb{R}^d$ and for $r, R \in \mathbb{R}$ with $0 \leq r \leq R$, we define the *spherical shell* (*shell*, for short, and, in the plane, *annulus*) $\mathcal{A}(c, r, R)$ to be the closed region lying between the two concentric spheres of radii r and R centered at c. The *width* of $\mathcal{A}(c, r, R)$ is $R - r$. The problem of measuring the roundness of S is equivalent to computing a shell, $\mathcal{A}^*(S)$, of the smallest width that contains S. See Figure 1.

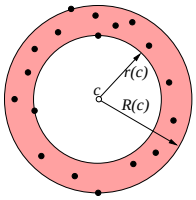

Figure 1: The annulus $\mathcal{A}^*(S)$.

The main motivation for computing a minimum-width shell or annulus comes from metrology. For example, the circularity of a two-dimensional object O in the plane is measured by sampling a set S of points on the surface of O (e.g. using coordinate measurement machines) and computing the width of the thinnest shell containing S [18]. Motivated by this and other applications, the problem of computing $\mathcal{A}^*(S)$ in the plane has been studied extensively [1, 5, 6, 7, 15, 16, 23, 25, 27, 28, 29, 30, 32, 33, 35]. Ebara *et al.* [15] noticed that in the planar case the center of $\mathcal{A}^*(S)$ is a vertex of the overlay of the nearest- and farthest-neighbor Voronoi diagrams of S. This property was later refined and extended in [19, 33]. These

observations immediately lead to an $O(n^2)$-time algorithm for computing $\mathcal{A}^*(S)$ in the plane. Subquadratic algorithms were later developed in [1, 5, 6]. The asymptotically fastest known randomized algorithm, by Agarwal and Sharir [5], computes $\mathcal{A}^*(S)$ in expected time $O(n^{3/2+\varepsilon})$, for any $\varepsilon > 0$. Since the subquadratic algorithms are rather complicated, simpler and faster algorithms have been developed for various special cases [11, 13, 23, 34]. Mehlhorn *et al.* [25] and Kumar and Sivakumar [22] have studied this problem under the *probing model* in which the set S of sample points is chosen adaptively; see the original papers for details.

Very little is known about computing $\mathcal{A}^*(S)$ efficiently in higher dimensions. Extending the observation by Ebara *et al.* [15] to $\mathbb{R}^3$, it can be shown that the center of $\mathcal{A}^*(S)$ is the intersection point of an edge of the nearest-neighbor Voronoi diagram of S with a face of the farthest-neighbor Voronoi diagram of S, or vice versa. Using this fact, $\mathcal{A}^*(S)$ can be computed in $O(n^3 \log n)$ time [13]. This idea can also be extended to higher dimensions.

This paper contains four main results.

(i) For $d = 2$, we describe in Section 3 a very simple $O(n \log n)$-time algorithm for computing an annulus that contains S and whose width is at most twice that of $\mathcal{A}^*(S)$. Duncan *et al.* [13] had described an approximation algorithm, but the approximation factor of their algorithm depends on the inner radius of $\mathcal{A}^*(S)$. No near-linear time algorithm with constant-factor approximation was previously known.

(ii) For $d = 2$, given a parameter $\varepsilon > 0$, we can compute an annulus that contains S whose width is at most $(1+\varepsilon)\omega^*$, where ω^* is the width of $\mathcal{A}^*(S)$ (section 4.1). The algorithm runs in $O(n \log n + n/\varepsilon^2)$ time.

(iii) For $d \geq 3$, given a parameter $\varepsilon > 0$, we present simple algorithms that run either in time $O\left(\frac{n}{\varepsilon^d} \log(\frac{\Delta}{\omega^* \varepsilon})\right)$ or in $O\left(\left(\frac{n \log n}{\varepsilon^{d-2}} + \frac{n}{\varepsilon^{d-1}}\right) \log\left(\frac{\Delta}{\omega^* \varepsilon}\right)\right)$ for computing a shell that contains S and whose width is at most $(1+\varepsilon)\omega^*$, where ω^* is the width of $\mathcal{A}^*(S)$ and $\Delta = \mathrm{diam}(S)$ (Section 4). If the middle radius (i.e., average of the inner and outer radii) of $\mathcal{A}^*(S)$ is at most $U \cdot \mathrm{diam}(S)$, then the running time of the algorithm is $O((n/\varepsilon^d) \log U)$. In most of the practical situations, U is a constant. For example, if the input points span an angle of at least θ with respect to the center of $\mathcal{A}^*(S)$, $U = O(1/\theta)$.

A main idea used in the algorithm is the observation that, in the plane, the minimum-area annulus containing S can be used to approximate $\mathcal{A}^*(S)$, and while this approximation might not always be good, it can at least be computed in linear time (using linear programming). We show how to refine this idea (and extend it to higher dimensions) to achieve the bounds stated above.

(iv) For $d = 3$, we present an $O(n^{3-1/19+\varepsilon})$-time algorithm for the exact computation of $\mathcal{A}^*(S)$ (Section 5). Although our algorithm is complicated, it is the first subcubic algorithm for this problem.

2 Geometric Preliminaries

Let S be a set of n points in $\mathbb{R}^d$. For a point $p \in \mathbb{R}^d$, let $\mathrm{r}(p)$ (resp. $\mathrm{R}(p)$) denote the distance between p and its nearest (resp. farthest) neighbor in S. $\mathcal{A}(p, r(p), R(p))$ is the shell of smallest width that is centered at p and contains S, which we denote by $\mathcal{A}(p)$. (In what follows, unless we consider the problem specifically in the plane, we will use the term "shell" to refer to a spherical shell in dimension higher than two and to an annulus in two dimensions.) Set $\omega(p) = \mathrm{R}(p) - \mathrm{r}(p)$ and $r_{\mathrm{mid}}(p) = (\mathrm{R}(p) + \mathrm{r}(p))/2$. We put $\omega^* = \omega^*(S) = \inf_{p \in \mathbb{R}^d} \omega(p)$ and denote by $\mathcal{A}^* = \mathcal{A}^*(S)$ a shell of width ω^* containing S. Note that the ω^* may not be attained by any finite point, in which case $\mathcal{A}^*(S)$ is a slab enclosed between two parallel planes, and $\omega^*(S)$ is then the standard *width* of S. See Figure 2 for an illustration of this case. The following lemma states two simple but usefulproperties of $r_{\mathrm{mid}}(p)$.

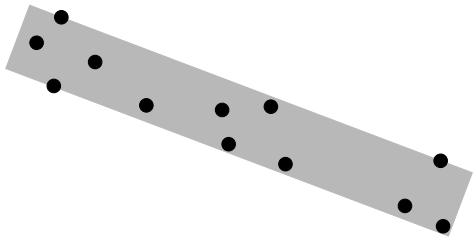

Figure 2: The minimum-width annulus is realized by a center at infinity

Lemma 2.1 *Let S be a set of points in $\mathbb{R}^d$. For any $p, q \in \mathbb{R}^d$, we have the following:*

(i) $r_{\mathrm{mid}}(p) \geq \mathrm{R}(p)/2 \geq \mathrm{diam}(S)/4$.

(ii) $|r_{\mathrm{mid}}(p) - r_{\mathrm{mid}}(q)| \leq d(p,q) \leq r_{\mathrm{mid}}(p) + r_{\mathrm{mid}}(q)$.

Proof: (i) is trivial to prove. To show (ii), use the inequalities $\mathrm{r}(p) \leq d(p,q) + r(q)$, $R(p) \leq d(p,q) + R(q)$, $d(p,q) \leq r(p) + R(q)$, whose proofs are straightforward. ■

Let $\mathrm{Vor}_N(S)$ (resp. $\mathrm{Vor}_F(S)$) denote the nearest-neighbor (resp. farthest-neighbor) Voronoi diagram of S. For $d = 2$, let $\mathrm{Vor}_N(S, \ell)$ denote the nearest-neighbor Voronoi diagram of S restricted to a line

ℓ. That is, $\mathrm{Vor}_N(S,\ell)$ is the partition of ℓ into maximal intervals so that for each interval I the same point of S is closest to all points in I. The vertices of $\mathrm{Vor}_N(S,\ell)$ are the intersection points of ℓ with the edges of $\mathrm{Vor}_N(S)$. We can obviously compute $\mathrm{Vor}_N(S,\ell)$ in $O(n\log n)$ time by first computing the entire $\mathrm{Vor}_N(S)$ and then intersecting ℓ with the diagram. However, $\mathrm{Vor}_N(S,\ell)$ can be computed directly, in $O(n\log n)$ time, using a considerably simpler algorithm; see e.g. [26]. We define $\mathrm{Vor}_F(S,\ell)$ analogously; it can also be computed directly in $O(n\log n)$ time.

