Searching for the Closest-Pair in a Query Translate

Jie Xue
University of Minnesota, Twin Cities, Minneapolis, MN, USA
http://cs.umn.edu/~xuexx193
xuexx193@umn.edu

Yuan Li
Facebook Inc., Seattle, WA, USA
lydxlx@fb.com

Saladi Rahul
University of Illinois at Urbana-Champaign, Urbana, IL, USA
http://cs.umn.edu/~rahuls
saladi.rahul@gmail.com

Ravi Janardan
University of Minnesota, Twin Cities, Minneapolis, MN, USA
http://cs.umn.edu/~janardan
janardan@umn.edu

Abstract

We consider a range-search variant of the closest-pair problem. Let Γ be a fixed shape in the plane. We are interested in storing a given set of n points in the plane in some data structure such that for any specified translate of Γ, the closest pair of points contained in the translate can be reported efficiently. We present results on this problem for two important settings: when Γ is a polygon (possibly with holes) and when Γ is a general convex body whose boundary is smooth. When Γ is a polygon, we present a data structure using $O(n)$ space and $O(\log n)$ query time, which is asymptotically optimal. When Γ is a general convex body with a smooth boundary, we give a near-optimal data structure using $O(n \log n)$ space and $O(\log^2 n)$ query time. Our results settle some open questions posed by Xue et al. at SoCG 2018.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Closest pair, Range search, Geometric data structures, Translation query

Digital Object Identifier 10.4230/LIPIcs.SoCG.2019.61

Related Version A full version of the paper is available at https://arxiv.org/abs/1807.09498.

Funding The research of Jie Xue is supported, in part, by a Doctoral Dissertation Fellowship from the Graduate School of the University of Minnesota.

1 Introduction

The range closest-pair (RCP) problem, as a range-search version of the closest-pair problem, aims to store a given set S of n points in some data structure such that for a specified query range $X \in \mathcal{X}$ chosen from a certain *query space* $\mathcal{X}$, the closest pair of points in $S \cap X$ can be reported efficiently. As a range-search problem, the RCP problem is non-decomposable in the sense that even if the query range X can be written as $X = X_1 \cup X_2$, the closest-pair in $S \cap X$ cannot be determined efficiently knowing the closest-pairs in $S \cap X_1$ and $S \cap X_2$. The non-decomposability makes the problem quite challenging and interesting, as many traditional range-search techniques are inapplicable.

The RCP problem in $\mathbb{R}^2$ has been well-studied over years [1, 4, 6, 7, 10, 11, 13, 14, 15]. Despite of much effort, the query ranges considered are still restricted to very simple shapes, typically orthogonal rectangles and halfplanes. It is then interesting to ask what if the query ranges are of more general shapes. In this paper, we consider a new variant of the RCP problem in which the query ranges are *translates* of a fixed shape (which can be quite general). Formally, let Γ be a fixed shape in $\mathbb{R}^2$ called *base shape* and $\mathcal{L}_\Gamma$ be the collection of all translates of Γ. We investigate the RCP problem with the query space $\mathcal{L}_\Gamma$ (or the $\mathcal{L}_\Gamma$-RCP problem). This type of query, which is for the first time mentioned in [15] (as an open question), is natural and well-motivated. First, in range-search problems, the query spaces considered are usually closed under translation; in this sense, the query space consisting of translates of a single shape seems the most "fundamental" query type. Some of the previously studied query ranges, e.g., quadrants and halfplanes [1, 7, 15], are in fact instances of translation queries (halfplanes can be viewed as translates of an "infinitely" large disc). Also, translation queries find motivation in practice. For instance, in many applications, the user may be interested in the information within a certain distance r from him/her. In this situation, the query ranges are discs of a fixed radius r, i.e., translates of a fixed disc; or more generally, if the distance r is considered under a general distance function induced by a norm $\|\cdot\|$, then the query ranges are translates of a $\|\cdot\|$-disc of radius r. Finally, there is another view of the translation queries: the base shape Γ can be viewed as static while the dataset is translating. With this view, a motivation of the translation queries is to monitor the information in a fixed region (i.e., Γ) for moving points (where the movement pattern only includes translation).

