Bipartite Diameter and Other Measures Under Translation

Boris Aronov
Department of Computer Science and Engineering, Tandon School of Engineering,
New York University, Brooklyn, NY 11201, USA
boris.aronov@nyu.edu

Omrit Filtser
Department of Computer Science, Ben-Gurion University of the Negev,
Beer-Sheva 84105, Israel
omritna@post.bgu.ac.il

Matthew J. Katz
Department of Computer Science, Ben-Gurion University of the Negev,
Beer-Sheva 84105, Israel
matya@cs.bgu.ac.il

Khadijeh Sheikhan
Department of Computer Science and Engineering, Tandon School of Engineering,
New York University, Brooklyn, NY 11201, USA
khadijeh.sheikhan@gmail.com

Abstract

Let A and B be two sets of points in $\mathbb{R}^d$, where $|A| = |B| = n$ and the distance between them is defined by some bipartite measure $dist(A, B)$. We study several problems in which the goal is to translate the set B, so that $dist(A, B)$ is minimized. The main measures that we consider are (i) the *diameter* in two and three dimensions, that is $\mathrm{diam}(A, B) = \max\{d(a, b) \mid a \in A, b \in B\}$, where $d(a, b)$ is the Euclidean distance between a and b, (ii) the *uniformity* in the plane, that is $\mathrm{uni}(A, B) = \mathrm{diam}(A, B) - d(A, B)$, where $d(A, B) = \min\{d(a, b) \mid a \in A, b \in B\}$, and (iii) the *union width* in two and three dimensions, that is $\mathrm{union\_width}(A, B) = \mathrm{width}(A \cup B)$. For each of these measures we present efficient algorithms for finding a translation of B that minimizes the distance: For diameter we present near-linear-time algorithms in $\mathbb{R}^2$ and $\mathbb{R}^3$, for uniformity we describe a roughly $O(n^{9/4})$-time algorithm, and for union width we offer a near-linear-time algorithm in $\mathbb{R}^2$ and a quadratic-time one in $\mathbb{R}^3$.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Translation-invariant similarity measures, Geometric optimization, Minimum-width annulus

Digital Object Identifier 10.4230/LIPIcs.STACS.2019.8

Funding *Boris Aronov*: Work on this paper by Boris Aronov was supported by NSF Grants CCF-11-17336, CCF-12-18791, and CCF-15-40656, and by grant 2014/170 from the US-Israel Binational Science Foundation.
Omrit Filtser: Work on this paper by Omrit Filtser was supported by the Israel Ministry of Science, Technology & Space, and by the Lynn and William Frankel Center.
Matthew J. Katz: Work on this paper by Matthew Katz was supported by grant 1884/16 from the Israel Science Foundation, and by grant 2014/170 from the US-Israel Binational Science Foundation.
Khadijeh Sheikhan: Work on this paper by Khadijeh Sheikhan was supported by NSF Grant CCF-12-18791.

Acknowledgements We would like to thank Pankaj K. Agarwal, Mark de Berg, and Timothy Chan for their assistance in the preparation of this manuscript.

36th International Symposium on Theoretical Aspects of Computer Science (STACS 2019).
Editors: Rolf Niedermeier and Christophe Paul; Article No. 8; pp. 8:1–8:14
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Determining the similarity between two sets of points in a metric space, and, in general, determining the value of some measure defined for two sets of points, is a well investigated problem in computational geometry. Sometimes, however, the answer that is obtained is meaningless, unless one of the sets undergoes some transformation before performing the computation. In this paper, we consider a family of problems in which the goal is to compute a translation which minimizes some bipartite measure. For example, one of the measures that we consider is the bipartite *diameter*, which is the distance between the farthest bichromatic pair, that is the maximum distance between a point from one set and a point from the other set.

The motivation for studying these problems is twofold. The first, as mentioned, is to find a translation for which the computed value is most meaningful. The second is when we are allowed to translate one of the sets in order to minimize some bipartite measure. In general, problems in which the goal is to find a transformation of a given type that minimizes or maximizes some measure are fundamental in computational geometry and have been studied extensively. It is therefore somewhat surprising that the natural versions that we study here have not been considered before. For example, another measure that we consider is the bipartite *uniformity*, which is the difference between the bipartite diameter and the distance between the closest bichromatic pair. When this difference is small, all bichromatic distances are similar, which is often a desirable property due to its close connection to the notions of fairness and balancing. Thus, the optimization problem in this case is to translate one of the sets to achieve the best possible uniformity.

Formally, let $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_m\}$ be two sets of points in $\mathbb{R}^d$. For the sake of simplicity, we assume that $m = n$, and obtain bounds that depend only on n; however, it is not difficult to adapt our algorithms and bounds to the case where the sets A and B have different sizes. We are interested in problems of the following kind: Find a translation t^* that minimizes some bipartite measure of A and $B + t$ over all translations t, where $B + t$ denotes B translated by t.[1] The main bipartite measures that we consider are (i) *diameter*, denoted $\mathrm{diam}(A, B)$, and defined as $\max\{d(a, b) \mid a \in A, b \in B\}$, where $d(a, b)$ is the Euclidean distance between a and b, (ii) *uniformity*, denoted $\mathrm{uni}(A, B)$, and defined as $\mathrm{diam}(A, B) - d(A, B)$, where $d(A, B) = \min\{d(a, b) \mid a \in A, b \in B\}$, and (iii) *union width*, denoted $\mathrm{union\_width}(A, B)$, and defined as $\mathrm{width}(A \cup B)$, i.e., the width of the union of the two sets.

Notice that while for the (one-sided) Hausdorff distance (see below) one considers the distance from each point of A to its *closest* point in B, for the bipartite diameter measure one considers the distance from each point of A to its *farthest* point in B: The former variant is more relevant when B represents a set of homogeneous facilities, equally acceptable, while the latter variant is more relevant when B represents a set of unique facilities such that it is desirable to be close to all of them.