3 A 2-Approximation Algorithm in the Plane

Let S be a set of n points in the plane. We present an $O(n\log n)$-time algorithm that computes an annulus containing S whose width is at most $2\omega^*$.

3.1 Algorithm

We first compute the width $width(S)$ of S (i.e., the minimum distance between a pair of parallel lines that contain S between them). Next, we compute a diametral pair of S, i.e., a pair $p,q \in S$ such that $d(p,q) = \mathrm{diam}(S) \equiv \max_{p',q'\in S} d(p',q')$. Both of these steps take $O(n\log n)$ time. Let ℓ be the perpendicular bisector of pq. We compute $\mathrm{Vor}_N(S,\ell)$ and $\mathrm{Vor}_F(S,\ell)$, merge the vertices of the two diagrams into a single sorted list V, and compute the point v^* that minimizes $\omega(v)$ over all $v\in\ell$. The latter stages can be done in $O(|V|)$ time, because $\omega(v)$ depends on a fixed pair of points of S between any pair of successive points of V. If $width(S) \geq \omega(v^*)$, we return $\mathcal{A}(v^*)$; otherwise, we return a strip of width $width(S)$ that contains S. The algorithm obviously returns an annulus that contains S, and it runs in $O(n\log n)$ time. We next prove that

$$\min\{\omega(v^*), width(S)\} \leq 2\omega^*.$$

Theorem 3.1 *The width of the annulus computed by the above algorithm is at most* $2\omega^*$.

Let $\Delta = \mathrm{diam}(S)$. Let C_O and C_I be the outer and inner circles of an annulus $\mathcal{A}^*$ of width ω^* that contains S, and let c be the center of $\mathcal{A}^*$. Let p,q be the diametral pair computed by the algorithm. Without loss of generality, we can assume that c is the origin, $p=(0,1)$, $1 = d(c,p) \geq d(c,q)$, and $x(q)\geq 0$ (see Figure 3). Let D be the circle of radius $d(p,q) = \Delta$ centered at p.

Lemma 3.2 *If $\Delta \leq 1$, then the width of the horizontal strip containing S is at most $\omega^* + \Delta^2/2$.*

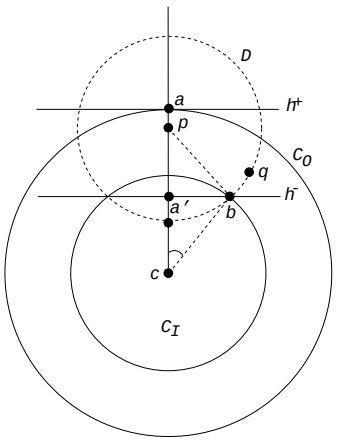

Figure 3: The minimum-width annulus and the strip defined by h^-, h^+.

Proof: Let a be the topmost point of C_O. Since $\Delta \leq 1$, $c \notin \mathrm{int}(D)$, which implies that either D lies fully above C_I (i.e., the horizontal line passing through the topmost point of C_I separates D and C_I) or ∂D and C_I intersect at two points with positive y-coordinates. (The case where the two circles touch can be handled by essentially the same argument.) The first situation is impossible: since $S\subseteq D$, we can grow C_I and still have S lie in the shrunken annulus, contrary to the minimality of $\mathcal{A}^*$. Let b be the intersection point of ∂D and C_I lying to the right of the y-axis. Let h^-, h^+ be the horizontal lines passing through b and a, respectively. Since $S\subseteq \mathcal{A}^*\cap D$, the strip bounded by h^-, h^+ contains S; see Figure 3. Let a' be the intersection point of h^- and the y-axis. Then

$$\begin{aligned} d(a',c) &= d(c,b)\cos(\angle bcp) \\ &= d(c,b)\frac{d(p,c)^2 + d(c,b)^2 - d(p,b)^2}{2d(p,c)d(c,b)} \\ &= \frac{1+r_I^2-\Delta^2}{2}, \end{aligned}$$

where r_I is the radius of C_I. Therefore the width of the strip is

$$\begin{aligned} d(a,c) - d(a',c) &= r_I + \omega^* - \frac{1+r_I^2-\Delta^2}{2} \\ &= \omega^* + \frac{\Delta^2}{2} - \frac{(1-r_I)^2}{2} \leq \omega^* + \frac{\Delta^2}{2}. \end{aligned}$$

■

Hence, if $\Delta \leq 1$ and $\omega^* \geq \Delta^2/2$, the algorithm computes an annulus (that is, a strip) of width at most $2\omega^*$. We now assume that either $\Delta > 1$ or $\omega^* < \Delta^2/2$.

Let C_{pq} be the circle that passes through p and q and whose center ξ lies on the y-axis; see Figure 4.

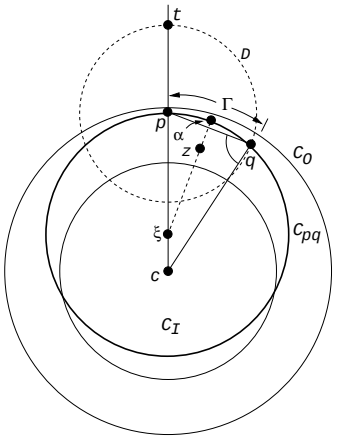

Figure 4: The minimum-width annulus and the circle C_{pq}.

We will show that all points of S lie within distance ω^* from C_{pq}, which implies that the annulus centered at ξ with the inner radius $d(\xi,p)-\omega^*$ and the outer radius $d(\xi,p)+\omega^*$ contains S. Since ξ lies on the perpendicular bisector of pq, the thinnest annulus that the algorithm computes is certainly no wider than $\mathcal{A}(\xi)$, i.e., its width is at most $2\omega^*$.

Since $d(c,p) \geq d(c,q)$, C_{pq} lies inside the circle passing through p and centered at c, and therefore it also lies inside C_O. But C_{pq} may intersect C_I (as in Figure Figure 4). Let $\Gamma \subseteq C_{pq}$ be the circular arc from p to q in the clockwise direction. A simple calculation shows that the distance from c to the points of Γ is monotonically decreasing along Γ. Since $p,q \in \mathcal{A}^*$, the entire arc Γ lies inside $\mathcal{A}^*$.

Lemma 3.3 *If $\Delta > 1$ or $\omega^* < \Delta^2/2$, then $\angle pqc < \pi/2$.*

Proof: If $\Delta > 1$, then $c \in \text{int}(D)$. We then have $\angle pqc < \angle pqm$, where m is the bottom-most point of D, and the latter angle is obviously less than $\pi/2$; see Figure 4. Next, assume that $\omega^* < \Delta^2/2$. Since $d(c,p)=1$, $d(p,q)=\Delta$, and $1 \geq d(c,q) \geq 1-\omega^*$, we obtain

$$\begin{aligned}\cos(\angle pqc) &= \frac{d(p,q)^2+d(c,q)^2-d(c,p)^2}{2d(p,q)d(c,q)} \\ &= \frac{\Delta^2+d(c,q)^2-1}{2\Delta d(c,q)} \geq \frac{\Delta^2+(1-\omega^*)^2-1}{2\Delta} \\ &= \frac{\Delta^2-2\omega^*+{\omega^*}^2}{2\Delta} > 0.\end{aligned}$$

The last inequality follows from the assumption that $\omega^* < \Delta^2/2$. This completes the proof of the lemma. ■

We now prove that for any point $z \in S$, the distance between C_{pq} and z, denoted by $d(z,C_{pq})$, is at most ω^*. We will prove the claim for points with positive x-coordinates; the same argument applies to points with negative x-coordinates. Let α be the intersection point of C_{pq} with the ray emanating from ξ in direction $\vec{\xi z}$; see Figure 4. Then $d(z,C_{pq}) = d(z,\alpha)$.

If $z \in \text{int}(C_{pq})$, then let β be the intersection point of C_{pq} with the ray emanating from z in direction $\vec{cz}$ (see Figure 5); otherwise, let β be the intersection point of C_{pq} with the ray emanating from z in direction $\vec{zc}$. The point β exists since c lies inside C_{pq}, as $\angle pqc < \pi/2$. Since α lies on the line passing through z and the center of C_{pq}, $d(z,\alpha) \leq d(z,\beta)$.

Lemma 3.4 $d(z,\beta) < \omega^*$.