We investigate the problem in two important settings: when Γ is a polygon (possibly with holes) and when Γ is a general convex body whose boundary is smooth (i.e., through each point on the boundary there is a unique tangent line to Γ). Our main goal is to design optimal or near-optimal data structures for the problems in terms of *space cost* and *query time*. The preprocessing of these data structures is left as an open question for future study.

Although we restrict the query ranges to be translates of a fixed shape, the problem is still challenging for a couple of reasons. First, the base shape Γ to be considered is quite general in both of our settings. When Γ is a polygon, it needs not be convex, and indeed can even have holes. In the case where Γ is a general convex body, we only need the aforementioned smoothness of its boundary. Second, we want the RCP data structures to be optimal or near-optimal, namely, use $O(n \cdot \mathrm{poly}(\log n))$ space and have $O(\mathrm{poly}(\log n))$ query time. This is usually difficult for a non-decomposable range-search problem.

1.1 Related work and our contributions

Related work. The closest-pair problem and range search are both classical topics; some surveys can be found in [3, 12]. The RCP problem in $\mathbb{R}^2$ has been studied in prior work [1, 4, 6, 7, 10, 11, 13, 14, 15]. State-of-the-art RCP data structures for quadrant, strip, rectangle, and halfplane queries were given in the recent work [15]. The quadrant and halfplane RCP data structures are optimal (i.e., with linear space and logarithmic query time). The strip RCP data structure uses $O(n \log n)$ space and $O(\log n)$ query time, while the rectangle RCP data structure uses $O(n \log^2 n)$ space and $O(\log^2 n)$ query time. The work [13] considered a colored version of the RCP problem and gave efficient approximate data structures. The paper [14] studied an approximate version of the RCP problem in which the returned answer can be slightly outside the query range.

Our contributions. We investigate a new variant of the RCP problem in which the query ranges are translates of a fixed shape Γ. In the first half of the paper, we assume Γ is a fixed polygon (possibly with holes), and give an RCP data structure for Γ-translation queries using $O(n)$ space and $O(\log n)$ query time, which is asymptotically optimal. In the second half of the paper, we assume Γ is a general convex body with a smooth boundary, and give a near-optimal RCP data structure for Γ-translation queries using $O(n \log n)$ space and $O(\log^2 n)$ query time. The $O(\cdot)$ above hides constants depending on Γ. Our results settle some open questions posed in [15], e.g., the RCP problem with fixed-radius disc queries, etc. In order to design these data structures, we make nontrivial geometric observations and exploit the properties of the problem itself (i.e., we are searching for the *closest-pair* in a *translate*). Many of our intermediate results are of independent interest and can probably be applied to other related problems. We describe our key ideas and techniques in Section 1.3 after establishing relevant notations in Section 1.2.

Organization. Section 1.2 presents the notations and preliminaries used throughout the paper. Section 1.3 gives an overview of the techniques we use to solve the problems. In Section 2, we study the problem when Γ is a polygon. In Section 3, we study the problem when Γ is a general convex body with a smooth boundary. Due to limited space, some proofs and details are omitted; these can be found in the full version [16]. For the convenience of the reader, we give short proof sketches for some technical lemmas.

1.2 Preliminaries

Basic notations and concepts. For $a, b \in \mathbb{R}^2$, we use $\mathrm{dist}(a, b)$ to denote the Euclidean distance between a and b, and use $[a, b]$ to denote the segment connecting a and b. The *length* of a pair $\phi = (a, b)$ of points, denoted by $|\phi|$, is the length of the segment $[a, b]$, i.e., $|\phi| = \mathrm{dist}(a, b)$. For a shape Γ in $\mathbb{R}^2$ and a point $p \in \mathbb{R}^2$, we denote by Γ_p the Γ-translate $p + \Gamma$. We write $\mathcal{L}_\Gamma = \{\Gamma_p : p \in \mathbb{R}^2\}$, i.e., the collection of all Γ-translates.

