Diamonds are Forever in the Blockchain: Geometric Polyhedral Point-Set Pattern Matching

Gill Barequet[*] Shion Fukuzawa[†] Michael T. Goodrich[†]
David M. Mount[‡] Martha C. Osegueda[†] Evrim Ozel[†]

Abstract

Motivated by blockchain technology for supply-chain tracing of ethically sourced diamonds, we study geometric polyhedral point-set pattern matching as minimum-width polyhedral annulus problems under translations and rotations. We provide two $(1+\varepsilon)$-approximation schemes under translations with $O(\varepsilon^{-d}n)$-time for d dimensions and $O(n \log \varepsilon^{-1} + \varepsilon^{-2})$-time for two dimensions, and we give an $O(\varepsilon^{1-2d}n)$-time algorithm when also allowing for rotations.

1 Introduction

A notable recent computational geometry application is for tracking supply chains for natural diamonds, for which the industry and customers are strongly motivated to prefer ethically-sourced provenance (***e.g.***, to avoid so-called "blood diamonds"). For example, the ***Tracr*** system employs a blockchain for tracing the supply chain for a diamond from its being mined as a rough diamond to a customer purchasing a polished diamond [20]. (See Figure 1.)

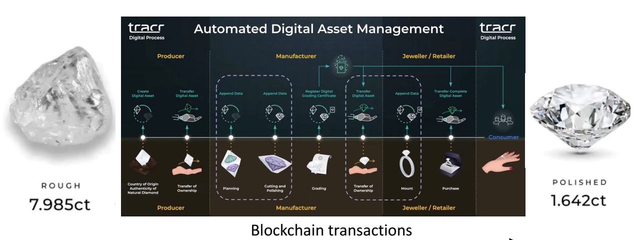

Figure 1: Blockchain transactions in a diamond supply chain, providing provenance, traceability, and authenticity of an ethically-sourced diamond.

Essential steps in the Tracr blockchain supply-chain process require methods to match point sets against geometric shapes, ***e.g.***, to guarantee that a diamond has not been replaced with one of questionable provenance [20]. Currently, the Tracr system uses standard machine-learning techniques to perform the shape matching steps; however, we believe better accuracy can be achieved by using computational geometry approaches. In particular, motivated by the Tracr application, we are interested in this paper in efficient methods for matching point sets against geometric shapes, such as polyhedra. Formalizing this problem, we study the problem of finding the best translation and/or rotation of the boundary of a convex polytope, P (***e.g.***, defining a polished diamond shape), to match a set of n points in a d-dimensional ($d \geq 3$) space, where the point set is a "good" sample of the boundary of a polytope that is purported to be P. Since there may be small inaccuracies in the sampling process, our aim is to compute a ***minimum width*** polyhedral annulus determined by P that contains the sampled points. In the interest of optimizing running time, rather than seeking an exact solution, we seek an approximate solution that deviates from the real solution by a predefined quantity $\varepsilon > 0$.

Related Work. We are not familiar with any previous work on the problems we study in this paper. Nevertheless, there is considerable prior work on the general area of matching a geometric shape to a set of points, especially in the plane. For example, Barequet, Bose, Dickerson, and Goodrich [12] give solutions to several constrained polygon annulus placement problems for offset and scaled polygons including an algorithm for finding the translation for the minimum offset of an m-vertex polygon that contains a set of n points in $O(n \log^2 n + m)$ time. Barequet, Dickerson, and Scharf [13] study the problem of covering a maximum number of n points with an m-vertex polygon (not just its boundary) under translations, rotations, and/or scaling, giving, ***e.g.***, an algorithm running in time $O(n^3m^4 \log(nm))$ for the general problem. There has also been work on finding a minimum-width annulus for rectangles and squares, ***e.g.***, see [9, 11, 17, 18].

Chan [14] presents a $(1+\varepsilon)$-approximation method that finds a minimum-width spherical annulus of n points in d dimensions in $O(n \log(1/\varepsilon) + \varepsilon^{O(1)})$ time, and Agarwal, Har-Peled, and Varadarajan [1] improve

[*]Dept. of Computer Science, Technion—Israel Inst. of Technology, barequet@cs.technion.ac.il

[†]Dept. of Computer Science, University of California, Irvine, {fukuzaws,goodrich,mosegued,eozel}@uci.edu

[‡]Dept. of Computer Science, University of Maryland, College Park, mount@umd.edu

this to $O(n + 1/\varepsilon^{O(d^2)})$ time via coresets [2, 3, 19, 21]. Arya, da Fonseca, and Mount [6] show how to find an ε-approximation of the width of n points in $O(n \log(1/\varepsilon)+ 1/\varepsilon^{(d-1)/2+\alpha})$ time, for a constant $\alpha > 0$. Bae [10] shows how to find a min-width d-dimensional hypercubic shell in $O(n^{\lfloor d/2 \rfloor} \log^{d-1} n)$ expected time.