Related work. When comparing two sets of points of the same size, a natural approach is to find a matching or a mapping of one set to the other, such that the distances between the matched points are small. For instance, in the problem of congruence testing [6, 17],

[1] This class of problems naturally extends to other types of transformations, such as rotations, rigid motions, homothethies, similarity transformations, etc. In this paper, we will confine ourselves to translations, unless otherwise stated.

one needs to decide if there exists a geometric transformation (a combination of translation, rotation, and reflection) that maps a point set A exactly or approximately into a point set B of the same size. Another example is the well-known RMS distance, where the goal is to minimize the sum of squares of distances in a perfect matching between A and B [2].

In general, when one of the sets is larger than the other, we can look for a minimum partial matching, which in some sense corresponds to a copy of the smaller set in the larger one. This version of the problem (under various geometric transformations) was also widely investigated for bottleneck matching [8, 15], RMS distance [7], and more [20].

Another way to compare two sets of points of different sizes, is to use some bipartite distance measure for point sets, such as the well-known Hausdorff distance. The Hausdorff distance between two sets of points is the maximum of the distances from a point in each of the sets to the nearest point in the other set (the one-sided version of Hausdorff distance is a special case of our framework, but we do not consider it here beyond this summary). Huttenlocher et al. [19] showed that the minimum Hausdorff distance under translation in $\mathbb{R}^2$ can be computed in $O(mn(m+n)\alpha(mn)\log(mn))$ time, where m and n are the sizes of the two sets. The minimum Hausdorff distance under geometric transformations was widely investigated in the literature, and we refer to [2] for a survey of the results. A different example of bipartite measure is the maximum overlap between the convex hulls of the sets A and B. This measure was considered in [5], where, assuming A and B are point sets of size n in $\mathbb{R}^3$, an algorithm is presented that computes the optimal translation in expected time $O(n^3 \log^4 n)$.

In this paper, we focus on three bipartite measures under translation: diameter, uniformity, and width. To our knowledge, all three measures are being considered here for the first time.

The diameter of a set of n points in the plane can be computed in $O(n \log n)$ time. However, in higher dimensions the problem becomes much harder. Clarkson and Shor [13] gave a randomized algorithm with expected running time $O(n \log n)$ for points in $\mathbb{R}^3$, which is not very efficient in practice. Then there was a sequence of attempts to find a (simple) deterministic algorithm, which led to an optimal $O(n \log n)$ deterministic algorithm by Ramos [22].

The *width* of a set A of n points in the plane is the smallest distance between a pair of parallel lines, such that the closed strip between the lines contains A, and it can be easily computed in time $O(n \log n)$ using the rotating calipers method. However, again, in three dimensions the problem becomes harder, and the best-known algorithm is an $O(n^{3/2+\varepsilon})$ expected time algorithm, due to Agarwal and Sharir [3].

To compute the uniformity of two point sets under translation, we construct the minimum enclosing annulus of a set of n points in the plane (with only $O(\sqrt{n})$ extreme points). In [3], it is shown that the minimum enclosing annulus of n points in the plane (without a constraint on the number of extreme points) can be computed in $O(n^{3/2+\varepsilon})$ expected time, which is the current state of the art for this problem.

Our results. Consider the set $\mathcal{P} = \{a - b \mid a \in A, b \in B\}$ of all translations that take a point $b \in B$ to a point $a \in A$. We show that the optimal translations in the diameter and uniformity problems are the centers of the minimum enclosing circle of $\mathcal{P}$ and the minimum-width annulus containing $\mathcal{P}$, respectively. Thus, we could apply the best known algorithms for computing these objects to obtain solutions to these problems. More precisely, applying the algorithm of Megiddo [21] for computing the minimum enclosing ball would yield an $O(n^2)$-time solution for the diameter problem, in any fixed dimension, and applying the algorithm of Agarwal and Sharir [3] for computing the minimum-width annulus would yield an $O(n^{3+\varepsilon})$-time solution for the uniformity problem in the plane. However, by making some additional observations and employing sophisticated known techniques, we are able to do much better. Specifically, we solve the diameter problem in $O(n \log n)$ time in the plane and in $O(n \log^2 n)$ expected time in three dimensions, and we solve the uniformity problem in the plane in $O(n^{9/4+\varepsilon})$ expected time, for any $\varepsilon > 0$. As a by-product of the latter result, we show that the minimum enclosing annulus of n points in the plane with only $O(\sqrt{n})$ extreme points can be computed in $O(n^{9/8+\varepsilon})$ expected time (in contrast to $O(n^{3/2+\varepsilon})$ expected time for the unconstrained case, see above).

For the union width problem under translation, we present an $O(n \log n)$-time solution in the plane and an $O(n^2)$-time one in three dimensions. Finally, we consider another new width-based measure, the *red-blue width*. The directional red-blue width (w.r.t. direction v) is the maximum red-blue distance after projecting the points onto a line parallel to direction v. The red-blue width is then defined as the minimum directional red-blue width over all directions. In other words, it measures the width of $A + (-B)$, the Minkowski sum of A and $-B$. We present solutions for the red-blue width problem under translation that run in time $O(n \log n)$ and $O(n^2)$, respectively, in the plane and in three dimensions.

2 Diameter

In the first problem that we consider, the measure is the bipartite diameter. Given two sets of points $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_n\}$ in $\mathbb{R}^d$, the *bipartite diameter* of A and B is $\mathrm{diam}(A, B) = \max\{d(a, b) \mid a \in A, b \in B\}$, where $d(a, b)$ is the Euclidean distance between a and b.

▶ **Problem 1** (Bipartite Diameter under Translation). *Find a translation t^* that minimizes the bipartite diameter of A and $B + t$ over all translations t. That is, for any translation t,* $\mathrm{diam}(A, B + t^*) \le \mathrm{diam}(A, B + t)$*.*

Consider the set $\mathcal{P} = \{a - b \mid a \in A,\, b \in B\}$ of all possible translations taking a point of B to a point of A. Clearly, $|\mathcal{P}| = O(n^2)$.

▷ Claim 1. Let t be a translation and let S_t be the minimum enclosing ball of $\mathcal{P}$ centered at t. Then, the radius r_t of S_t is equal to $\mathrm{diam}(A, B + t)$.