Candidate pairs. Let S be a set of points in $\mathbb{R}^2$ and $\mathcal{X}$ a collection of ranges. A *candidate pair* in S with respect to $\mathcal{X}$ refers to a pair of points in S that is the closest-pair in $S \cap X$ for some $X \in \mathcal{X}$. We denote by $\Phi(S, \mathcal{X})$ the set of the candidate pairs in S w.r.t. $\mathcal{X}$.

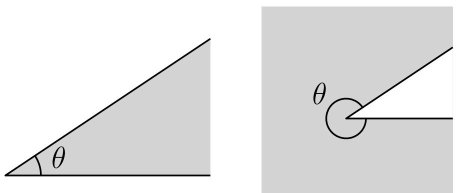


Figure 1 Examples of wedges and co-wedges.

Wedges and co-wedges. A *wedge* is a range in $\mathbb{R}^2$ defined by an angle $\theta \in (0, \pi)$, which is the intersection of two halfplanes (see the left figure in Figure 1). A *co-wedge* is a range in $\mathbb{R}^2$ defined by an angle $\theta \in (\pi, 2\pi)$, which is the union of two halfplanes (see the right figure in Figure 1). The boundary of a wedge or co-wedge W consists of two rays sharing a common initial point, called the two *branches* of W. When appropriate, we refer to wedges and co-wedges collectively as (*co-*)*wedges*.

Convex bodies. A *convex body* in $\mathbb{R}^2$ refers to a compact convex shape with a nonempty interior. If C is a convex body in $\mathbb{R}^2$, we denote by ∂C the boundary of C, which is a simple cycle, and by C° the interior of C, i.e., $C^\circ = C \backslash \partial C$.

The following two lemmas will be used in various places in this paper.

▶ **Lemma 1.** *Let $\varGamma$ be a fixed bounded shape in $\mathbb{R}^2$, and $\mu > 0$ be a constant. Also, let S be a set of points in $\mathbb{R}^2$. Then for any point $p \in \mathbb{R}^2$, either the closest-pair in $S \cap \varGamma_p$ has length smaller than μ, or $|S \cap \varGamma_p| = O(1)$.*

▶ **Lemma 2.** *Let S be a set of points in $\mathbb{R}^2$ and $\mathcal{X}$ be a collection of ranges in $\mathbb{R}^2$. Suppose $(a, b), (a', b') \in \varPhi(S, \mathcal{X})$ are two pairs such that the segments $[a, b]$ and $[a', b']$ cross. Then there exists $X \in \mathcal{X}$ such that either $X \cap \{a, b, a', b'\} = \{a, b\}$ or $X \cap \{a, b, a', b'\} = \{a', b'\}$.*

1.3 Overview of key ideas and techniques

When $\varGamma$ is a polygon (possibly with holes), we solve the problem as follows. First, we use a grid-based approach to reduce the $\mathcal{L}_\varGamma$-RCP problem to the RCP problem with wedge/co-wedge translation queries and the range-reporting problem with $\varGamma$-translation queries. The range-reporting problem can be easily solved by again reducing to the wedge/co-wedge case. Therefore, it suffices to study the RCP problem with wedge/co-wedge translation queries. For both wedge and co-wedge translation queries, we solve the problem by using candidate pairs. Specifically, we store the candidate pairs and search for the answer among them. In this approach, the critical point is the number of the candidate pairs, which determines the performance of our data structures. For both wedge and co-wedge, we prove *linear* upper bounds on the number of the candidate pairs. Although the bounds are the same, the wedge case and co-wedge case require very different proofs, both of which are quite technical and may be of independent geometric interest. These upper bounds and the above-mentioned reduction are our main technical contributions for the polygonal case.