Our Results. Given a set of n points in $\mathbf{R}^d$, we provide an $O(\varepsilon^{-d} n)$-time $(1+\varepsilon)$-approximate polytope-matching algorithm under translations, for $d \geq 3$, and $O(n \log \varepsilon^{-1} + \varepsilon^{-2})$ time for $d = 2$, and we provide an $O(\varepsilon^{1-2d} n)$-time algorithm when also allowing for rotations, where the complexity of the polytope is constant.

2 Preliminaries

Following previous convention [4,5,7,8,16], we say that a point set S is a ***δ-uniform sample*** of a surface $\Sigma \subset \mathbb{R}^d$ if for every point $p \in \Sigma$, there exists a point $q \in S$ such that $d(p, q) \leq \delta$. Let $C \subset \mathbb{R}^d$ be a polyhedron containing the origin. Given C, and $x \in \mathbb{R}^d$, define $x+C = \{x+y : y \in C\}$ (the translation of C by x), and for $r \in \mathbb{R}$, define $rC = \{ry : y \in C\}$. A ***placement*** of C is a pair (x, r), where $x \in \mathbb{R}^d$ and $r \in \mathbb{R}^{\geq 0}$, representing the translated and scaled copy $x + rC$. We refer to x and r as the ***center*** and ***radius*** of the placement, respectively. Two placements are ***concentric*** if they share the same center. Let C be any closed convex body in $\mathbb{R}^d$ containing the origin in its interior. The convex distance function induced by C is the function $d_C : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^{\geq 0}$, where $d_C(p, q) = \min\{r : r \geq 0 \text{ and } q \in p + rC\}$. Thus, the convex distance between p and q is determined by the minimum radius placement of C centered at p that contains q (see Figure 2). When C is centrally symmetric, this defines a metric, but for general C, the function d_C may not be symmetric. We call the original shape C the ***unit ball*** U_C under the distance function d_C. Note that $d_C(a, c) = d_C(a, b) + d_C(b, c)$ when a, b and c are colinear and appear in that order. Define an ***annulus*** for C to be the set-theoretic difference of two concentric placements $(p + RC) \setminus (p + rC)$, for $0 \leq r \leq R$. The ***width*** of the annulus is $R - r$. Given a δ-uniform sample of points, S, there are three placements of C we are interested in:

- **Minimum enclosing ball (MinBall)**: A placement of C of the smallest radius that contains all of the points in S.
- **Maximum enclosed ball (MaxBall)**: A placement of C of the largest radius, centered within the convex hull of S, that contains no points in S.
- **Minimum width annulus (MWA)**: Given a set $S \subset \mathbb{R}^d$ and a convex body C, the minimum width annulus of S is the annulus for C of the smallest width that contains S.

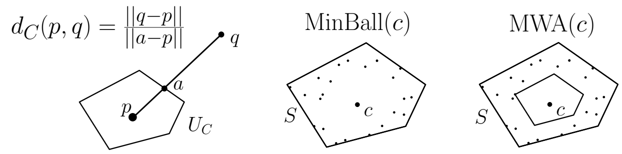

Figure 2: **Left:** a visual representation of a polyhedral distance function and the distance between two points. **Center:** The MinBall under d_C containing all points in S, centered at c. **Right:** The MWA of S with all points within MinBall(c)\MaxBall(c).

Note that, following the definition of the MaxBall, we require that the center of the MWA must also lie within the convex hull of S. For each of the above placements, we also refer to parameterized versions, for example MinBall(p), MaxBall(p), or MWA(p). These respectively refer to the minimum enclosing ball, maximum enclosed ball, or minimum width annulus centered at the point p. Further, we use $|\text{MinBall}(p)|$ and $|\text{MaxBall}(p)|$ to denote the radius of MinBall(p) and MaxBall(p), respectively, and we use $|\text{MWA}(p)|$ to denote the width of MWA(p).

Further, the ratio, F, of the MinBall over the MaxBall of $S \subset \mathbb{R}^d$ under distance function d_C defines the ***fatness*** of S under d_C. Also, we define the ***concentric fatness*** as the ratio of the MinBall and MaxBalls centered at the MWA, and we define the ***slimness*** to be $f^{-1} = 1 - F^{-1}$, which for concentric fatness corresponds to the ratio of the MWA over the MinBall.

Remark 1 *In order for a δ-uniform sample to represent the surface, Σ, with enough accuracy for a meaningful* MWA*, the sample must contain at least one point between corresponding facets of the* MWA*. Where corresponding facets refer to facets of the Min and* MaxBall *representing the same facet of U_C. Therefore, in the remainder of the paper, we have a δ-uniform sample and that δ is small enough to guarantee this condition for even the smallest facets.*

In practice, it would be easy to determine a small enough δ before sampling Σ, since only sufficiently slim surfaces would benefit from finding the MWA and very fat surfaces would yield increasingly noisy MaxBall. Thus, setting δ to the smallest facet of the MinBall and scaling down by an arbitrary constant larger than the maximum expected fatness, such as 100.