Proof: We will prove that β lies in the annulus $\mathcal{A}^*$. Let z' be the intersection point of D with the ray emanating from z in direction $\vec{cz}$. For two points $x,y \in D$, let $D[x,y] \subseteq D$ denote the circular arc from x to y in the clockwise direction. Let t be the topmost point of D. There are two cases to consider:

Case (i) $z' \in D[t,q]$. By Lemma 3.3, $\angle pqc < \pi/2$, therefore $D[t,q]$ lies in the wedge formed by the positive y-axis and the ray emanating from c in direction $\vec{cq}$. This in turn implies that $\beta \in \Gamma$ irrespective of whether z lies inside or outside C_{pq}; see Figure 5(i). As noted earlier, $\Gamma \subset \mathcal{A}^*$, so $\beta \in \mathcal{A}^*$, as claimed.

Case (ii) $z' \notin D[t,q]$. Note that q is an intersection point of circles D and C_{pq} and their second point of intersection is the mirror image of q on the other side of y-axis. Therefore the portion of D between q and its bottom-most point lies inside C_{pq}. Since z' has positive x-coordinate and $z' \notin D[t,q]$, z' lies on the portion of ∂D inside C_{pq}. Therefore β lies after z' in direction $\vec{cz}$ (see Figure 5(ii)) and

$$r_I \leq d(c,z) \leq d(c,z') < d(c,\beta) < r_O,$$

where the last inequality follows from the fact that $C_{pq} \subset \text{int}(C_O)$. This implies that $\beta \in \mathcal{A}^*$, as desired.

We thus have $d(z,\beta) < \omega^*$. ■

Lemmas 3.2 and 3.4 imply the theorem.

4 A $(1+\varepsilon)$-Approximation Algorithm in Any Dimension

Let S be a set of n points in $\mathbb{R}^d$. Set $\Delta = \text{diam}(S)$. We will first describe an approximation algorithm for computing the thinnest shell $\mathcal{A}(p)$ containing S with

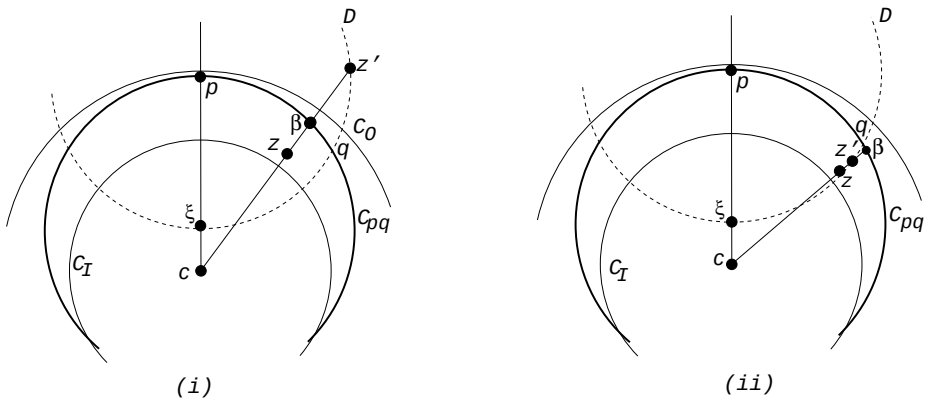

Figure 5: Illustration of the proof of Lemma 3.4. (i) $z' \in D[a,q]$, (ii) $z' \notin D[a,q]$.

the constraint that $r_{\mathrm{mid}}(p) = (\mathrm{r}(p) + \mathrm{R}(p))/2 \leq U \cdot \Delta$ for some given parameter $U \in \mathbb{R}$. Let $\mathcal{A}^*(S,U)$ denote this constrained minimum-width shell, and let $\omega^*(S,U)$ denote the width of $\mathcal{A}^*(S,U)$. Computing $\mathcal{A}^*(S,U)$ can be formulated as the following optimization problem in the variables $(x_1, x_2, \ldots, x_d, r, R)$: Find

$$\begin{aligned} &\min\ R - r \\ r \leq \left(\sum_{i=1}^{d} (x_i - p_i)^2\right)^{1/2} &\leq R \qquad \forall p = (p_1, \ldots, p_d) \in S \\ r + R &\leq 2U\Delta. \end{aligned}$$

Let C be a d-dimensional hyper-rectangle of the form $\prod_{i=1}^{d} [\alpha_i, \beta_i]$. We define another constrained shell $\mathcal{E}(S,C)$ (which becomes, when $d = 2$, the minimum-area annulus containing S with center constrained to lie in C), in the same variables, as follows:

$$\begin{aligned} &\min\ R^2 - r^2 \\ r \leq \left(\sum_{i=1}^{d} (x_i - p_i)^2\right)^{1/2} &\leq R \qquad \forall p = (p_1, \ldots, p_d) \in S \\ \alpha_i \leq x_i &\leq \beta_i \qquad 1 \leq i \leq d. \end{aligned}$$

If we substitute Σ for $R^2 - \sum_{i=1}^{d} x_i^2$ and σ for $r^2 - \sum_{i=1}^{d} x_i^2$, then $\Sigma - \sigma = R^2 - r^2$, and we can restate the optimization problem defining $\mathcal{E}(S,C)$ as:

$$\begin{aligned} &\min\ \Sigma - \sigma \\ \sigma \leq -\sum_{i=1}^{d} 2p_i x_i + \sum_{i=1}^{d} p_i^2 &\leq \Sigma \qquad \forall p = (p_1, \ldots, p_d) \in S \\ \alpha_i \leq x_i &\leq \beta_i \qquad 1 \leq i \leq d. \end{aligned}$$

This is, however, an instance of linear programming with $d+2$ variables, and can be solved in $O(n)$ time [14, 24], assuming d to be a fixed constant. Let $\hat{\omega}(S,C)$ denote the width of $\mathcal{E}(S,C)$.

We now describe our approximation algorithm. Let
$C(p,s)$ be the d-dimensional axis-parallel cube of side length s and centered at p.

Algorithm APPROX_SHELL (S, U, ε)

1. Compute $\mathcal{E}(S, \mathbb{R}^d)$. If $\hat{\omega}(S, \mathbb{R}^d) = 0$, then return $\mathcal{E}(S, \mathbb{R}^d)$.
2. Pick a point $o \in S$ and set $\mathfrak{C} = C(o, (2U+2)\Delta)$.
3. Partition $\mathfrak{C}$ into a collection $\mathcal{C} = \{C_1, \ldots, C_k\}$ of axis-parallel cubes so that, for all points p, q inside the same cube C_i, $r_{\mathrm{mid}}(p) \leq (1+\varepsilon) r_{\mathrm{mid}}(q)$. (An efficient method of doing this is given below.)
4. For each $C_i \in \mathcal{C}$, compute $A_i = \mathcal{E}(S, C_i)$.
5. Return the thinnest shell among $A_1, \ldots, A_k$.

Lemma 4.1 APPROX_SHELL(S, U, ε) *returns a shell whose width is at most* $(1+\varepsilon)\omega^*(S,U)$.

Proof: If $\hat{\omega}(S, \mathbb{R}^d) = 0$, then the lemma is obvious. Otherwise, let p be the center of $\mathcal{A}^*(S,U)$. Since $r_{\mathrm{mid}}(o) \leq \mathrm{R}(o) \leq \Delta$ and $r_{\mathrm{mid}}(p) \leq U\Delta$, we have, by Lemma 2.1(ii), that $p \in \mathfrak{C}$. Let C_i be the cube containing p. Let $q \in C_i$ be the center of $\mathcal{E}(S, C_i)$. Then

$$\begin{aligned} \mathrm{R}^2(q) - \mathrm{r}^2(q) &\leq \mathrm{R}^2(p) - \mathrm{r}^2(p), \text{ or} \\ r_{\mathrm{mid}}(q)\omega(q) &\leq r_{\mathrm{mid}}(p)\omega(p). \end{aligned}$$

Equivalently,

$$\omega(q) \leq \frac{r_{\mathrm{mid}}(p)}{r_{\mathrm{mid}}(q)} \omega(p) \leq (1+\varepsilon)\omega^*(S,U).$$

■

We now describe how to partition $\mathfrak{C}$ into the collection $\mathcal{C}$. A similar construction is described in [20].

Lemma 4.2 *Let U, ε be two positive numbers. Then $\mathfrak{C} = C(o, (2U+2)\Delta)$ can be partitioned into a set $\mathcal{C}$ of $O((1/\varepsilon)^d \log U)$ cubes so that in each cube C of the partition, $r_{\mathrm{mid}}(p) \leq (1+\varepsilon) r_{\mathrm{mid}}(q)$ for all $p, q \in C$. This tiling can be computed in $O(n + (1/\varepsilon)^d \log U)$ time.*

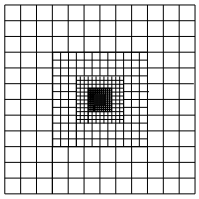

Figure 6: Tiling of $\mathfrak{C}$.