Proof. Since r_t is the radius of the minimum enclosing ball of $\mathcal{P}$ centered at t,

$$\begin{aligned} r_t = \max_{a \in A,\, b \in B} d(a - b, t) &= \max_{a \in A,\, b \in B} \|(a - b) - t\| \\ &= \max_{a \in A,\, b \in B} \|a - (b + t)\| \\ &= \max_{a \in A,\, b \in B} d(a, b + t) = \mathrm{diam}(A, B + t). \end{aligned}$$ ◁

▶ **Corollary 2.** *The optimal translation t^* minimizing bipartite diameter coincides with the center $c = c(\mathcal{P})$ of the minimum enclosing ball $S = S(\mathcal{P})$ of $\mathcal{P}$.*

Notice that Corollary 2 implies that the optimal translation t^* is unique. The minimum enclosing ball of a set of n points can be computed in linear time or expected linear time using, e.g., Megiddo's algorithm [21] or Welzl's randomized algorithm [24], respectively. Therefore, by Corollary 2, one can compute the optimal translation by simply finding c in $O(n^2)$ time. In this section we present near-linear-time algorithms for the problem in two and three dimensions.

Diameter in the plane. Let $\mathcal{Q}$ be the set of extreme points of $\mathcal{P}$. Denote by $\mathrm{CH}(X)$ the convex hull of a point set X.

Since $\mathcal{P}$ is the Minkowski sum of two sets of size n, it is well known [14] that $\mathcal{Q}$ has size $O(n)$ and can be constructed in linear time from $\mathrm{CH}(A)$ and $\mathrm{CH}(B)$ using the rotating calipers method of [18, 23].

Once $\mathcal{Q}$ is constructed, we compute its minimum enclosing disk $S' = S(\mathcal{Q}) = S(\mathcal{P})$.

▶ **Theorem 3.** *Let A and B be two sets of points in $\mathbb{R}^2$, both of size n. A translation t^* that minimizes the bipartite diameter of A and $B + t$ can be found in $O(n \log n)$ time.*

Diameter in three dimensions. We describe an algorithm for computing the minimum enclosing ball of $\mathcal{P}$, without computing $\mathcal{P}$ (whose size may be $\Theta(n^2)$) explicitly. We adapt Clarkson's scheme for solving LP-type problems [12] to the problem of computing the minimum enclosing ball of a set of points; see [1] for a similar-in-spirit adaptation of Clarkson's scheme to an entirely different situation.

The high-level algorithm uses an initially empty set X of points. It repeats the following process until the minimum enclosing ball is found.

1: Pick a random sample $\mathcal{R}$ of $\mathcal{P}$ of size $4n$.
2: Compute the minimum enclosing ball $S = S(\mathcal{R} \cup X)$.
3: Find the set of *violators* V, i.e., the set of all points of $\mathcal{P}$ that are not in S. If $|V| > 2n$ (there are too many violators), go to 1.
4: If $V = \emptyset$, then return S and stop, else $X \leftarrow X \cup V$ and go to 1.

We call an iteration of the algorithm that reaches line 4 "successful." Clarkson's analysis establishes that in each iteration the expected size of V is n. Therefore, for a random choice of $\mathcal{R}$, the probability of the number of violators being at most $2n$ is at least $\frac{1}{2}$, so an iteration is unsuccessful with probability at most $\frac{1}{2}$. In particular, a constant expected number of unsuccessful iterations is followed by a successful one.

On the other hand, it is not difficult to check (see Clarkson's analysis once again) that, when violators are found, one of the violators must be a point defining the minimum enclosing ball. Therefore, the number of successful iterations cannot exceed five: each iteration adds at least one of the points defining the desired ball to X and once all of them are in X, the optimal ball is discovered in line 2, there are no further violators, and the algorithm stops. Therefore the total number of iterations is expected to be $O(1)$ and the size of the set X never grows beyond $O(n)$. Thus in each iteration we invoke a standard minimum-ball algorithm on $O(n)$ points, requiring $O(n)$ expected time.

Next, we describe how to efficiently implement steps 1 and 3. A random sample of $\mathcal{P}$ can be obtained by repeatedly picking random points $a \in A$ and $b \in B$ and returning $a - b$.

The set of violators V can be found by modifying an algorithm by Chazelle et al. [10] for kth nearest neighbor search. First, consider the following problem:

▶ **Problem 2.** *Given two sets A and B, each of n points in $\mathbb{R}^3$, and a distance r, decide whether there are two points $a \in A$ and $b \in B$ with $d(a, b) > r$.*

This problem can be solved in $O(n \log n)$ expected time by the following algorithm:

1: Set $\mathcal{I}_A = \bigcap_{a \in A} D(a, r)$, where $D(a, r)$ is the ball of radius r centered at a, and construct a corresponding inside/outside point-location data structure. (This structure preprocesses the set $\{D(a, r) | a \in A\}$ to facilitate point location queries of the form "Given a point q, is it contained in $\mathcal{I}_A$ or not?"). $\mathcal{I}_A$, together with its corresponding inside/outside point-location data structure, can be computed using the randomized $O(n \log n)$-time algorithm of Clarkson and Shor [13], after which queries are answered in $O(\log n)$ time.

2: If $\mathcal{I}_A = \emptyset$, then clearly there exist two such points. Otherwise, check for each $b \in B$ whether $b \in \mathcal{I}_A$. This can be done by n point-location queries in total $O(n \log n)$ time. If for some $b \in B$, $b \notin \mathcal{I}_A$, there exists some $a \in A$ for which $d(a, b) > r$.

Now we consider the thresholded reporting version of Problem 2:

▶ **Problem 3.** *Given two sets A and B, each of n points in $\mathbb{R}^3$, a distance r and a parameter k, report all the pairs of points $a \in A$, $b \in B$ with $d(a, b) > r$, if there are at most k such pairs. Otherwise, return* `TOO_MANY` *without necessarily listing them.*

The reporting problem can be solved by building a binary tree of point-location data structures. The root of the tree corresponds to $\mathcal{I}_A$. Next, we arbitrarily divide A into two subsets A_1, A_2 of size $n/2$, and build two new point-location data structures, corresponding to $\mathcal{I}_{A_1}$ and $\mathcal{I}_{A_2}$, respectively. Then we continue recursively for A_1 and A_2. The total expected preprocessing time is $O(n \log^2 n)$.