Also, note that, for a center point c, the problem of finding MWA(c) has a unique solution, because a unit ball, U_C, is convex and once the placement's radius grows past $d_C(c, p)$, it must contain $p \in S$. Thus, the inner and outer radii are defined as $\min_{p \in S} d_C(c, p)$ and $\max_{p \in S} d_C(c, p)$, respectively. Further, let us assume that the reference polytope defining our polyhedral distance function has m facets, where m is a fixed constant, since the sample size is expected to be much larger than m. Thus, d_C can be calculated in $O(m)$ time; hence, MWA(c) can be found in $O(mn)$ time, which is $O(n)$ under our assumption.

3 Approximating the Min. Width Annulus

Let us first describe how to find a constant factor approximation of MWA under translations. Note that, by assumption, the center c of our approximation lies within the convex hull of S. Let us denote the center, outer radius, inner radius, and width of the optimal MWA as c_{opt}, R_{opt}, r_{opt}, and w_{opt}.

Lemma 1 *The center of the* MWA*, c_{opt}, is under w_{opt} distance away from the center of the* MinBall*, c. That is, $d_C(c, c_{opt}) \leq w_{opt}$.*

Proof. Recall our assumption from Remark 1. Since some points must always exist in each facet, the MinBall cannot shrink past any facets of MaxBall(c). Suppose for contradiction that $d_C(c, c_{opt}) > w_{opt}$. Let s be the point where a ray projected from c through c_{opt} intersects the boundary of MaxBall(c_{opt}) and let R denote the radius of the MinBall.

$$\begin{aligned} R > d_C(c,s) &= d_C(c,c_{opt}) + d_C(c_{opt},s) && \text{by colinearity} \\ &> w_{opt} + d_C(c_{opt},s) && \text{by assumption} \\ &> w_{opt} + r_{opt} && \text{by MaxBall}(c_{opt}). \end{aligned}$$

Thus, since $w_{opt} + r_{opt} = R_{opt}$, we find $R > R_{opt}$, which is a contradiction since R must be the smallest radius of the MinBall across all possible centers. Therefore, we have that $d_C(c, c_{opt})$ cannot be larger than w_{opt}. □

Lemma 1 helps us constrain the region within which c must be contained. Let us now reason about how these different center points would serve as approximations.

Lemma 2 *Suppose c is an arbitrary center-point in our search region, and the two directed distances between c and c_{opt} are at most t,* ***i.e.****, $t \geq \max\{d_C(c, c_{opt}), d_C(c_{opt}, c)\}$. Then, we have that $|\mathrm{MWA}(c)| \leq w_{opt} + 2t$.*

Proof. Let p be the point where the ray from c through c_{opt} intersects the boundary of MinBall(c_{opt}). In the worst case, MinBall(c) would need to contain p; hence,

$$d_C(c,p) = d_C(c,c_{opt}) + d_C(c_{opt},p) \leq t + d_C(c_{opt},p)$$
$$R \leq R_{opt} + t.$$

Conversely, let q be the intersection point where the ray projected from c_{opt} through c intersects the boundary of MaxBall(c_{opt}), in which case

$$d_C(c,q) = d_C(c_{opt},q) - d_C(c_{opt},c) \geq d_C(c_{opt},q) - t$$
$$r \geq r_{opt} - t.$$

Putting together these worst cases for MinBall(c) and MaxBall(c) implies that $|\mathrm{MWA}(c)| \leq w_{opt} + 2t$. □

For simplicity, let us consider two points a, b to be ***t-close*** (under C) whenever $t \geq \max\{d_C(a, b), d_C(b, a)\}$.

Lemma 3 *If c is the center of* MinBall*, then* MWA(c) *is a constant factor approximation,* ***i.e.****, $|\mathrm{MWA}(c)| \leq b|\mathrm{MWA}|$, for some constant $b \geq 1$, under translations.*

Proof. From Lemma 1, we have that $d_C(c, c_{opt}) \leq w_{opt}$. If c and c_{opt} are w_{opt}-close, then we can directly apply the second part of Lemma 2 to find $r \geq r_{opt} - w_{opt}$ and $R \leq R_{opt}$, such that $|\mathrm{MWA}(c)| \leq R_{opt} - (r_{opt} - w_{opt})$, thus proving that this is a 2-approximation. If d_C is a metric, then $d_C(c_{opt}, c) = d_C(c, c_{opt})$ and this must always be the case. However, if $d_C(c_{opt}, c) > w_{opt}$, then we must use the Euclidean distance to find $d_C(c_{opt}, c)$. Let vector $u := c - c_{opt}$, and let us define unit vectors with respect to d_C and $d_{\overline{C}}$, such that

$$\hat{u_C} = \frac{u}{d_C(c_{opt},c)} \quad , \quad \hat{u_{\overline{C}}} = \frac{\overline{u}}{d_C(c,c_{opt})}$$
$$||\hat{u_C}||d_C(c_{opt},c) = ||u|| = ||\hat{u_{\overline{C}}}||d_C(c,c_{opt})$$
$$d_C(c_{opt},c) \leq \frac{||\hat{u_{\overline{C}}}||}{||\hat{u_C}||} w_{opt} \qquad \text{from Lemma 1.}$$

Under any convex distance function, $\frac{||\hat{u_{\overline{C}}}||}{||\hat{u_C}||}$ is bounded from above by $A = \max_{v \in \mathbb{R}^d} \frac{||\hat{v_{\overline{C}}}||}{||\hat{v_C}||}$, which corresponds to finding the direction, v, of the largest asymmetry in U_C. Thus, by Lemma 2, $|\mathrm{MWA}(c)| \leq (A + 1)w_{opt}$. Under our (fixed) polyhedral distance function, A is constant; hence, MWA(c) is a constant-factor approximation. □

$(1 + \varepsilon)$-approximation. Let us now describe how to compute a $(1 + \varepsilon)$-approximation of MWA.