Proof: Let μ be a real number such that $\Delta/2 \leq \mu \leq \Delta$. (See [17] for a simple $O(n)$ algorithm for approximating the diameter to within a factor of $\sqrt{3}$ in any dimension. Alternatively, fix any $p \in S$ and take $\mu = \mathrm{R}(p) \geq \Delta/2$, by Lemma 2.1(i).)

Set $m = \lceil \log_2(U+1) \rceil$. For $i = 1, \ldots, m$, we define

$$B_0 = C(o, 4\mu), \quad B_i = C(o, 2^{i+2}\mu) \setminus C(o, 2^{i+1}\mu).$$

We can tile B_0 by $O(1/\varepsilon^d)$ axis-parallel cubes having side length $r_0 = \mu\varepsilon/(4\sqrt{d})$. Let C be a cube in this tilling. For $p, q \in C$, we have, by Lemma 2.1,

$$\begin{aligned} r_{\mathrm{mid}}(p) &\leq r_{\mathrm{mid}}(q) + d(p,q) \leq r_{\mathrm{mid}}(q) + \varepsilon\mu/4 \\ &\leq (1+\varepsilon) r_{\mathrm{mid}}(q), \end{aligned}$$

since $r_{\mathrm{mid}}(q) \geq \Delta/4 \geq \mu/4$.

Let $r_i = 2^i \mu\varepsilon/\sqrt{d}$, for $i = 1, \ldots, m$. B_i can be tiled by

$$O\left(\left(\frac{2^{i+2}\mu}{r_i}\right)^d\right) = O\left(\left(\frac{2^{i+2}\mu}{2^i\mu\varepsilon/\sqrt{d}}\right)^d\right) = O\left(\frac{1}{\varepsilon^d}\right)$$

axis-parallel cubes with side length r_i, for $i = 1, \ldots, m$.

Let C be a cube in this tiling of B_i, and let p, q be two points in C. Using Lemma 2.1 and the fact that $r_{\mathrm{mid}}(o) \leq \Delta \leq 2\mu$, we have

$$r_{\mathrm{mid}}(q) \geq d(q,o) - r_{\mathrm{mid}}(o) \geq 2^{i+1}\mu - 2\mu \geq 2^i\mu.$$

We also have

$$\begin{aligned} r_{\mathrm{mid}}(p) &\leq r_{\mathrm{mid}}(q) + d(q,p) \leq r_{\mathrm{mid}}(q) + \sqrt{d} r_i \\ &= r_{\mathrm{mid}}(q) + 2^i\mu\varepsilon \leq r_{\mathrm{mid}}(q)(1+\varepsilon). \end{aligned}$$

See Figure 6 for an illustration of the resulting tiling. This completes the proof of the lemma, since B_m contains $\mathfrak{C}$ and the total number of cubes is $O((1/\varepsilon^d)\log U)$. (The bound on the running time of this construction is trivial.) ∎

Theorem 4.3 *Let S be a set of n points in $\mathbb{R}^d$, $\varepsilon > 0$, and $U > 0$. One can compute, either in $O((n/\varepsilon^d)\log U)$ or in $O\left(\left(\frac{n \log n}{\varepsilon^{d-2}} + \frac{n}{\varepsilon^{d-1}}\right)\log U\right)$ time, a shell $\mathcal{A} \supset S$ whose width is at most $(1+\varepsilon)\omega^*(S,U)$.*

Proof: The first bound is a consequence of the preceding discussion. The second bound follows by observing that the execution of the algorithm of Approx_Shell can be interpreted as follows: We compute a sequence of cubes $\mathcal{C}_1, \ldots, \mathcal{C}_m$, where $m = O(\log U)$. Each such cube is decomposed into $O(1/\varepsilon^d)$ subcubes using an appropriate uniform grid. For each subcube C we obtain $\mathcal{E}(S, C_i)$ as a solution of an appropriate linear programming problem.

Let $\mathcal{C}_i$ be such a cube, and let $V = \{C_1, \ldots, C_\mu\}$ be the resulting decomposition of $\mathcal{C}_i$ into subcubes. The linear programming instances on each C_j are almost identical except for the $2d$ inequalities restricting the solution to lie inside C_j. This implies that, with the possible exception of one subcube, the solutions to all those linear programming instances must lie on the boundaries of the respective cubes $C_i, \ldots, C_\mu$. Moreover, the solution of the at most one instance of the linear programming that does lie inside its cube, can be computed directly, by solving a single linear-programming instance, without restricting the location of the solution to any subcube (i.e. by dropping the inequalities $\alpha_i \leq x_i \leq \beta_i$).

In particular, we conclude that we can reduce the d-dimensional problem to a $(d-1)$-dimensional problem, as follows:

- Solve the unrestricted version of the linear programming (i.e., compute the global "minimum area" shell).
- For each axis-parallel $(d-1)$-dimensional hyperplane H of the grid defining the decomposition V, find recursively a $(1+\varepsilon)$-approximate shell containing S whose center is constrained to lie on $H \cap \mathcal{C}_i$. There are $O(d/\varepsilon)$ such hyperplanes.
- Return the shell of minimum width among all those generated by the algorithm.

The recursion bottoms out at $d = 2$, where we proceed as follows. Let H be our two-dimensional plane. We can compute in $O(n \log n)$ time the maps induced on H by the d-dimensional nearest- and furthest-neighbor Voronoi diagrams of S (those maps are called power diagrams [8], they have linear complexity, and they can be computed in $O(n \log n)$ time). Our target is to approximate the minimum difference between the farthest and nearest neighbors of points on H (i.e. this is the width of the minimum-width

shell whose center is restricted to lie on H). We note that we can compute this minimum along a line ℓ in $O(n)$ time, by performing a walk through the overlay of those two diagrams along ℓ. We do this along each line of the grid, and also solve the global linear-programming instance where the center of the shell is restricted to lie on H. Thus, we can solve a two-dimensional instance in $O(n \log n + n/\varepsilon)$ time.

Overall, the running time of the recursive algorithm for the subcubes of $\mathcal{C}_i$ is $O((n/\varepsilon^{d-2}) \log n + n/\varepsilon^{d-1})$ time. Thus, solving all the linear programming instances for $\mathcal{C}_1, \ldots, \mathcal{C}_m$ requires

$$O\left(\left(\frac{n \log n}{\varepsilon^{d-2}} + \frac{n}{\varepsilon^{d-1}}\right) \log U\right)$$

time. ■

Even though Theorem 4.3 is not satisfying from a strict theoretical point of view, for all practical purposes the assumptions in the theorem are reasonable. For example, in the plane, if the points in S span an angle of at least $\theta \in [0, \pi/2]$ with respect to the center c of $\mathcal{A}^*(S)$, then $r_{\text{mid}}(c) = O(\Delta/\theta)$. In this case we can compute an annulus of width at most $(1+\varepsilon)\omega^*(S)$ containing S in time $O(\frac{n}{\varepsilon^2} \log \frac{1}{\theta})$.

Remark 4.4 For $d = 2$ the algorithm of Theorem 4.3 can be further simplified and improved, by noting that in this case the power diagrams are (regular) nearest- and furthest-neighbor Voronoi diagrams, and that they need to be computed only once. The running time of the resulting algorithm is thus $O\left(n \log n + \frac{n}{\varepsilon} \log U\right)$.

Remark 4.5 If $n \gg \exp(1/\varepsilon^2)$ then the first bound of Theorem 4.3 is better than the second bound. Such values of n are realistic only when ε is very coarse. This remark should be taken into account also for the following algorithms.

We next modify the algorithm Approx_Shell so that it produces in all cases a shell containing S of width $\leq (1+\varepsilon)\omega^*(S)$.

Lemma 4.6 *For $U > 5$ we have*

$$\omega^*(S, U) \leq \omega^*(S) + \frac{8\text{diam}(S)}{U}.$$

Proof: Let $\mathcal{A}^*$ be a minimum-width shell containing S, with center p and width $\omega^* = \omega^*(S)$. Put $\Delta = \text{diam}(S)$. If $\omega^*(S, U) \neq \omega^*(S)$ then $r_{\text{mid}}(p) > U\Delta$, so assume this is the case.

Let $\mathcal{V}$ be a circular cone centered at p, containing S, and having the smallest opening angle. Let

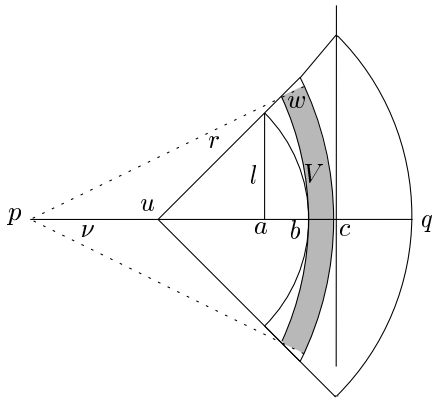

Figure 7: Construction for the proof of Lemma 4.6.