To report the pairs with distance larger than r, we simply query the nodes of the tree as in step 2 of the decision algorithm above. Given some $b \in B$, if $b \in \mathcal{I}_A$, then $d(a, b) \leq r$ for all $a \in A$ and we can stop the search with b. Else, if $b \notin \mathcal{I}_A$, then there exists some $a \in A$ for which $d(a, b) > r$. In this case we check $\mathcal{I}_{A_1}$ and $\mathcal{I}_{A_2}$ recursively. At a leaf, $\mathcal{I}_{\{a\}} = D(a, r)$, so $b \notin \mathcal{I}_{\{a\}}$ means $d(a, b) > r$; in that case, report the pair (a, b). Keep count of the pairs reported so far. If more than k pairs have been reported, stop and return `TOO_MANY`. For any reported pair, we visit $O(\log n)$ nodes of the tree, including the ones where no pairs were reported, and perform a logarithmic-time point-location query at each node. An additional query is performed for every $b \in B$ that is not part of any pair to be reported. Thus the running time of the reporting phase is no more than $O(n \log n + k \log^2 n)$.

The expected total running time of the algorithm is $O((n + k) \log^2 n)$.

▶ **Observation 4.** *Let o be the center of the ball S and r its radius. A point $a - b \in \mathcal{P}$ is in S if and only if $d(a, b + o) \leq r$.*

Proof. The point $a - b$ is in S if and only if $d(a - b, o) \leq r$, and $d(a - b, o) = \|(a - b) - o\| = \|a - (b + o)\| = d(a, b + o)$. ◀

The set of violators V can be found by solving problem 3 with the input A, $B + o = \{b + o \mid b \in B\}$, the radius of S, and $k = 2n$. We summarize our result.

▶ **Theorem 5.** *Let A and B be two sets of points in $\mathbb{R}^3$, both of size n. A translation t^* that minimizes the bipartite diameter of A and $B + t$ can be found in $O(n \log^2 n)$ expected time.*

3 Uniformity

Define $\text{uni}(A, B)$ as the difference between the largest and the smallest distances between a point of A and a point of B. Formally, we set $\text{uni}(A, B) = \text{diam}(A, B) - d(A, B)$, where $d(A, B) = \min\{d(a, b) \mid a \in A, b \in B\}$. The quantity $\text{uni}(A, B)$ measures the uniformity of the red-blue distances. The smaller it is, the more uniform are the distances. One may consider minimizing the ratio rather than the difference of these quantities, which we leave for future research. In this section we consider the following problem:

▶ **Problem 4** (Bipartite Uniformity under Translation). *Find a translation t^* that minimizes the uniformity of A and $B + t$. That is, for any translation t, $\text{uni}(A, B + t^*) \leq \text{uni}(A, B + t)$.*

We study this problem in the plane. Notice that in general t^* may not be unique.

▷ Claim 6. Let c be the center of a minimum-width enclosing annulus of $\mathcal{P}$. Then, $t^* = c$.

Proof. Similarly to the proof of Claim 1, for any translation t, the annulus S_t centered at the point t with radii $\text{diam}(A, B+t)$ and $d(A, B+t)$ (S_t's width is thus $\text{uni}(A, B+t)$), contains all the points of $\mathcal{P}$. Indeed, given some $a - b \in \mathcal{P}$, we have $d(a-b, t) = d(a, b+t)$ and $d(A, B+t) \le d(a, b+t) \le \text{diam}(A, B+t)$. Since c is the center of the minimum-width enclosing annulus of $\mathcal{P}$, we get $\text{uni}(A, B+c) \le \text{uni}(A, B+t)$ for any translation t. ◁

We are thus left with the following algorithmic problem.

▶ **Problem 5** (Restricted Minimum-Width Annulus). *Given a set $\mathcal{P}$ of n^2 points in the plane with only $O(n)$ extreme points, compute the minimum-width annulus covering $\mathcal{P}$.*

Note that if we apply a standard quadratic-time algorithm from the textbook [14] to $\mathcal{P}$ as a black box, we would obtain running time $O(n^4)$. Instead, we could apply the cutting-edge algorithm of Agarwal and Sharir [3] to $\mathcal{P}$, again as a black box, to achieve $O(n^{3+\varepsilon})$ expected running time. But, as we shall see below, we improve these bounds by a more refined use of these and other tools, for the specific situation presented above.

Let $\mathcal{Q} \subset \mathcal{P}$ be the set of extreme points of $\mathcal{P}$. Let $F = \text{FVor}(\mathcal{Q})$ be the farthest-point Voronoi diagram of $\mathcal{Q}$, and let $V = \text{Vor}(\mathcal{P})$ be the closest-point Voronoi diagram of $\mathcal{P}$. We compute V of size $O(n^2)$ in time $O(n^2 \log n)$ and F of size $O(n)$ in time $O(n \log n)$. It is known (see, for example, [14, Section 7.4]) that the center of the minimum-width annulus covering $\mathcal{P}$ must lie at (i) a vertex of F, (ii) a vertex of V, or (iii) an intersection point between an edge of F and an edge of V. Cases (i) and (ii) can be handled in $O(n^2 \log n)$ time. Indeed, one can preprocess both F and V for point location and then locate vertices of each diagram in the other, obtaining the identities of the closest and farthest points of $\mathcal{P}$ for each Voronoi vertex and allowing one to compute the width of the annulus centered at it.

Hereafter we focus on case (iii). Its naïve implementation requires $\Omega(n^3)$ time, as the number of intersections between edges of F and V might be cubic in the worst case. (Indeed, an $O(n^3)$-time algorithm exists that simply overlays F and V. The vertices of the overlay are precisely the points described in cases (i) through (iii) above. We can now process each point in amortized constant time. See Section 4 of [14] for the routine details.)

Complete bipartite clique decomposition To do better, we start by recalling a variant of a classical fact, first observed in [11].