Lemma 4 *Suppose c_{opt} and c are $(\varepsilon w/(2b))$-close, where $w = |\mathrm{MWA}(c_M)|$, c_M is the center of* MinBall*, and b is the constant from Lemma 3. Then,* MWA(c) *is a $(1 + \varepsilon)$-approximation of* MWA *under translations.*

Proof. To be a $(1 + \varepsilon)$-approximation of MWA, the width of our approximated annulus must be at most $(1 + \varepsilon)$ times the width of the optimal one. Assuming c and c_{opt} are t-close, and using Lemma 2, we require that $w_{opt} + 2t \leq (1 + \varepsilon)w_{opt}$, ***i.e.***, $t \leq \varepsilon w_{opt}/2$. Let us then choose $t \leq \varepsilon w/(2b)$, knowing that $w \leq b w_{opt}$ from Lemma 3, which is sufficient for achieving a $(1 + \varepsilon)$-approximation. □

Knowing how close our approximation's center must be, we can now put together a $(1 + \varepsilon)$-approximation algorithm to find a center satisfying this condition.

Theorem 5 *One can achieve a $(1 + \varepsilon)$-approximation of the* MWA *under translations in $O(\varepsilon^{-d} n)$ time.*

Proof. The MinBall can be computed in $O(n)$ time [15]. By Lemma 1, we have that $d_C(c, c_{opt}) \leq w_{opt}$, where c is the MinBall center. This implies that c_{opt} must lay within the placement $c + w_{opt}C$ or more generously in P, defined as $c + wC$. Furthermore, from Lemma 4, we know that being $(\varepsilon w/(2b))$-close to c_{opt} suffices for an $(1+\varepsilon)$-approximation. Therefore, overlaying a grid G that covers P, such that any point in $p \in P$ is $(\varepsilon w/(2b))$-close to a gridpoint, guarantees the existence of a point $g \in G$ for which $\mathrm{MWA}(g)$ is a $(1+\varepsilon)$-approximation.

Since P and $(\varepsilon w/(2b))$-closeness are both defined under d_C, we translate this to a cubic grid for simplicity. Let Q be the smallest cube enclosing P and q be the largest cube enclosed by $(\varepsilon w/(2b))C$. Let us now define a grid Q with cubes of size q, Grid Q has points Fb/ε apart and $F^d b^d \varepsilon^{-d}$ gridpoints in total, where F corresponds to the fatness of C under the distance function defined by the unit cube.

Let q define the distance function d_q where the unit ball U_q is $(\varepsilon w/(2b))$ times smaller than q. The grid G guarantees that for every point p, there exists a gridpoint $g \in G$ such that $d_q(p, g) \leq \varepsilon w/(2b)$. Since the unit cube is contained within the unit polyhedron, we have that $d_C(a, b) \leq d_q(a, b)\ \forall a, b$; and since d_q defines a metric, p must also be $(\varepsilon w/(2b))$-close under d_C. Finding the gridpoint providing the $(1+\varepsilon)$-approximation takes $O(F^d b^d \varepsilon^{-d} n)$ time,[1] which, under a fixed d_C, is $O(\varepsilon^{-d} n)$ time. □

Faster grid-search in two dimensions. The algorithm of Theorem 5 recalculates the MWA at every gridpoint. However, small movements along the grid should not affect the MWA much. We use this insight to speed up MWA recalculations for two dimensions.

Let us first define the ***contributing edge*** of a sample point, $p \in S$, as the edge of $C + g$ intersected by the ray emanating from a gridpoint, g, towards p. Under this center-point, p will only directly affect the placement of the contributing edge. Observe that given vectors $\overrightarrow{v} \in C$, defined as the vectors directed from the center towards each vertex, the planar subdivision, created by rays for each $\overrightarrow{v}$ originating from g, separates points by their contributing edge. For any two gridpoints, g_1 and g_2, and rays projected from them parallel to $\overrightarrow{v}$, any points within these two rays will contribute to different edges under g_1 and g_2. We denote this region as the ***vertex slab*** of vertex v, and the regions outside of this as ***edge slabs***. Points within an edge slab will contribute to the same edge under both gridpoints, maintaining the constraints this imposes on the MWA,

[1] For metrics, MinBall provides a 2-approximation, thus b=2. For non-metrics, we can remove this constant by first using this algorithm with ε=1 in order to find a 2-approximation in linear-time, and using this approximation for gridding in the main step.