$V = \mathcal{V} \cap \mathcal{A}^*$. Since $r_{\text{mid}}(p) > 5\Delta$, $\mathcal{V}$ spans less than a halfspace. Let ν be the ray which is the axis of symmetry of $\mathcal{V}$. Refer to Figure 7. Let b and c be the points where ν meets the inner and outer spheres of $\mathcal{A}^*$, respectively. Let u be a point on the segment pb at distance $r = U\Delta/2$ from b. Let $\mathcal{W}$ be the smallest cone centered at u, with axis of symmetry along ν and containing V. Consider the portion of $\mathcal{W}$ lying on the same side as p and u of the hyperplane through c and orthogonal to ν, and let R denote the maximum distance from u to a point in this portion. The shell $\mathcal{A}'$ centered at u with radii r and R, encloses V and thus also covers S. We now estimate $\omega(u)$ by upper bounding the width of $\mathcal{A}'$.

Let q be the point on $\mathcal{V}$ at distance R from u, as pictured. We have $\omega(u) \leq \omega^* + d(c, q)$. However, $d(u, a) = \sqrt{r^2 - l^2}$ and

$$d(a, b) = r - \sqrt{r^2 - l^2} = \frac{l^2}{r + \sqrt{r^2 - l^2}} \leq \frac{l^2}{r}.$$

By similarity, we have $d(c, q) = d(a, b)\frac{r + \omega^*}{d(u, a)}$.

Note that $\omega^* < \Delta < r$ and that $l \leq \Delta = 2r/U \leq 2r/5$. To see the latter inequality, project S, centerally towards u, on the sphere σ of radius r about u. The image $\hat{S}$ of S falls inside the cap $\sigma \cap \mathcal{W}$, which is a smallest cap on σ enclosing $\hat{S}$. Since the projection does not increase the distances between points, the diameter of $\hat{S}$ is at most Δ, which is easily seen to imply that $l \leq \Delta$. This implies that $d(u, a) = \sqrt{r^2 - l^2} \geq r\sqrt{1 - \frac{4}{25}} \geq r/2$. Hence, we have

$$d(c, q) \leq \frac{l^2}{r} \cdot \frac{2r}{r/2} = \frac{4l^2}{r}.$$

Overall,

$$\begin{aligned}\omega(u) &\leq \omega^* + d(c,q) \leq \omega^* + \frac{4l^2}{r} \leq \omega^* + \frac{4\Delta^2}{r} \\ &\leq \omega^* + \frac{4\Delta^2}{U\Delta/2} = \omega^* + \frac{8\Delta}{U}.\end{aligned}$$

Note that

$$\begin{aligned}r_{\mathrm{mid}}(u) &= r + \frac{w(u)}{2} \leq r + \frac{\omega^*}{2} + \frac{4\Delta}{U} < \frac{3r}{2} + \frac{4\Delta}{U} \\ &= \Delta\left(\frac{3U}{4} + \frac{4}{U}\right) < U \cdot \Delta.\end{aligned}$$

Hence $\omega^*(S,U) \leq w(u) \leq \omega^* + \frac{8\Delta}{U}$, as asserted. ■

Corollary 4.7 *Let $\varepsilon > 0$, $U > 5$ be two positive constants. One can compute in $O\left(\left((n/\varepsilon^{d-2})\log n + n/\varepsilon^{d-1}\right)\log U\right)$ or $O(n/\varepsilon^d \log U)$ time a shell of width at most $(1+\varepsilon/4)[\omega^*(S) + \frac{8\Delta}{U}]$ that contains S, where $\Delta = \mathrm{diam}(S)$.*

Finally, we describe the general approximation algorithm. Let APPROX_DIAM(S) be the procedure that computes in linear time a $\sqrt{3}$-approximation Δ_0 of $\Delta(S) = \mathrm{diam}(S)$ [17].

Algorithm APPROX_SHELL_2 (S, ε)

$\omega = \Delta_0 =$ APPROX_DIAM (S); $\quad \omega_{\mathrm{old}} = \infty$;
while $\omega < \omega_{\mathrm{old}}/2$ **do**
$\quad U = \frac{50\sqrt{3}\Delta_0}{\varepsilon} \cdot \frac{1}{\omega}$;
$\quad A(p) =$ APPROX_SHELL(S, U, $\varepsilon/8$);
$\quad \omega_{\mathrm{old}} = \omega$; $\quad \omega = \omega(p)$;
end while
return ($A(p)$);

Theorem 4.8 *Given a set S of n points in $\mathbb{R}^d$ and a parameter $0 < \varepsilon < 1$, APPROX_SHELL_2 computes a shell of width at most $(1+\varepsilon)\omega^*(S)$. With an appropriate optimization of the calls to APPROX_SHELL, the running time is $O\left(\frac{n}{\varepsilon^d}\log(\frac{\Delta}{\omega^*(S)\varepsilon})\right)$ or $O\left(\left(\frac{n\log n}{\varepsilon^{d-2}} + \frac{n}{\varepsilon^{d-1}}\right)\log(\frac{\Delta}{\omega^*(S)\varepsilon})\right)$.*

Proof: Suppose the while loop is executed m times. Let ω_i, U_i be the values of ω and U computed in the i-th iteration of the loop. Then, putting $\omega^* = \omega^*(S)$,

$$\begin{aligned}\omega_m &\leq (1+\varepsilon/8)\,\omega^* + (1+\varepsilon/8)\,\frac{8\Delta}{U_m} \\ &\leq (1+\varepsilon/8)\,\omega^* + (1+\varepsilon/8)\,\frac{8\Delta}{50\sqrt{3}\Delta_0/(\omega_{m-1}\varepsilon)} \\ &\leq (1+\varepsilon/8)\,\omega^* + (1+\varepsilon/8)\,\frac{4\varepsilon\omega_{m-1}}{25} \\ &\leq (1+\varepsilon/8)\,\omega^* + \frac{9\varepsilon\omega_m}{25},\end{aligned}$$

by Lemma 4.6, and since $w_m \geq w_{m-1}/2$. Thus,

$$\omega_m \leq \frac{1+\varepsilon/8}{1-9\varepsilon/25}\omega^* \leq (1+\varepsilon)\omega^*.$$

Note that for all $i < m$ we have $\omega_i < \frac{\Delta_0\sqrt{3}}{2^i}$. Hence, $\omega^* \leq \omega_{m-1} \leq \frac{\Delta_0\sqrt{3}}{2^{m-1}}$, implying that $m = O(\log\frac{\Delta}{\omega^*})$ and $U_m = O(\Delta/(\omega^*\varepsilon))$.

Note that the i-th call to APPROX_SHELL (executed, say, by the first algorithm of Theorem 4.3) constructs a tiling of $\mathfrak{C}_i = C(o, (2U_i+2)\Delta)$, and computes $\mathcal{E}(S,C)$ for each cube C in this tiling. By modifying the algorithm so that it computes $\mathcal{E}(S,C)$ only for the new cubes C in the tiling (that is, ignoring cubes that are covered by cubes produced in earlier iterations), it follows that the running time of the i-th iteration can be improved to $O\left(\frac{n}{\varepsilon^d}\left(1+\log\frac{U_i}{U_{i-1}}\right)\right)$, for $i = 2, \ldots, m$. Overall, the running time of the algorithm is thus

$$\begin{aligned}&O\left(\frac{n}{\varepsilon^d}\log U_1 + \sum_{i=2}^{m}\frac{n}{\varepsilon^d}\left(1+\log\frac{U_i}{U_{i-1}}\right)\right) \\ &\quad = O\left(\frac{n}{\varepsilon^d}\left(m + \log U_m\right)\right) = O\left(\frac{n}{\varepsilon^d}\log\frac{\Delta}{\omega^*\varepsilon}\right).\end{aligned}$$

The other time bound follows if we execute APPROX_SHELL using the second algorithm of Theorem 4.3. ■

Remark 4.9 Note that the exponential search performed by APPROX_SHELL_2, for the "true" width is not necessary in the planar case: Simply put $U = \frac{8\Delta}{\varepsilon\omega}$, where ω is the 2-approximate width provided by Theorem 3.1, and apply Corollary 4.7. Somewhat surprisingly, this does not improve the running time of APPROX_SHELL_2.

Nevertheless, one can ε-approximate the minimum-width annulus in the planar case, in strongly polynomial time $O(n\log n + n/\varepsilon^2)$ (which does not depend on Δ and ω^*), by combining the algorithm of Theorem 3.1 with APPROX_SHELL_2. This is done in the following subsection.