▶ **Fact 7.** *Let C and D be two sets, each consisting of non-crossing line segments in the plane, with $|C| = n$, $|D| = m$, and $n < m$. Then there exists a collection of pairs $\{(C_i, D_i)\}$ such that:*

(a) *$C_i \subset C$ and $D_i \subset D$.*

(b) *For every intersecting pair of segments $(c, d) \in C \times D$, there exists a unique i such that $(c, d) \in C_i \times D_i$.*

(c) *For every i, every segment in C_i intersects every segment in D_i and the slopes of all segments in C_i are larger than the slopes of all segments of D_i, or vice versa.*

(d) *The collection $\{(C_i, D_i)\}$ can be constructed in time $O((n+m)\log^2 n)$.*

(e) *The number of pairs in the collection is $O(n \log n)$.*

(f) *$\sum_i |C_i| = O(n \log^2 n)$ and $\sum_i |D_i| = O(m \log^2 n)$.*

We outline the proof here, as the version we need is slightly more general than the most commonly used one, such as in [1,3] (see [1] for a very similar construction; the distinction is in item (f), where we need separate bounds on $\sum_i |C_i|$ and $\sum_i |D_i|$); the usual assumption is that $n = m$ while in the application below we will set $m = n^2$.

Proof. Construct a 2-level hereditary segment tree on C: Build a segment tree on the segments of C so that each segment appears in $O(\log n)$ nodes and each node ν corresponds to a canonical vertical strip S_ν and a vertically ordered list C_ν of (parts) of segments of C that completely cross S_ν left-to-right. For the second level, store each of the sets C_ν in a separate balanced binary tree T_ν in vertical order; each node μ of T_ν stores a canonical subset C_μ of contiguous segments of C_ν; at the node μ we also store a second set $D_\mu \subset D$ of segments, initially empty. Now query the structure with each segment $d \in D$. It crosses $O(\log n)$ canonical vertical strips completely and its endpoints land in two leaves of the primary tree, which correspond to elementary canonical strips.

For each strip S_ν completely spanned by d, d crosses a contiguous portion of the segments of C_ν, represented by $O(\log n)$ canonical subsets, each corresponding to a node μ in T_ν. We add d to D_μ, for all such choices of ν and μ.

A very similar process handles the endpoints of D.

Having repeated this process for each $d \in D$,[2] we output (C_μ, D_μ) for all secondary tree nodes μ. It is easily verified that a pair of segments $(c, d) \in C \times D$ cross if and only if there is a (unique) μ with $d \in C_\mu$ and $d \in D_\mu$.

The number of nodes μ in the secondary tree of S_ν is $O(|C_\nu|)$ and hence the number of pairs (C_μ, D_μ) is $O(n \log n)$. By construction, each segment $c \in C$ appears in $O(\log^2 n)$ nodes of the structure and we touch $O(\log^2 n)$ nodes when searching for $d \in D$. This implies the bounds on $\sum_\mu |C_\mu|$ and on $\sum_\mu |D_\mu|$. ◀

Reduction to the minimum-"distance" problem between lines in three dimensions. We now use a reinterpretation of the problem, first noticed in [4] and most recently used in [3] to efficiently compute the minimum-width annulus covering a finite point set in the plane.

Lift the points of $\mathcal{P}$ to the standard paraboloid $z = x^2 + y^2$, obtaining the set $\mathcal{P}^*$ and the corresponding set $\mathcal{Q}^* \subset \mathcal{P}^*$; we will use an asterisk to denote a lifted object. As is well known, a minimal disk enclosing $\mathcal{P}$ in the plane corresponds to an upper tangent plane to the convex hull $\mathrm{CH}(\mathcal{P}^*)$ of $\mathcal{P}^*$ (which coincides with the upper convex hull of $\mathcal{Q}^*$), while a maximal disk empty of points of $\mathcal{P}$ corresponds to a lower tangent plane to $\mathrm{CH}(\mathcal{P}^*)$. In case (iii) described above, the upper plane passes through an edge $q_1^* q_2^*$ of the upper hull of $\mathcal{Q}^*$ and the lower plane through an edge $p_1^* p_2^*$ of the lower hull of $\mathrm{CH}(\mathcal{P}^*)$. The two planes are parallel and this event corresponds precisely to the intersection of an edge c of V separating the regions of p_1 and of p_2 and an edge d of F separating the regions of q_1 and of q_2.

It was observed in [4] that the width of the (minimal) annulus containing $\mathcal{P}$ and centered at $c \cap d$ corresponds to the "distance" between two parallel planes passing through the lines supporting edges $q_1^* q_2^*$ and $p_1^* p_2^*$ in $\mathbb{R}^3$; the distance is not measured using the conventional Euclidean metric, but using a different function that satisfies the properties enumerated in [3] (another application of their machinery is for computing the three-dimensional width of a finite point set in $\mathbb{R}^3$; in that application the distance is Euclidean, for pairs of edges that support parallel planes sandwiching the set; see [3] for the details).

In other words, we need to solve the following problem: For all pairs of lines $q_1^* q_2^*$, $p_1^* p_2^*$ supporting upper and lower edges of $\mathrm{CH}(\mathcal{P}^*)$ as above that correspond to a pair of crossing edges of F and V, find the shortest "distance" between the lines $q_1^* q_2^*$ and $p_1^* p_2^*$.

[2] One needs to repeat the process twice: once for d's that are "steeper" than segments of C and once for those "less steep." More precisely, for each strip S_ν, we classify segments d that span S_ν into two classes: Those that cross the left edge of S_ν lower than the right edge, *relative to the segments* of C_ν, and those that cross the left edge higher than the right edge. This way in the final pair (C_μ, D_μ) either all segments of C_μ cross those from D_μ "from below," or all "from above."

Complete bipartite case. Apply Fact 7 to the two sets of edges (line segments) C and D, producing a decomposition into pairs $\{(C_i, D_i)\}$ with the described properties. We now focus on one such pair, (C_i, D_i). By construction, each pair of edges $(c, d) \in C_i \times D_i$ intersect. We now perform the calculation on the corresponding pair of sets of lifted lines (C_i^*, D_i^*), using the "distance" defined above:

▶ **Fact 8** (Agarwal and Sharir [3]). *Given a set X of n lines and a set Y of m lines, so that every line of X lies above every line of Y, the shortest "distance" between a line of X and a line of Y can be computed in expected time $O(n^{3/4+\varepsilon} m^{3/4+\varepsilon} + n^{1+\varepsilon} + m^{1+\varepsilon})$, for any $\varepsilon > 0$.*

In particular, the best annulus width corresponding to points $c \cap d$, with $(c, d) \in (C_i \times D_i)$, corresponds precisely to the shortest "distance" between C_i^*, D_i^* as above and can be computed using Fact 8 in time $O(n_i^{3/4+\varepsilon} m_i^{3/4+\varepsilon} + n_i^{1+\varepsilon} + m_i^{1+\varepsilon})$, where $n_i = |C_i|$ and $m_i = |D_i|$.