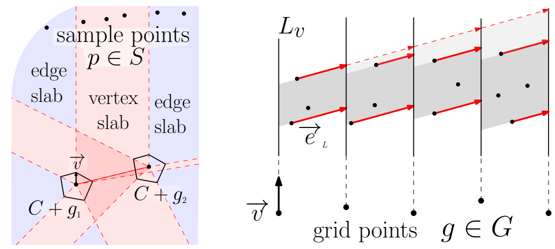

Figure 3: **Left:** Planar subdivision defining vertex slabs (red) and edge slabs (blue) for two candidate center-points, and showing membership of some sample points. **Right:** L_v and the extreme-most points under $\overrightarrow{e}_L$ for each region (solid) and for all regions to its left (dashed).

can therefore be achieved with the two extreme-most points per edge slab. If we consider vertex slabs for all $g \in G$, we must be able to quickly calculate the strictest constraints imposed by points in a subset of vertex slabs. An example of the planar subdivision for two points is shown in Figure 3 (left).

Given a grid G, we write $g_{i,j} \in G$ to be the gridpoint at index (i, j). Consider the set of all grid lines L_v defined by rays parallel to $\overrightarrow{v}$ starting at each gridpoint. To quickly recalculate changes to edges incident on v as we traverse through gridpoints, we need to quickly identify which slab a sample-point p belongs to, given a planar subdivision defined by L_v.

Lemma 6 *For a specific vector $\overrightarrow{v}$ and an $m \times m$ grid, we can identify which slab a sample point, p, belongs to in $O(\log m)$ time with $O(m^2)$-time preprocessing.*

Proof. Consider the orthogonal projection of grid lines in L_v onto a line $\overrightarrow{v_\perp}$ perpendicular to $\overrightarrow{v}$, the order in which these lines appear in $\overrightarrow{v_\perp}$ defines the possible slabs that p could belong to. We can project a given grid line $l \in L_v$ onto $\overrightarrow{v_\perp}$ in constant time. After sorting these grid lines, we can perform a binary search through the m^2 points in $O(\log m)$ time to identify the slab which p would belong to.

Using general sorting algorithms, we could sort the grid lines in $O(m^2 \log m)$ time. However, since these lines belong to a grid, we can exploit the uniformity to sort them in only $O(m^2)$ time. Consider the two basis vectors defining gridpoint positions $\hat{\imath} = g_{(1,0)} - g_{(0,0)}$ and $\hat{\jmath} = g_{(0,1)} - g_{(0,0)}$, and their sizes after orthogonal projection onto $\overrightarrow{v_\perp}$, $|\hat{\imath}_\perp|$, and $|\hat{\jmath}_\perp|$. Without loss of generality, assume that $|\hat{\imath}_\perp| \geq |\hat{\jmath}_\perp|$, in which case grid lines originating from adjacent gridpoints in the same row must be exactly $|\hat{\imath}_\perp|$ away. In addition, any region $|\hat{\imath}_\perp|$-wide, that does not start at a grid line, must contain at most a single point from each row. Furthermore, since points in the same row are always $|\hat{\imath}_\perp|$ away, they must appear in the same order in each region.

We can therefore initially split $\overrightarrow{v_\perp}$ into regions $|\hat{\imath}_\perp|$ wide. Sorting the grid lines $l \in L_v$ into their region can therefore be calculated in $O(m^2)$ time. Now we can sort the m points in the region containing points from every row in $O(m \log m)$ time. Since each region has the same order, we can place points in other regions by following the order found in our sorted region, thus taking $O(m^2)$ preprocessing time for sorting the points. □

Recall that points to the left of a given line $l \in L_v$ contribute to the edge to the left of v, ***i.e.***, all points belonging to slabs to the left of l. We can therefore isolate the points in these slabs causing the largest potential change in MWA.

Lemma 7 *For a vertex $v \in C$ and grid line $l \in L_v$ through gridpoint g, let l_L and l_R refer to the slabs on the subdivision imposed by L_v immediately to the left and right of l, respectively. Assuming l_L maintains the points to the left of l imposing the strictest constraints on* $\mathrm{MWA}(g)$*, and l_R to the right, one can calculate* $\mathrm{MWA}(g)$ *in $O(1)$ time.*

Proof. Finding $\min_{p \in S} d_C(g,p)$ and $\max_{p \in S} d_C(g,p)$ can now be achieved by optimizing only over the set of points in $\{l_L \cup l_R \ \forall v \in C\}$ and all points in edge slabs. This set would contain two points per vertex and two points per edge, yielding a constant number of points. Thus, $\mathrm{MWA}(g)$ can be found in constant time. □

Theorem 8 *A $(1+\varepsilon)$-approximation of the* MWA *in two dimensions can be found in $O(n \log \varepsilon^{-1} + \varepsilon^{-2})$ time under translations.*

Proof. For each vertex, v, we use Lemma 6 to identify the slab for every sample point. For each slab, we maintain only the two extreme-most points for each of the edges incident on $\overrightarrow{v}$. Let $\overrightarrow{e}_L \in C$ denote the vector describing the edge incident on $\overrightarrow{v}$ from the left, and vice versa for $\overrightarrow{e}_R \in C$ incident from the right. For each slab, we maintain only points which when projected in the relevant direction, $\overrightarrow{e}$, cause the furthest and closest intersections with the boundary (shown for $\overrightarrow{e}_L$ in Figure 3 (right)). With a left-to-right pass, we update a slab's extreme-most points relative to $\overrightarrow{e}_L$ to maintain the extreme-most points for itself and slabs to its left. With a right-to-left pass, we do the same for $\overrightarrow{e}_R$ and maintain points in its slab and slabs to its right.