4.1 A strongly polynomial ε-approximation algorithm for the minimum-width annulus in the plane

In this subsection, we present a strongly polynomial ε-approximation algorithm for the minimum-width annulus. The algorithm is a combination of the approximation techniques developed in the previous subsections.

Algorithm PLANAR_APPROX_SHELL

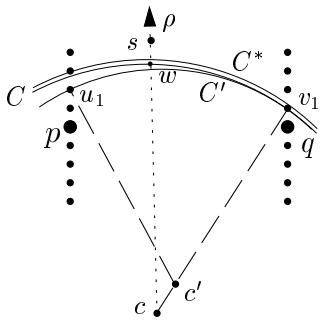

Figure 8: Proof of correctness of Planar_Approx_Shell

1. Run the 2-approximation algorithm of Theorem 3.1. Let A' be the resulting annulus. If $w' = w(A') = 0$ then return A'.

2. Compute the nearest- and farthest-neighbor Voronoi diagrams $\mathrm{Vor}_F(S), \mathrm{Vor}_N(S)$, in $O(n \log n)$ time.

3. Compute, in $O(n \log n + (n/\varepsilon) \log U)$ time, an annulus A'' of width $\leq (1+\varepsilon) w^*(S, U)$, using the algorithm of Remark 4.4, with $U = 10000/\varepsilon$. (Either A'' is the required ε-approximation, or $r_{\mathrm{mid}}(\mathcal{A}^*(S)) > U\Delta(S)$.)

4. Compute, in $O(n \log n)$ time, a pair of points $p, q \in S$ that realize the diameter of S. We assume without loss of generality that $p = (-1, 0), q = (1, 0)$. Let $\delta = \varepsilon w'/20$, Let $P_p = P(p, \delta, w')$, $P_q = P(q, \delta, w')$, where

$$P(p, \delta, w') = \left\{ p + (0, \delta) i \;\middle|\; i = -\lceil 20/\varepsilon \rceil, \ldots, \lceil 20/\varepsilon \rceil \right\}.$$

   See Figure 8.

5. For each pair $u \in P_p, v \in P_q$ compute the minimum-width annulus whose center lies on the perpendicular bisector of uv. Using the precomputed $\mathrm{Vor}_F(S)$ and $\mathrm{Vor}_N(S)$, this takes $O(n)$ time per pair, as in the algorithm of Theorem 4.3.

6. Output the minimum width annulus among those computed.

Theorem 4.10 *The width of the annulus output by* Planar_Approx_Shell *is* $\leq (1+\varepsilon)\omega^*(S)$*, and the running time of the algorithm is* $O(n \log n + n/\varepsilon^2)$.

Proof: If $r_{\mathrm{mid}}(\mathcal{A}^*(S)) \leq U\Delta(S)$, the correctness and the bound on the running time are consequences of the previous algorithms, so assume that $r_{\mathrm{mid}}(\mathcal{A}^*(S)) > U\Delta(S)$. Let C^* be the middle circle of $\mathcal{A}^*(S)$, and let c^*, r^* denote the center and the radius of C^*, respectively. Let I_p and I_q denote the segments spanned by the points of P_p and of P_q, respectively.

It is clear that C^* crosses both I_p and I_q, at two respective points u, v. Let u_1 (resp. v_1) denote the point of P_p (resp. of P_q) that lies immediately below u (resp. v). We first translate C^* downwards, till it first hits either u_1 or v_1. Suppose, without loss of generality, that it first hits v_1. Let C denote the translated circle. Clearly, the center c of C lies vertically below c^* at distance less than δ. In particular, for any $s \in S$ we have $|d(c, s) - d(c^*, s)| \leq d(c, c^*) < \delta$. Put $\omega = 2d(C, S)$ and observe that

$$\omega < 2(d(C^*, S) + \delta) = \omega^* + 2\delta \leq (1 + \varepsilon/5)\omega^*.$$

Next, shrink C by moving its center from c towards v_1 and by keeping v_1 on the circle, until it also passes through u_1. Let C' denote the new circle and let c' denote its center. See Figure 8.

The distance from c to points on C' decreases monotonically as we traverse C' from v_1 counterclockwise till we reach the point on C' antipodal to v_1. Let s be any point of S. The ray ρ from c towards s crosses C at a point w and C' at a point w'. We have $d(w', s) \leq d(w, s) + d(w, w') \leq \omega/2 + d(w, w')$. It easily follows from the preceding discussion that $d(w, w')$ attains its maximum when w' is near u_1, and this maximum is smaller than 2δ. This implies that

$$\omega(c') \leq 2d(C', S) \leq \omega + 2\delta \leq (1 + 2\varepsilon/5)\omega^* \leq (1+\varepsilon)\omega^*.$$

Since c' lies on the perpendicular bisector of $u_1 v_1$, it follows that the width of the annulus output by the algorithm is at most $\omega(c') < (1+\varepsilon)\omega^*$, as asserted. The bound on the running time is obvious: We have $O(1/\varepsilon^2)$ bisectors to process, and the processing of each of them takes $O(n)$ time, as noted in the algorithm. ■

5 Minimum-Width Shell in Three Dimensions

Let S be a set of n points in $\mathbb{R}^3$. Combining the techniques in [1, 6] with a few new ideas, we present an $o(n^3)$-time algorithm to compute $\mathcal{A}^*(S)$. Although our algorithm is impractical, it is the first subcubic algorithm and is significant from a theoretical point of view. As mentioned in the introduction, the center of $\mathcal{A}^*(S)$ is the intersection point of either an edge of $\mathrm{Vor}_N(S)$ with a 2-face of $\mathrm{Vor}_F(S)$ or of an edge of $\mathrm{Vor}_F(S)$ with a 2-face of $\mathrm{Vor}_N(S)$. There are $\Theta(n^3)$ such intersection points in the worst case, so we cannot afford to check all of them explicitly. We will instead check them in an implicit manner. We first

sketch the overall algorithm and then describe the main steps in detail.

5.1 Overview of the algorithm

We present an algorithm to determine a minimum-width shell centered at an intersection point of an edge of $\mathrm{Vor}_N(S)$ and a 2-face of $\mathrm{Vor}_F(S)$ that contains S. The other case is treated similarly.

Algorithm MIN_WIDTH_SHELL (S)

1. Compute $\mathrm{Vor}_N(S)$. Let E be the set of edges of $\mathrm{Vor}_N(S)$.
2. Fix a parameter r and choose a random sample $R \subseteq S$ of size $O(r \log r)$. Each subset of this size is chosen with equal probability.
3. Compute $\mathrm{Vor}_F(R)$ and triangulate each Voronoi cell using the Dobkin-Kirkpatrick hierarchical algorithm [12]. Let Δ be the set of resulting tetrahedra; $|\Delta| = O(r^2 \log^2 r)$. Each tetrahedron of Δ lies inside the Voronoi cell $V_F(p_\triangle, R)$ of some point $p_\triangle \in R$.
4. For each tetrahedron $\triangle \in \Delta$ do the following:
   - Compute the set
     $$S_\triangle = \{q \in S \mid \exists z \in \triangle \; s.t. \; d(z,q) \geq d(z, p_\triangle)\}.$$
     The farthest neighbor of any point in $\triangle$ belongs to $S_\triangle$. Set $n_\triangle = |S_\triangle|$.
   - Compute $\mathrm{Vor}_F(S_\triangle)$ within $\triangle$ and triangulate each 2-face of $\mathrm{Vor}_F(S_\triangle)$. Let $T_\triangle$ be the set of resulting triangles; set $t_\triangle = |T_\triangle| = O(n_\triangle^2)$.
   - Compute the subset $E_\triangle \subseteq E$ of the edges of $\mathrm{Vor}_N(S)$ that intersect the interior of $\triangle$; set $m_\triangle = |E_\triangle|$.
5. For each $\triangle \in \Delta$, compute $\mathcal{A}_\triangle = \mathcal{A}^*(E_\triangle, T_\triangle)$, a minimum-width shell containing S among all those centered at the intersection points of edges in $E_\triangle$ and triangles in $T_\triangle$.
6. Return the thinnest shell computed in the previous step.

The correctness of the algorithm follows from the following observations. $\mathrm{Vor}_F(S_\triangle)$ is the same as $\mathrm{Vor}_F(S) \cap \triangle$. Let σ be the intersection point of an edge $e \in E$ with a 2-face f of $\mathrm{Vor}_F(S)$ that minimizes $\omega(\sigma)$ over all such intersection points. Suppose σ lies in the tetrahedron $\triangle \in \Delta$ and f lies on a 2-face π of $\mathrm{Vor}_F(S)$ separating Voronoi cells of points p and q in S. Then, by construction, $\pi \cap \triangle$ is also a 2-face in $\mathrm{Vor}_F(S_\triangle) \cap \triangle$. Therefore σ is also an intersection point of e and a triangle $\tau \in T_\triangle$, which implies that $\mathcal{A}(\sigma)$ will be computed in Step 5.