Putting it all together. Recall that in our case $m = n^2$. Therefore, the total work required includes $O(n^2 \log n)$ for cases (i) and (ii), $O((n + m) \log^2 n) = O(n^2 \log^2 n)$ for constructing the pairs $\{(C_i, D_i)\}$, and finally the following for processing every pair (C_i, D_i), using Fact 8:

$$\sum_i O(n_i^{3/4+\varepsilon} m_i^{3/4+\varepsilon} + n_i^{1+\varepsilon} + m_i^{1+\varepsilon}),$$

subject to the constraints described in Fact 7. We bound the above expression by

$$O(n^{3\varepsilon}) \cdot \sum_i O(n_i^{3/4} m_i^{3/4} + n_i + m_i),$$

where we have used the facts that $n_i \le n$ and $m_i \le n^2$, for all i. Since $\sum_i n_i = O(n \log^2 n) = o(n^{1+\varepsilon})$ and $\sum_i m_i = O(m \log^2 n) = o(n^{2+\varepsilon})$, the last two terms are bounded by $o(n^{2+\varepsilon})$. We proceed to focus on the larger first term.

Using Hölder's inequality, we have

$$\sum_i m_i^{3/4} n_i^{3/4} = \sum_i m_i^{3/4} (n_i^3)^{1/4} \le (\sum_i m_i)^{3/4} \cdot (\sum_i n_i^3)^{1/4}$$
$$\le O(n^2 \log^2 n)^{3/4} (n^3 \cdot O(\log^2 n))^{1/4} = O(n^{9/4} \log^2 n) = O(n^{9/4+\varepsilon}),$$

where we have used the fact that $\sum_i m_i = O(n^2 \log^2 n)$, $\sum_i n_i = O(n \log^2 n)$, and $n_i \le n$. Plugging everything together, the expected running time of the entire algorithm is $O(n^{9/4+4\varepsilon})$. Replacing ε by $\varepsilon/4$ in the above reasoning, we obtain:

▶ **Theorem 9.** *Given a set $\mathcal{P}$ of n^2 points in the plane that has $O(n)$ extreme points, the total expected time required to compute the minimum-width annulus enclosing $\mathcal{P}$ is $O(n^{9/4+\varepsilon})$, for any positive ε.*

Returning to our original motivation, we conclude:

▶ **Theorem 10.** *Let A and B be two sets of points in $\mathbb{R}^2$, both of size n. A translation t^* that minimizes the uniformity of A and $B + t$ can be found in $O(n^{9/4+\varepsilon})$ time.*

4 Width

In this section, first we minimize the union width measure. The width of a point set is the smallest distance between two parallel supporting hyperplanes of the set. Given two sets of points $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_n\}$ in $\mathbb{R}^d$, the *union width* of A and B is defined as the width of their union, namely, $\text{union\_width}(A, B) = \text{width}(A \cup B)$.

▶ **Problem 6** (Union Width under Translation). *Find a translation t^* that minimizes the union width of A and $B+t$ over all translations t. That is, for any translation t, we have* $\text{union\_width}(A, B+t^*) \leq \text{union\_width}(A, B+t)$.

Directional width. The *directional width* function $\text{width}_v(X)$ of a compact set X in $\mathbb{R}^d$ gives, for every direction v, the distance between the two supporting hyperplanes of X that are orthogonal to v:

$$\text{width}_v(X) = \max_{x_1,x_2 \in X} (x_1 - x_2) \cdot v.$$

In particular, the width of a set corresponds to the minimum of its directional widths. We define the *directional union width* of A and B as the directional width of their union:

$$\text{union\_width}_v(A, B) = \text{width}_v(A \cup B).$$

▶ **Problem 7** (Directional Union Width under Translation). *For a given direction v, find a translation t^* that minimizes the directional union width of A and $B+t$ over all translations t. That is, for any translation t,* $\text{union\_width}_v(A, B+t^*) \leq \text{union\_width}_v(A, B+t)$.

▷ Claim 11. For a given direction v, the minimum directional union width under translation, $\text{union\_width}_v(A, B+t)$, is equal to the maximum of $\text{width}_v(A)$ and $\text{width}_v(B)$.

Proof. To obtain the smallest directional width we translate B so that the slab between the supporting hyperplanes of the wider set contains the other set, then $\text{union\_width}_v(A, B+t)$ will be equal to the directional width of the wider set which means:

$$\text{union\_width}_v(A, B+t) = \max(\text{width}_v(A), \text{width}_v(B)).$$ ◁

This claim reduces Problem 7 to finding the maximum of two directional widths. Now we return to Problem 6, which now reduces to finding the minimum value of the function $\max(\text{width}_v(A), \text{width}_v(B))$ over all directions v. In Sections 4.1 and 4.2, we present an $O(n \log n)$-time algorithm for the two-dimensional and a quadratic-time algorithm for the three-dimensional version of the problem. Finally, in Section 4.3, we define a bipartite measure closely related to width and show that it can be computed using similar methods with slight modifications.

4.1 Width in the plane

To compare the width of the two sets in different directions and measure the union width, we first compute the convex hulls of the two sets in $O(n \log n)$ time. We use the rotating calipers method of [18,23] to construct, for both A and B, their directional width functions $\text{width}_v(A)$ and $\text{width}_v(B)$. Each of these two functions is a piecewise-algebraic function (with a suitable choice of parametrization) of low degree with $\Theta(n)$ breakpoints. Now consider their pointwise maximum defined by $\max(\text{width}_v(A), \text{width}_v(B))$. The global minimum of this function, according to Claim 11, determines the answer to Problem 7. It can be computed by merging the two lists of breakpoints and computing the intersections between the function graphs in each interval; the resulting function still has $O(n)$ breakpoints in total and its minimum can be computed in linear time which results in the following theorem.