Thus, for each vertex, we create the slabs in $O(\varepsilon^{-2})$ time, identify a sample points slab in $O(\log \varepsilon^{-1})$ time per sample point, and keep the extreme-most points per slab in constant time per sample point. With $O(\varepsilon^{-2})$ time to update the slabs after processing all sample points, we can update the slabs such that they hold the extreme-most points across all slabs to their left or right (relative to $\overrightarrow{e}_L$ and $\overrightarrow{e}_R$, respectively).

For each edge slab, finding the extreme-most points is much simpler since finding $\min d_C(g,p)$ and $\max d_C(g,p)$ across all points in the edge slab will always be based on the contributing facet.

Thus, after finding the extreme-most points in both vertex slabs, we can calculate $\mathrm{MWA}(g)$ in constant time as described in Lemma 7. Taking $O(\varepsilon^{-2})$ time to find $\min_{g \in G} \mathrm{MWA}(g)$, which by Theorem 5 provides a $(1+\varepsilon)$-approximation of the minimum width annulus, completes the proof of the claimed time bound. □

4 Approximating MWA allowing rotations

In this section we consider rotations. As with Lemma 4, our goal is to find the maximum tolerable rotation sufficient for a $(1+\varepsilon)$-approximation. Observe that when centered about the global optimum, the solution found under both rotation and translation is at least as good as the solution found solely through rotation (***i.e.***, under a fixed center). We will therefore first prove necessary bounds for a $(1+\varepsilon)$-approximation under rotation only with the understanding that they remain when also allowing for translation.

Consider the polyhedral cone around $\overrightarrow{v}$ and define the ***bottleneck angle*** as the narrowest angle between a point on the surface of the polyhedral cone and $\overrightarrow{v}$. Let θ be the smallest bottleneck angle across all $\overrightarrow{v} \in C$. Let $\mathrm{MWA}_\alpha(c)$ describe the MWA centered at c, where C has been rotated by angle α. Let us also use similar notations for MinBall and MaxBall.

Lemma 9 *Rotating by α causes* $\mathrm{MinBall}_\alpha(c)$ *to grow by at most $\frac{\sin(\pi-\theta-\alpha)}{\sin\theta}$ (and the reciprocal for* $\mathrm{MaxBall}_\alpha(c)$*).*

Proof. In the worst case, $\mathrm{MinBall}(c)$ must be completely contained within $\mathrm{MinBall}_\alpha(c)$. Let us now consider the triangle formed between c, the vertex v of the original MinBall, v_0, and the rotated vertex v_α (shown in Figure 4). Since our calculations focus towards the same vertex, we will be working with Euclidean distances. The quantity $|v_0 - c|$ defines the radius r_1 of the original polyhedron, and $r_2 = |v_\alpha - c|$ the radius of the rotated one. With $\gamma = \pi - \theta - \alpha$ as the remaining angle in our triangle and using the sine rule, we find that

$$\frac{r_2}{r_1} = \frac{\sin\gamma}{\sin\theta} = \frac{\sin(\pi-\theta-\alpha)}{\sin\theta}.$$

Observe that θ is the angle maximizing this scale difference. This applies to rotating by α in any direction about $\overrightarrow{v}$, and since this direction needs not coincide with θ, the scaled polyhedron might not touch the original. For $\mathrm{MaxBall}_\alpha(c)$ to be contained within $\mathrm{MaxBall}(c)$, the same example holds after switching references to the scaled and original. In this case, θ minimizes r_1/r_2. □

Let us now determine the rotation from the optimal orientation that achieves a $(1+\varepsilon)$-approximation.

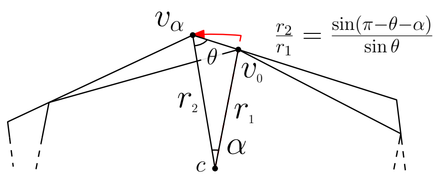

Figure 4: The scale increase necessary for a polyhedron rotated by α to contain the original.

Lemma 10 *Given a center c, we have that* $\mathrm{MWA}_\alpha(c)$ *is a* $(1+\varepsilon)$*-approximation when* α *is smaller than*

$$\arcsin\left(\frac{\sin\theta}{2f}\left(1{+}\varepsilon \pm \sqrt{(1{+}\varepsilon)^2 + 4f(f-1)}\right)\right) - \theta.$$

Proof. Define f as the ratio of the radius of $\mathrm{MinBall}(c_{opt})$ to w_{opt} (***i.e.***, $fw_{opt} = |\mathrm{MinBall}(c_{opt})|$). Note that f corresponds to the inverse of the concentric slimness of S under d_C over all rotations of C.