When $\varGamma$ is a general convex body with a smooth boundary, we solve the problem as follows. First, exploiting the smoothness of $\partial\varGamma$, we show that "short" candidate pairs (i.e., of length upper bounded by some constant τ) cannot "cross" each other[1]. It immediately follows that there are only a linear number of short candidate pairs (because they form a planar graph). We try to store these short candidate pairs in a data structure $\mathcal{D}_1$ such that the shortest one contained in any query $\varGamma_q$ can be found efficiently. However, this is a nontrivial task, as $\varGamma$ is quite general here. To this end, we reduce the task of "searching for the shortest pair in $\varGamma_q$" to several point-location queries for q in planar subdivisions. We bound the complexity of these subdivisions (and thus the cost of $\mathcal{D}_1$) by making geometric observations for convex translates and using properties of the pseudo-discs. Using $\mathcal{D}_1$, we can answer any query $\varGamma_q$ in which the closest-pair is short. What if the closest-pair in $\varGamma_q$ is long (i.e., of length greater than τ)? In this case, $\varGamma_q$ contains only $O(1)$ points by Lemma 1. Therefore, if $\mathcal{D}_1$ fails to find the answer, we can simply report the $O(1)$ points contained in $\varGamma_q$ and find the closest-pair by brute-force. The range-reporting is done by point location in the $\leq k$-level of a pseudo-disc arrangement. These are our main contributions for this part.

2 Translation RCP queries for polygons

Let $\varGamma$ be a fixed polygon (possibly with holes). Assume the boundary of $\varGamma$ has no self-intersection[2]. We investigate the $\mathcal{L}_\varGamma$-RCP problem (where the closest-pair is in terms of the Euclidean metric). Throughout this section, $O(\cdot)$ hides constants depending on $\varGamma$. Our main result is the following theorem, to prove which is the goal of this section.

[1] We say two pairs (a, b) and (a', b') cross if the segments $[a, b]$ and $[a', b']$ cross.
[2] That is, the outer boundary and the boundaries of holes are disjoint simple cycles.

▶ **Theorem 3.** *Let Γ be a fixed polygon (possibly with holes) in $\mathbb{R}^2$. Then there is an $O(n)$-space $\mathcal{L}_\Gamma$-RCP data structure with $O(\log n)$ query time.*

Let S be the given dataset in $\mathbb{R}^2$ of size n. Suppose for convenience that the pairwise distances of the points in S are distinct (so the closest-pair in any subset of S is unique).

2.1 Reduction to (co-)wedge translation queries

Our first step is to reduce a Γ-translation RCP query to several wedge/co-wedge translation RCP queries and a range-reporting query. For a vertex v of Γ (either on the outer boundary or on the boundary of a hole), we define a wedge (or co-wedge) W^v as follows. Consider the two edges adjacent to v in Γ. These two edges define two (explementary) angles at v, one of which (say σ) corresponds to the interior of Γ (while the other corresponds to the exterior of Γ). Let W^v be the (co-)wedge defined by σ depending on whether $\sigma < \pi$ or $\sigma > \pi$.

Let $\mathcal{W}_\Gamma = \{W^v : v \text{ is a vertex of } \Gamma\}$. Without loss of generality, suppose that the outer boundary of Γ consists of at least four edges, and so does the boundary of each hole[3]; with this assumption, no three edges of Γ are pairwise adjacent. For two edges e and e' of Γ, let $\mathrm{dist}(e, e')$ denote the minimum distance between one point on e and one point on e'. Define $\delta = \min\{\mathrm{dist}(e, e') : e \text{ and } e' \text{ are non-adjacent edges of } \Gamma\}$. Clearly, δ is a positive constant depending on Γ only. Let $\Box$ be a square of side-length less than $\delta/\sqrt{2}$. Due to the choice of δ, for any $q \in \mathbb{R}^2$, $\Box$ cannot intersect two non-adjacent edges of Γ_q. It follows that $\Box$ intersects at most two edges of Γ_q (as no three edges of Γ are pairwise adjacent); moreover, if $\Box$ intersects two edges, they must be adjacent. Thus, $\Box \cap \Gamma_q = \Box \cap W_q$ for some $W \in \mathcal{W}_\Gamma$.