As for the running time, $\mathrm{Vor}_N(S)$ can be computed in $O(n^2)$ time, and $\mathrm{Vor}_F(R)$ can be computed and triangulated in $O(r^2 \log^2 r)$ time. By the virtue of the Dobkin-Kirkpatrick algorithm, each edge in E intersects $O(\log r)$ tetrahedra of the triangulation of a cell of $\mathrm{Vor}_F(R)$, and they can be computed in $O(\log r)$ time. Hence,

$$\sum_\triangle m_\triangle = O(n^2 r \log^2 r)$$

and the sets $E_\triangle$ can be computed in $O(n^2 r \log^2 r)$ time. By the ε-net theory [21], $n_\triangle \leq n/r$ for all $\triangle$ with high probability, so

$$t_\triangle = O(n^2/r^2)$$

for all $\triangle$ with high probability. As shown in [2], $S_\triangle$, for all $\triangle \in \Delta$, can be computed in overall time $O(nr \log^2 r)$. Since $\mathrm{Vor}_F(S_\triangle)$ can be computed and triangulated in $O(n_\triangle^2)$ time, the total time spent in computing $T_\triangle$ over all tetrahedra is $\sum_\triangle O(n_\triangle^2)$, which is $O(n^2 \log^2 r)$ with high probability. Hence, the total time spent in Steps 3 and 4 is $O(n^2 r \log^2 r)$ with high probability. We will show below in Lemma 5.4 that Step 5 can be implemented in

$$\sum_{\triangle \in \Delta} O(t_\triangle m_\triangle^{9/10+\varepsilon} + m_\triangle \log^2 t_\triangle)$$

time, for any $\varepsilon > 0$. Hence, with high probability, the overall running time of the algorithm is

$$\sum_{\triangle \in \Delta} O\left(t_\triangle m_\triangle^{9/10+\varepsilon} + m_\triangle \log^2 \frac{n}{r}\right) + O(n^2 r \log^2 r) =$$
$$O\left(\frac{n^2}{r^2} \sum_{\triangle \in \Delta} m_\triangle^{9/10+\varepsilon} + n^2 r \log^2 r \log^2 \frac{n}{r}\right) =$$
$$O\left(n^2 \log^2 r \left(\frac{n^{9/5+2\varepsilon}}{r^{9/10+\varepsilon}} + r \log^2 \frac{n}{r}\right)\right).$$

Choosing $r = n^{18/19}$, we obtain that the running time of the algorithm is $O(n^{3-1/19+\varepsilon})$, for any $\varepsilon > 0$, with high probability. Putting everything together, we can conclude the following.

Theorem 5.1 *Given a set S of n points in $\mathbb{R}^3$, $\mathcal{A}^*(S)$ can be computed by a randomized algorithm whose running time is $O(n^{3-1/19+\varepsilon})$ with high probability.*

5.2 Computing $\mathcal{A}^*(E_\triangle, T_\triangle)$

Let A be a set of a segments in $\mathbb{R}^3$, each of which is an edge of $\mathrm{Vor}_N(S)$. Let B be a set of b triangles in

$\mathbb{R}^3$, each of which is a part of a 2-face of $\mathrm{Vor}_F(S)$. We want to compute $\mathcal{A}^*(A,B)$, a minimum-width shell centered at an intersection point of a segment in A and a triangle in B and containing S. Each segment $e \in A$ has three nearest neighbors $p_e, q_e, r_e \in S$, and each triangle $\tau \in B$ has two farthest neighbors $u_\tau, v_\tau \in S$. If e and τ intersect at a point σ, then $\omega(\sigma) = d(\sigma, u_\tau) - d(\sigma, p_e)$. Let $\omega^*(A,B)$ be the width of $\mathcal{A}^*(A,B)$. It is the minimum $\omega(\sigma)$, over all $\sigma = e \cap \tau$, $e \in A$, $\tau \in B$.

We first describe the algorithm for the restricted case in which every pair in $A \times B$ intersect. As in [1, 5, 6], we will describe the algorithm for the decision problem that, given a real number ω, determines whether $\omega > \omega^*(A,B)$, $\omega = \omega^*(A,B)$, or $\omega < \omega^*(A,B)$. Define $\delta_{e,\tau} = \omega^*(e \cap \tau) - \omega$. The goal is to determine whether there exists a pair e, τ so that $\delta_{e,\tau} \leq 0$, i.e.,

$$\min_{(e,\tau)\in A\times B} \delta_{e,\tau} \leq 0.$$

Since each pair in $A \times B$ intersect, we can replace each segment in A by the line supporting e and each triangle in τ by its supporting plane without introducing any new intersection points between entities in A and entities in B. Let us assume that no edge in A is parallel to the xy-plane. We map each line e in A to a point $\bar{e} = (e_1, \ldots, e_7) \in \mathbb{R}^7$ as follows: Let $p \in S$ be one of the three nearest neighbors defining e, and let q be the point on e nearest to p. The first 4 coordinates $(e_1, \ldots, e_4)$ parameterize the line e (e.g., as in [5]), e_5 specifies the z-coordinate of q, e_6 specifies the orientation of the vector $\vec{qp}$ in the plane normal to e (see Figure 9), and e_7 specifies the length of the vector $\vec{qp}$. For each triangle τ, we define a 7-variate function $f_\tau(x_1, \ldots, x_7)$ such that $f_\tau(\bar{e}) = \delta_{e,\tau}$. With an appropriate parametrization, these functions are semi-algebraic of "constant description complexity" (in the sense of [31]). Determining whether $\min_{e,\tau} \delta_{e,\tau} \leq 0$ is equivalent to determining whether $\min_{\tau\in B} f_\tau(\bar{e}) \leq 0$ for at least one line $e \in A$.

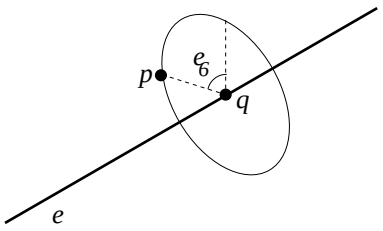

Figure 9: Parametrizing a line e by a point in $\mathbb{R}^7$.

By construction, f_τ is a monotonically decreasing function of x_7. Hence, the surface $f_\tau = 0$ is the graph of a 6-variate function $x_7 = \gamma_\tau(x_1, \ldots, x_6)$. For all $x_7 < \gamma_\tau(x_1, \ldots, x_6)$, $f(x_1, \ldots, x_7) > 0$, and for all $x_7 > \gamma_\tau(x_1, \ldots, x_6)$, $f(x_1, \ldots, x_7) < 0$. The problem now reduces to determining whether there exists a line $e \in A$ such that $\bar{e}_7 \geq \min_{\tau\in B} \gamma_\tau(\bar{e}_1, \ldots, \bar{e}_6)$, i.e., whether any point $\bar{e}$ lies above the lower envelope of $\Gamma = \{\gamma_\tau \mid \tau \in B\}$. The algorithm thus computes this lower envelope, preprocess the portion above it for efficient point location, and tests each point $\bar{e}$, for $e \in A$, whether it lies in this region. Chazelle *et al.* [10] showed that Γ can be processed in $O(b^{2d-3+\varepsilon}) = O(b^{11+\varepsilon})$ time and space so that a point-location query can be answered in $O(\log b)$ time. Agarwal *et al.*. [6] subsequently showed that if we only want to know whether a query point lies above the lower envelope of Γ, the preprocessing cost can be improved to $O(b^{2d-4+\varepsilon}) = O(b^{10+\varepsilon})$. Using this enhanced technique we obtain the following:

Lemma 5.2 *Γ can be preprocessed in time $O(b^{10+\varepsilon})$, for any $\varepsilon > 0$, into a data structure of roughly the same size, so that, for a query point $(x_1, \ldots, x_7)$, we can determine in $O(\log b)$ time whether $x_7 \geq \min_{\gamma\in\Gamma} \gamma(x_1, \ldots, x_6)$.*

Hence, we can determine in $O(b^{10+\varepsilon} + a \log b)$ time whether $\bar{e}_7 \geq \min_\gamma \gamma(\bar{e}_1, \ldots, \bar{e}_6)$ for any $e \in A$.