Using Lemma 9, we know that

$$|\mathrm{MWA}_\alpha(c)| \leq \frac{\sin\gamma}{\sin\theta}|\mathrm{MinBall}(c)| - \frac{\sin\theta}{\sin\gamma}|\mathrm{MaxBall}(c)|$$

$$\frac{\sin\gamma}{\sin\theta} f w_{opt} - \frac{\sin\theta}{\sin\gamma}(f{-}1)w_{opt} \leq (1{+}\varepsilon)w_{opt} \tag{1}$$

$$\frac{\sin\gamma}{\sin\theta} f - \frac{\sin\theta}{\sin\gamma}(f{-}1) \leq (1{+}\varepsilon) \tag{2}$$

To be a $(1+\varepsilon)$-approximation, we need $|\mathrm{MWA}_\alpha(c)| \leq (1{+}\varepsilon)w_{opt}$ imposing the right side of Inequality **??**, its left side follows by definition of f, and Inequality **??** by cancellation of w_{opt}. Since θ is constant, we can rearrange the above into a quadratic equation and solve for $\sin\gamma$.

$$\sin\gamma = \frac{\sin\theta}{2f}\left(1{+}\varepsilon \pm \sqrt{(1{+}\varepsilon)^2 + 4f(f{-}1)}\right). \tag{3}$$

However, arcsin will find $\gamma \leq \pi$, whereas we need the obtuse angle $\pi - \gamma$. Thus, proving this lemma's titular bound, and achieving a $(1+\varepsilon)$-approximation. □

Let us now establish a more generous lower-bound that will prove helpful when developing algorithms.

Lemma 11 *The angular deflection required for a* $(1+\varepsilon)$*-approximation is larger than* $\theta\varepsilon/(2f)$*.*

Proof. Observe that γ is of the form $\arcsin(k\sin\theta)$ and thus, in order for $\alpha = \gamma{-}\theta$ to be positive, we must have $\theta < \pi/2$ and $k > 1$. We will prove this is the case.

$$k = \frac{1{+}\varepsilon}{2f} + \sqrt{\left(\frac{1{+}\varepsilon}{2f}\right)^2 - \frac{1}{f} + 1} \tag{4}$$

$$\sqrt{\frac{1}{4f^2} - \frac{1}{f} + 1} = \left|1 - \frac{1}{2f}\right| \tag{5}$$

$$k > \frac{1{+}\varepsilon}{2f} + \left|1 - \frac{1}{2f}\right| = 1 + \frac{\varepsilon}{2f} \tag{6}$$

Equation (4) follows from Equation (3) after expanding. Equation (6) follows after using Equation (5) as a lower bound for the square root term in Equation (4) since $\varepsilon > 0$ and $f > 1$. This allows us to bound $\arcsin\left(\left(1 + \dfrac{\varepsilon}{2f}\right)\sin\theta\right)$ by using Taylor's series expansion to find $(1{+}k)\cdot\theta \leq \arcsin((1{+}k)\sin\theta)$, thus proving that the bound from Lemma 10 is greater than $\frac{\theta\varepsilon}{2f}$. □

Lemma 12 *For fixed rotation of C, assume we have an* $O(f(n))$*-time algorithm for the optimal minimum-width annulus under translation. We can find a* $(1+\varepsilon)$*-approximation of the* MWA *under rotations and translations in* $O((d-1)^{(d-1)/2}\varepsilon^{1-d}f(n))$ *time.*

Proof. A d-dimensional shape has a $(d{-}1)$-dimensional axis of rotation. Let us evenly divide the unit circle into k directions. Let us also define a collection of all possible direction combinations as a grid of directions. For each grid direction, rotate C by the defined direction and calculate the MWA in $O(f(n))$ time. The optimal orientation must lie between the $(d{-}1)$-dimensional cube formed by 2^{d-1} grid directions. Therefore, as long as the diagonal is smaller than $\frac{\theta\varepsilon}{f}$, there will always exist a grid direction within $\frac{\theta\varepsilon}{2f}$ of the optimal orientation and therefore achieving a $(1+\varepsilon)$-approximation by Lemma 11. Thus, we can achieve a $(1+\varepsilon)$-approximation in $O\left(f(n)\cdot\left(\frac{2\pi f\sqrt{d-1}}{\theta\varepsilon}\right)^{d-1}\right)$ time, where d, θ, and f are constant under a fixed distance function d_C. □

With a fixed center, Lemma 12 can be used to approximate MWA under rotations in $O(n\varepsilon^{d-1})$ time.