For a decomposable range-search problem (e.g. range reporting) on S, the above simple observation already allows us to reduce a Γ-translation query to (co-)wedge translation queries (roughly) as follows. Let G be a grid of width $\delta/2$ on the plane. For a cell $\Box$ of G, we define $S_\Box = S \cap \Box$. Due to the decomposability of the problem, to answer a query Γ_q on S, it suffices to answer the query Γ_q on $S_\Box$ for all $\Box$ that intersect Γ_q. Since each cell $\Box$ of G is a square of side-length $\delta/2$ (which is smaller than $\delta/\sqrt{2}$), we have $\Box \cap \Gamma_q = \Box \cap W_q$ for some $W \in \mathcal{W}_\Gamma$ and thus $S_\Box \cap \Gamma_q = S_\Box \cap W_q$. In other words, the query Γ_q on each $S_\Box$ is equivalent to a (co-)wedge translation query for some (co-)wedge $W \in \mathcal{W}_\Gamma$. Applying this idea to range-reporting, we conclude the following.

▶ **Lemma 4.** *There exists an $O(n)$-space range-reporting data structure for Γ-translation queries, which has an $O(\log n + k)$ query time, where k is the number of the reported points.*

However, the above argument fails for a non-decomposable range-search problem, since when the problem is non-decomposable, we are not able to recover efficiently the global answer even if the answer in each cell is known. Unfortunately, our RCP problem belongs to this category. Therefore, more work is required to do the reduction. We shall take advantage of our observation in Lemma 1. We still lay a planar grid G. But this time, we set the width of G to be $\delta/4$. A *quad-cell* $\boxplus$ of G is a square consisting of 2×2 adjacent cells of G. For a quad-cell $\boxplus$ of G, let $S_\boxplus = S \cap \boxplus$. Note that the side-length of a quad-cell of G is $\delta/2$, and each cell of G is contained in exactly four quad-cells of G, so is each point in S. Consider a query range $\Gamma_q \in \mathcal{L}_\Gamma$. The following observation follows from Lemma 1.

▶ **Lemma 5.** *For a quad-cell $\boxplus$ of G such that $|S_\boxplus \cap \Gamma_q| \geq 2$, let $\phi_\boxplus$ be the closest-pair in $S_\boxplus \cap \Gamma_q$. Define ϕ^* as the shortest element among all $\phi_\boxplus$. If the length of ϕ^* is at most $\delta/4$, then ϕ^* is the closest-pair in $S \cap \Gamma_q$; otherwise $|S \cap \Gamma_q| = O(1)$.*

[3] If this is not the case, we can add a new vertex at the midpoint of each edge to "break" it into two.

Using the above observation, we are able to do the reduction.

▶ **Theorem 6.** *Let $f, g : \mathbb{N} \to \mathbb{N}$ be increasing functions where $f(a+b) \geq f(a) + f(b)$. If for any $W \in \mathcal{W}_\Gamma$ there is an $O(f(n))$-space $\mathcal{L}_W$-RCP data structure with $O(g(n))$ query time, then there is an $O(f(n)+n)$-space $\mathcal{L}_\Gamma$-RCP data structure with $O(g(n) + \log n)$ query time.*