▶ **Theorem 12.** *The union width of two n-point sets in the plane can be computed in $O(n \log n)$ time.*

4.2 Width in three dimensions

To better understand the problem, we first review the tools used to compute the standard width of a set in $\mathbb{R}^3$. Recall that computing the width is equivalent to finding the smallest-width slab enclosing the set. In their paper [18], Houle and Toussaint showed that two supporting planes with minimum distance apart pass through either an antipodal vertex-face (VF) pair or an antipodal edge-edge (EE) pair of the convex hull. To compute and compare the antipodal pairs, they used the *Gauss map* (also called the *normal diagram*). In this transformation, which was originally introduced to computational geometry by Brown [9], the convex hull of the point set is mapped to the surface of a unit sphere $\mathbb{S}^2$. Every face is mapped to a point (the direction of its outer normal), every edge is mapped to the great circle arc connecting its two neighboring faces (the locus of the directions of all planes supporting the set at the edge), and every vertex is mapped to a region (the locus of the directions of all supporting planes at the vertex). Then they overlay the upper hemisphere of the Gauss map on the lower hemisphere and compute the intersections between them. We call the resulting diagram the *antipodal diagram*. Each vertex of the overlay corresponds to an antipodal VF or EE pair, and the width can be determined by computing the distance of the antipodal pair at these vertices and choosing the one with the smallest such distance.

The antipodal diagram encodes the antipodal pair of features for all directions and can be viewed as a representation of the directional width function; in particular, it can be used to compute the directional width for any given direction. As mentioned above, Houle and Toussaint showed that the minimum can only occur at the vertices of the antipodal diagram, not in the middle of an edge nor in the interior of a face [18].

To solve Problem 6, we need to represent the antipodal pairs and directional pairs for both sets together. We create the new combined antipodal diagram by overlaying the antipodal diagrams for A and for B.

▶ **Observation 13.** *If the minimum directional union width under translation occurs at direction v^*, then one of the following must occur (as it holds for the maximum of any two functions):*

1. $\text{width}_{v^*}(A) \geq \text{width}_{v^*}(B)$ *and v^* is a local minimum for* $\text{width}_{v^*}(A)$,
2. $\text{width}_{v^*}(B) \geq \text{width}_{v^*}(A)$ *and v^* is a local minimum for* $\text{width}_{v^*}(B)$, *or*
3. $\text{width}_{v^*}(A) = \text{width}_{v^*}(B)$ *and neither function has a local minimum at v^*.*

In cases 1 and 2, the optimal direction is a local optimum of one of the two sets as well and occurs at a vertex of the antipodal diagram. But what happens in case 3? Is it possible that the minimum occurs in the middle of an edge or in the interior of a face? In order to answer these questions we use the following lemma [18]:

▶ **Lemma 14** (Houle and Toussaint [18]). *Let ℓ_1 and ℓ_2 be parallel lines in $\mathbb{R}^3$. Let π_1 and π_2 be distinct parallel planes containing ℓ_1 and ℓ_2, respectively. Then there exists a preferred direction of rotation such that if π_1 and π_2 are rotated about ℓ_1 and ℓ_2, respectively, in that direction to form new parallel planes π_1' and π_2', then $d(\pi_1', \pi_2') < d(\pi_1, \pi_2)$.*

▷ Claim 15. The minimum value of $\max(\text{width}_v(A), \text{width}_v(B))$ cannot occur in the interior of a face of the antipodal diagram.

Proof. Suppose for the sake of contradiction that the optimal direction v^* lies in the interior of a face of the diagram. Being in the interior means each set has an antipodal VV pair in direction v^*. Since a VV pair cannot be an optimal direction for either of the sets separately, according to Observation 13 the directional widths of A and B are equal, and we may translate them so that the two corresponding parallel slabs coincide. Therefore, each supporting plane passes through exactly one vertex from each set. Let π_1 and π_2 be the two supporting planes with $a_1, b_1 \in \pi_1$ and $a_2, b_2 \in \pi_2$. We can translate B so that b_1 is translated to a_1. After translation, let $\ell_2 \subset \pi_2$ be the line through a_2 and b_2 and let $\ell_1 \subset \pi_1$ be the line through a_1 parallel to ℓ_1. According to Lemma 14, there is a direction to rotate the two planes so that they remain supporting for both sets, but the distance between them is reduced, contradicting v^* being the optimal direction. ◁

We proved that minimum union width cannot occur in the interior of a face; however, unlike the width of a single set, the minimum union width may occur in the interior of an antipodal diagram edge.

(An example when this happens will be described in the full version of this paper.) Even though comparing directional width at vertices of the antipodal diagram is not sufficient anymore, the following theorem proves that the union width still can be computed in quadratic time.

▶ **Theorem 16.** *The union width of two n-point sets in three dimensions can be computed in $O(n^2)$ time.*

Proof. Each of the four subdivisions used to create the antipodal diagram for the union width has linear complexity, so their overlay has complexity $O(n^2)$ and can be computed in $O(n^2)$ time using convex subdivision overlay algorithm of Guibas and Seidel [16]. Although the minimum union width can occur at an interior point of a diagram edge, we can still compute it in $O(n^2)$. Directional union width function along each edge has constant complexity and we can find its minimum value in constant time. Since there are at most $O(n^2)$ edges and vertices in the diagram, we can compute the minimum union width in $O(n^2)$ time. ◀

4.3 Red-blue width

We now present a different interpretation of the width of a set, to motivate the definition of a new bipartite measure. Directional width of a point set X in a given direction v is the maximum of all the pairwise distances projected on that direction, $\max_{x_1,x_2 \in X}(x_1 - x_2) \cdot v$. For two sets A and B, we define the *directional red-blue width* as

$$\text{rb\_width}_v(A, B) = \max_{a \in A, b \in B}(a - b) \cdot v,$$

and the *red-blue width* of A and B as the minimum of all the directional red-blue widths:

$$\text{rb\_width}(A, B) = \min_v \text{rb\_width}_v(A, B).$$

▶ **Problem 8** (Red-blue Width under Translation). *Find a translation t^* that minimizes the red-blue width of A and $B + t$ over all translations t. That is, for any translation t, we have* $\text{rb\_width}(A, B + t^*) \leq \text{rb\_width}(A, B + t)$.