Finally, if not every pair of $A \times B$ intersect, we modify the above algorithm as follows. Using the line-triangle intersection data structure of [4], we can a construct a family $\mathcal{F}$ of *canonical pairs*

$$\mathcal{F} = \{(A_1, B_1), \ldots, (A_u, B_u)\}$$

with the following properties:

(i) $A_i \subseteq A$, $B_i \subseteq B$;

(ii) each edge in A_i intersects every triangle in B_i;

(iii) for every intersecting edge-triangle pair (e, τ), there is an i such that $e \in A_i$, $\tau \in B_i$; and

(iv) there exists a constant $r > 0$ (which can be chosen to be arbitrarily large) such that for any integer $0 \leq i \leq \lceil \log_r b \rceil$, the number of canonical pairs with $r^i < |B_j| \leq r^{i+1}$ is $O((b/r)^{4i+\varepsilon})$ and $\sum |A_j| = O(a)$, where the sum is taken over all canonical pairs for which $r^i < |B_j| \leq r^{i+1}$.

$\mathcal{F}$ can be constructed in $O(b^{4+\varepsilon} + a \log b)$ time. We apply the above algorithm for each pair (A_i, B_i) separately and return the thinnest annulus. The total

running time is

$$\sum_{j=1}^{u} O\left(|A_j| \log |B_j| + |B_j|^{10+\varepsilon}\right) = \sum_{i=0}^{\log_r b} O(a \log b + (b/r)^{4i+\varepsilon} r^{(i+1)(10+\varepsilon)}) = O(a \log^2 b + b^{10+\varepsilon'});$$

for any $\varepsilon' > \varepsilon > 0$. Using a standard batching technique, we can modify the algorithm so that its running time becomes $O(ba^{9/10+\varepsilon} + a \log^2 b)$, for any $\varepsilon > 0$. Hence, we obtain the following.

Lemma 5.3 *Let A be a set of a edges of $\mathrm{Vor}_N(S)$, B a set of b triangles lying in the 2-faces of $\mathrm{Vor}_F(S)$, and $\omega > 0$ a real number. We can determine in $O(ba^{9/10+\varepsilon} + a \log b)$ time the relative order of $\omega^*(A, B)$ and ω.*

Combining this procedure with the techniques in [5, 9], we can finally derive the following lemma, which, as argued above, completes the analysis of the running time of our algorithm.

Lemma 5.4 *Let A be a set of a edges of $\mathrm{Vor}_N(S)$ and B a set of b triangles lying in the 2-faces of $\mathrm{Vor}_F(S)$. We can compute $\mathcal{A}^*(A, B)$ in $O(ba^{9/10+\varepsilon} + a \log^2 b)$ time, for any $\varepsilon > 0$.*

References

[1] P. K. Agarwal, B. Aronov, and M. Sharir, Computing envelopes in four dimensions with applications, *SIAM J. Comput.*, 26 (1997), 1714–1732.

[2] P. K. Agarwal, H. Edelsbrunner, O. Schwarzkopf, and E. Welzl, Euclidean minimum spanning trees and bichromatic closest pairs, *Discrete Comput. Geom.*, 6 (1991), 407–422.

[3] P. K. Agarwal and J. Erickson, Geometric range searching and its relatives, in: *Advances in Discrete and Computational Geometry* (J. E. G. B. Chazelle and R. Pollack, eds.), AMS Press, Providence, RI, 1998, pp. 1–56.

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

[5] P. K. Agarwal and M. Sharir, Efficient randomized algorithms for some geometric optimization problems, *Discrete Comput. Geom.*, 16 (1996), 317–337.

[6] P. K. Agarwal, M. Sharir, and S. Toledo, Applications of parametric searching in geometric optimization, *J. Algorithms*, 17 (1994), 292–318.

[7] A. Aggarwal, L. J. Guibas, J. Saxe, and P. W. Shor, A linear-time algorithm for computing the Voronoi diagram of a convex polygon, *Discrete Comput. Geom.*, 4 (1989), 591–604.

[8] F. Aurenhammer. Voronoi diagrams: A survey of a fundamental geometric data structure. *ACM Comput. Surv.*, 23:345–405, 1991.

[9] T. M. Chan, Geometric applications of a randomized optimization technique, *Proc. 14th Annu. ACM Sympos. Comput. Geom.*, 1998, pp. 269–278.

[10] B. Chazelle, H. Edelsbrunner, L. J. Guibas, and M. Sharir, A singly-exponential stratification scheme for real semi-algebraic varieties and its applications, *Theoret. Comput. Sci.*, 84 (1991), 77–105.

[11] M. de Berg, J. Bose, D. Bremner, S. Ramaswami, and G. Wilfong, Computing constrained minimum-width annuli of point sets, *Proc. 5th Workshop Algorithms Data Struct.*, *Lecture Notes Comput. Sci.*, Vol. 1272, Springer-Verlag, 1997, pp. 3–16.

[12] D. P. Dobkin and D. G. Kirkpatrick, A linear algorithm for determining the separation of convex polyhedra, *J. Algorithms*, 6 (1985), 381–392.

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

[14] M. Dyer and N. Megiddo, Linear programming in low dimensions, in: *Handbook of Discrete and Computational Geometry* (J. E. Goodman and J. O'Rourke, eds.), CRC Press LLC, Boca Raton, FL, 1997, pp. 699–710.

[15] H. Ebara, N. Fukuyama, H. Nakano, and Y. Nakanishi, Roundness algorithms using the Voronoi diagrams, *Abstracts 1st Canad. Conf. Comput. Geom.*, 1989, p. 41.

[16] H. Edelsbrunner, L. J. Guibas, and J. Stolfi, Optimal point location in a monotone subdivision, *SIAM J. Comput.*, 15 (1986), 317–340.

[17] O. Egecioglu and B. Kalantari, Approximating the diameter of a set of points in the Euclidean space, *Inform. Process. Lett.*, 32 (1989), 205–211.

[18] L. W. Foster, *GEO-METRICS II: The Application of Geometric Tolerancing Techniques*, Addison-Wesley, Reading, MA, 1982.

[19] J. García-Lopez and P. Ramos, Fitting a set of points by a circle, *Proc. 13th Annu. ACM Sympos. Comput. Geom.*, 1997, pp. 139–146.

[20] S. Har-Peled, Constructing approximate shortest path maps in three dimensions, *Proc. 14th Annu. ACM Sympos. Comput. Geom.*, 1998, pp. 383–391.

[21] D. Haussler and E. Welzl, Epsilon-nets and simplex range queries, *Discrete Comput. Geom.*, 2 (1987), 127–151.

[22] R. Kumar, and D. Sivakumar, Roundness estimation via random sampling, *Proc. 10th ACM-SIAM Sympos. Discrete Algorithms*, 1999, pp. 603–612.

[23] V. B. Le and D. T. Lee, Out-of-roundness problem revisited, *IEEE Trans. Pattern Anal. Mach. Intell.*, PAMI-13 (1991), 217–223.

[24] J. Matoušek, M. Sharir, and E. Welzl, A subexponential bound for linear programming, *Algorithmica*, 16 (1996), 498–516.

[25] K. Mehlhorn, T. Shermer, and C. Yap, A complete roundness classification procedure, *Proc. 13th Annu. ACM Sympos. Comput. Geom.*, 1997, pp. 129–138.

[26] J. S. B. Mitchell, D. M. Mount, and C. H. Papadimitriou, The discrete geodesic problem, *SIAM J. Comput.*, 16 (1987), 647–668.

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

[28] T. J. Rivlin, Approximating by circles, *Computing*, 21 (1979), 93–104.

[29] U. Roy, C. R. Liu, and T. C. Woo, Review of dimensioning and tolerancing: representation and processing, *Comput. Aided Design*, 23 (1991), 466–483.

[30] U. Roy and X. Zhang, Establishment of a pair of concentric circles with the minimum radial separation for assessing roundness error, *Comput. Aided Design*, 24 (1992), 161–168.

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

[32] T. C. Shermer and C. K. Yap, Probing for near centers and relative roundness, *Proc. ASME Workshop on Tolerancing and Metrology*, 1995.

[33] M. Smid and R. Janardan, On the width and roundness of a set of points in the plane, *Proc. 7th Canad. Conf. Comput. Geom.*, 1995, pp. 193–198.

[34] K. Swanson, D. T. Lee, and V. L. Wu, An optimal algorithm for roundness determination on convex polygons, *Comput. Geom. Theory Appl.*, 5 (1995), 225–235.

[35] C. K. Yap and E.-C. Chang, Issues in the metrology of geometric tolerancing, *Algorithms for Robotic Motion and Manipulation* (J.-P. Laumond and M. Overmars, ed.), A.K. Peters, Wellesley, MA, 1997, pp. 393–400.