Proof. For a quad-cell $\boxplus$ of G, let $m_\boxplus$ be the number of the points in $S_\boxplus$. First, we notice that there are $O(n)$ quad-cells $\boxplus$ of G such that $m_\boxplus > 0$ since each point in S is contained in at most four quad-cells; we call them *nonempty* quad-cells. For each nonempty quad-cell $\boxplus$ and each $W \in \mathcal{W}_\Gamma$, we build an $\mathcal{L}_W$-RCP data structure on $S_\boxplus$; by assumption, this data structure uses $O(f(m_\boxplus))$ space. Now observe that $m_\boxplus \leq n$ for all $\boxplus$ and $\sum m_\boxplus \leq 4n$. From the condition $f(a+b) \geq f(a)+f(b)$, it follows that $\sum f(m_\boxplus) = O(f(n))$. Since $|\mathcal{W}_\Gamma| = O(1)$, the total space cost of these data structures is $O(f(n))$. Besides these data structures, we also build a range-reporting data structure on S for Γ-translation queries. As argued in Lemma 4, this data structure uses $O(n)$ space.

To answer a query $\Gamma_q \in \mathcal{L}_\Gamma$, we first find all nonempty quad-cells of G that intersect Γ_q. The number of these quad-cells is $O(1)$, as it is bounded by $O(\Delta^2/\delta^2)$ where Δ is the diameter of Γ. These quad-cells can be found in $O(\log n)$ time (see [16]). For each such quad-cell $\boxplus$, we find $W \in \mathcal{W}_\Gamma$ such that $\boxplus \cap \Gamma_q = \boxplus \cap W_q$ and query the $\mathcal{L}_W$-RCP data structure built on $S_\boxplus$ to obtain the closest-pair $\phi_\boxplus$ in $S_\boxplus \cap \Gamma_q$, which takes $O(g(m_\boxplus))$ time. Since only $O(1)$ quad-cells are considered, the time for this step is $O(g(n))$. Once these $\phi_\boxplus$ are computed, we take the shortest element ϕ^* among them. If the length of ϕ^* is at most $\delta/4$, then ϕ^* is the closest-pair in $S \cap \Gamma_q$ by Lemma 5 and we just report ϕ^*. Otherwise, $|S \cap \Gamma_q| = O(1)$ by Lemma 5. We then compute the $O(1)$ points in $S \cap \Gamma_q$ using the range-reporting data structure, and compute the closest-pair in $S \cap \Gamma_q$ by brute-force (in constant time). Since the query time of the range-reporting data structure is $O(\log n + k)$ and $k = O(1)$ here, the overall query time is $O(g(n) + \log n)$, as desired. ◀

By the above theorem, it now suffices to give efficient RCP data structures for wedge and co-wedge translation RCP queries. We resolve these problems in the following two sections.

2.2 Handling wedge translation queries

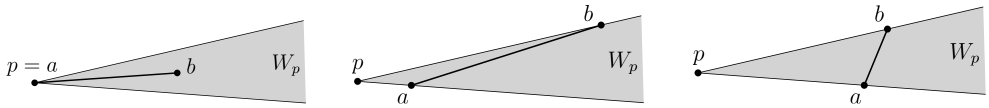
Let W be a fixed wedge in $\mathbb{R}^2$ and $\theta \in (0, \pi)$ be the angle of W. We denote by r and r' the two branches of W. For convenience, assume the vertex of W is the origin, and thus the vertex of a W-translate W_p is the point p. In this section, we shall give an $O(n)$-space $\mathcal{L}_W$-RCP data structure with $O(\log n)$ query time.

The key ingredient of our result is a nontrivial linear upper bound for the number of the candidate pairs in S with respect to $\mathcal{L}_W$. This generalizes a result in [7], and requires a much more technical proof. Before working on the proof, we first establish an easy fact.