Theorem 13 *One can find a* $(1+\varepsilon)$*-approximation of* MWA *under rotations and translations in* $O(n\varepsilon^{1-2d})$ *time for d≥3, and* $O(n\varepsilon^{-1}\log\varepsilon^{-1} + \varepsilon^{-3})$ *time for d=2.*

Proof. Consider using an approximation algorithm (from Theorems 5 or 8) instead of an exact algorithm as in Lemma 12. Let us define $(1{+}\xi)$ as the approximation ratio necessary from the subroutines in order to achieve an overall approximation ratio of $(1+\varepsilon)$, such that $(1+\xi)^2 = 1+\varepsilon$. Since $\xi = \sqrt{1+\varepsilon} - 1$ and $0 < \varepsilon < 1$, then ξ must always be larger than $(\sqrt{2}-1)\varepsilon$, and thus, we can always pick a value for ξ which is $O(\varepsilon)$ and achieves the desired approximation. Thus, by following Lemma 12, we can find a $(1{+}(\sqrt{2}{-}1)\varepsilon)$-approximation using the $(1 + (\sqrt{2} - 1)\varepsilon)$-approximation algorithm from Theorem 5 to find a $(1+\varepsilon)$-approximation in $O(\varepsilon^{1-d}\cdot\varepsilon^{-d}n)$ time. Alternatively, for two dimensions, we can instead use the algorithm from Theorem 8 to find a $(1+\varepsilon)$-approximation in $O(n\varepsilon^{-1}\log\varepsilon^{-1} + \varepsilon^{-3})$ time. □

References

[1] P. K. Agarwal, S. Har-Peled, and K. R. Varadarajan. Approximating extent measures of points. *J. ACM*, 51(4):606–635, 2004.

[2] P. K. Agarwal, S. Har-Peled, K. R. Varadarajan, et al. Geometric approximation via coresets. *Combinatorial and Computational Geometry*, 52(1), 2005.

[3] P. K. Agarwal, S. Har-Peled, and H. Yu. Robust shape fitting via peeling and grating coresets. *Discrete & Computational Geometry*, 39(1):38–58, 2008.

[4] N. Amenta, D. Attali, and O. Devillers. Size of Delaunay triangulation for points distributed over lower-dimensional polyhedra: a tight bound. *Neural Information Processing Systems (NeurIPS): Topological Learning*, 2007.

[5] N. Amenta and M. Bern. Surface reconstruction by Voronoi filtering. *Discrete & Computational Geometry*, 22(4):481–504, 1999.

[6] S. Arya, G. D. da Fonseca, and D. M. Mount. Approximate convex intersection detection with applications to width and Minkowski sums. In *26th European Symposium on Algorithms (ESA)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[7] D. Attali and J.-D. Boissonnat. Complexity of the delaunay triangulation of points on polyhedral surfaces. *Discrete & Computational Geometry*, 30(3):437–452, 2003.

[8] D. Attali and J.-D. Boissonnat. A linear bound on the complexity of the Delaunay triangulation of points on polyhedral surfaces. *Discrete & Computational Geometry*, 31(3):369–384, 2004.

[9] S. W. Bae. Computing a minimum-width square annulus in arbitrary orientation. *Theoretical Computer Science*, 718:2–13, 2018.

[10] S. W. Bae. Computing a minimum-width cubic and hypercubic shell. *Operations Research Letters*, 47(5):398–405, 2019.

[11] S. W. Bae. On the minimum-area rectangular and square annulus problem. *Computational Geometry*, 92:101697, 2021.

[12] G. Barequet, P. Bose, M. T. Dickerson, and M. T. Goodrich. Optimizing a constrained convex polygonal annulus. *J. Discrete Algorithms*, 3(1):1–26, 2005.

[13] G. Barequet, M. T. Dickerson, and Y. Scharf. Covering points with a polygon. *Computational Geometry*, 39(3):143–162, 2008.

[14] T. M. Chan. Approximating the diameter, width, smallest enclosing cylinder, and minimum-width annulus. In *16th Symposium on Computational Geometry (SoCG)*, pages 300–309, 2000.

[15] S. Das, A. Nandy, and S. Sarvottamananda. Linear time algorithm for 1-Center in $\mathfrak{R}^d$ under convex polyhedral distance function. In D. Zhu and S. Bereg, editors, *Frontiers in Algorithmics*, volume 9711, pages 41–52. Springer, 2016.

[16] J. Erickson. Nice point sets can have nasty Delaunay triangulations. In *17th Symposium on Computational Geometry (SoCG)*, pages 96–105, 2001.

[17] O. N. Gluchshenko, H. W. Hamacher, and A. Tamir. An optimal $O(n \log n)$ algorithm for finding an enclosing planar rectilinear annulus of minimum width. *Operations Research Letters*, 37(3):168–170, 2009.

[18] J. Mukherjee, P. R. Sinha Mahapatra, A. Karmakar, and S. Das. Minimum-width rectangular annulus. *Theoretical Computer Science*, 508:74–80, 2013.

[19] J. M. Phillips. Coresets and sketches. In *Handbook of Discrete and Computational Geometry*, pages 1269–1288. Chapman and Hall/CRC, 2017.

[20] U. Thakker, R. Patel, S. Tanwar, N. Kumar, and H. Song. Blockchain for diamond industry: Opportunities and challenges. *IEEE Internet of Things Journal*, pages 1–1, 2020.

[21] H. Yu, P. K. Agarwal, R. Poreddy, and K. R. Varadarajan. Practical methods for shape fitting and kinetic data structures using coresets. *Algorithmica*, 52(3):378–402, 2008.