▷ Claim 17. For a given direction v, the minimum directional red-blue width under translation is equal to the average of $\text{width}_v(A)$ and $\text{width}_v(B)$.

Proof. Since all the distances are projected on a line parallel to v, one can use the projection of the points to compute the width. The two sets A and B get projected to intervals with lengths equal to $\text{width}_v(A)$ and $\text{width}_v(B)$, respectively. A translation of B will translate its corresponding interval along the line without changing its length. The extreme distances that define the red-blue width are between the leftmost point of A and the rightmost point of B, and vice versa. The maximum of these two distances is always at least $\text{width}_v(A)/2+\text{width}_v(B)/2$ and is realized when the two interval centers are aligned. ◁

So to solve Problem 8, we need to minimize the sum of the two directional widths, rather than their maximum, as in Problem 6. Using the same techniques with slight modifications we obtain the following result; details are omitted in this version.

▶ **Theorem 18.** *The red-blue width of two n-point sets A and B under translation can be computed in $O(n \log n)$ time in the plane and $O(n^2)$ time in the three-dimensional space.*

References

1 Pankaj K. Agarwal, Boris Aronov, Marc van Kreveld, Maarten Löffler, and Rodrigo Silveira. Computing Correlation between Piecewise-Linear Functions. *SIAM J. Comput.*, 42:1867–1887, 2013.

2 Pankaj K. Agarwal, Sariel Har-Peled, Micha Sharir, and Yusu Wang. Hausdorff distance under translation for points and balls. In *Proc. 19th Ann. Symp. Comput. Geometry*, pages 282–291. ACM, 2003.

3 Pankaj K. Agarwal and Micha Sharir. Efficient randomized algorithms for some geometric optimization problems. *Discr. Comput. Geometry*, 16(4):317–337, 1996.

4 Pankaj K. Agarwal, Micha Sharir, and Sivan Toledo. Applications of Parametric Searching in Geometric Optimization. *J. Algorithms*, 17(3):292–318, 1994. doi:10.1006/jagm.1994.1038.

5 Hee-Kap Ahn, Peter Brass, and Chan-Su Shin. Maximum overlap and minimum convex hull of two convex polyhedra under translations. *Computational Geometry*, 40(2):171–177, 2008.

6 Helmut Alt, Kurt Mehlhorn, Hubert Wagener, and Emo Welzl. Congruence, similarity, and symmetries of geometric objects. *Discr. Comput. Geometry*, 3(3):237–256, 1988.

7 Rinat Ben-Avraham, Matthias Henze, Rafel Jaume, Balázs Keszegh, Orit E. Raz, Micha Sharir, and Igor Tubis. Minimum partial-matching and Hausdorff RMS-distance under translation: Combinatorics and algorithms. In *European Symp. Algorithms*, pages 100–111. Springer, 2014.

8 Arijit Bishnu, Sandip Das, Subhas C. Nandy, and Bhargab B. Bhattacharya. Simple algorithms for partial point set pattern matching under rigid motion. *Pattern Recognition*, 39(9):1662–1671, 2006.

9 K. Q. Brown. *Geometric Transforms for Fast Geometric Algorithms.* Ph.D. thesis, Carnegie-Mellon University, Pittsburgh, PA, 1980.

10 Bernard Chazelle, Richard Cole, Franco P. Preparata, and Chee Yap. New upper bounds for neighbor searching. *Information and Control*, 68(1-3):105–124, 1986.

11 Bernard Chazelle, Herbert Edelsbrunner, Leonidas J. Guibas, and Micha Sharir. Algorithms for bichromatic line segment problems and polyhedral terrains. *Algorithmica*, 11:116–132, 1994.

12 Kenneth L. Clarkson. Las Vegas algorithms for linear and integer programming when the dimension is small. *J. ACM*, 42:488–499, 1995.

13 Kenneth L. Clarkson and Peter W. Shor. Applications of random sampling in computational geometry, II. *Discr. Comput. Geometry*, 4(1):387–421, 1989.

14 M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars. *Computational Geometry: Algorithms and Applications.* Springer-Verlag, third edition, 2008.

15 Alon Efrat, Alon Itai, and Matthew J. Katz. Geometry helps in bottleneck matching and related problems. *Algorithmica*, 31(1):1–28, 2001.

16 Leonidas J. Guibas and Raimund Seidel. Computing convolutions by reciprocal search. *Discr. Comput. Geometry*, 2(2):175–193, 1987.

17 Paul J. Heffernan and Stefan Schirra. Approximate decision algorithms for point set congruence. *Computational Geometry*, 4(3):137–156, 1994.

18 M. E. Houle and G. T. Toussaint. Computing the width of a set. *IEEE Trans. Pattern Anal. Mach. Intell.*, PAMI-10(5):761–765, 1988.

19 Daniel P. Huttenlocher, Klara Kedem, and Micha Sharir. The Upper Envelope of Voronoi Surfaces and Its Applications. *Discr. Comput. Geometry*, 9:267–291, 1993. doi:10.1007/BF02189323.

20 Piotr Indyk and Suresh Venkatasubramanian. Approximate congruence in nearly linear time. *Computational Geometry*, 24(2):115–128, 2003.

21 Nimrod Megiddo. Linear-time algorithms for linear programming in R^3 and related problems. *SIAM J. Comput.*, 12(4):759–776, 1983.

22 Edgar A. Ramos. Deterministic algorithms for 3-D diameter and some 2-D lower envelopes. In *Proc. 16th Ann. Symp. Comput. Geometry*, pages 290–299. ACM, 2000.

23 M. I. Shamos. *Computational Geometry.* PhD thesis, Yale University, 1978.

24 Emo Welzl. Smallest Enclosing Disks (balls and Ellipsoids). In *Results and New Trends in Computer Science*, pages 359–370. Springer-Verlag, 1991.