▶ **Lemma 7.** *Let $A \subseteq \mathbb{R}^2$ be a finite set. There exists a (unique) smallest W-translate (under the $\subseteq$-order) that contains A. Furthermore, a W-translate is the smallest W-translate containing A iff it contains A and its two branches both intersect A.*

We notice that if $\phi = (a, b)$ is a pair of points in S and W_p is the smallest W-translate containing $\{a, b\}$ described in Lemma 7, then $\phi \in \Phi(S, \mathcal{L}_W)$ iff ϕ is the closest-pair in $S \cap W_p$. Using Lemma 7, we define the following notions.

▶ **Definition 8.** *Let $\phi = (a, b)$ be a pair of points in $\mathbb{R}^2$, and W_p be the smallest W-translate containing $\{a, b\}$ described in Lemma 7. If $p \notin \{a, b\}$ and the smallest angle of the triangle $\triangle pab$ is $\angle apb$, then we say ϕ is* ***steep****; otherwise, we say ϕ is* ***flat****. See Figure 2 for examples.*



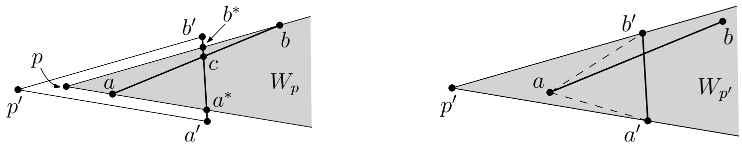
(a) Flat (as $p = a$). **(b)** Flat (as $\angle apb > \angle pba$). **(c)** Steep.

Figure 2 Examples of flat and steep pairs (the wedge W_p shown in the figures is the smallest W-translate containing $\{a, b\}$ described in Lemma 7).

Our first observation is the following.

▶ **Lemma 9.** *If two candidate pairs $\phi, \phi' \in \Phi(S, \mathcal{L}_W)$ cross, then either ϕ or ϕ' is steep.*

Proof. Suppose $\phi = (a, b)$ and $\phi' = (a', b')$. Since ϕ and ϕ' cross, by Lemma 2 there exists some $W_t \in \mathcal{L}_W$ whose intersection with $\{a, b, a', b'\}$ is either $\{a, b\}$ or $\{a', b'\}$; assume $W_t \cap \{a, b, a', b'\} = \{a, b\}$. Let $p \in \mathbb{R}^2$ be the point such that W_p is the smallest W-translate containing $\{a, b\}$. It follows that $W_p \cap \{a, b, a', b'\} = \{a, b\}$, because $W_p \subseteq W_t$. Let c be the intersection point of the segments $[a, b]$ and $[a', b']$. Since $a, b \in W_p$ and W_p is convex, $c \in W_p$. The two endpoints a', b' of the segment $[a', b']$ are not contained in W_p (by assumption), but $c \in W_p$. Hence, the segment $[a', b']$ intersects the boundary of W_p at two points, say a^* and



(a) The points a^*, b^*, and p'. **(b)** Illustrating why (a', b') is steep.

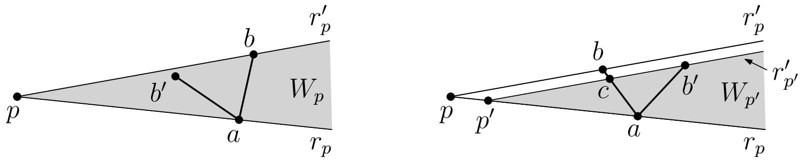
Figure 3 Illustrating Lemma 9.

b^*; assume a^* (resp., b^*) is the point adjacent to a' (resp., b'). Clearly, there exists a unique point $p' \in \mathbb{R}^2$ such that $\triangle pa^*b^* \subseteq \triangle p'a'b'$ and $\triangle pa^*b^*$ is similar to $\triangle p'a'b'$. See Figure 3a. It is easy to see that $W_{p'}$ is the smallest W-translate containing $\{a', b'\}$. Indeed, $W_{p'}$ just corresponds to the angle $\angle a'p'b'$, so a' and b' lie on the two branches of $W_{p'}$ respectively (see Figure 3b). Thus, by the criterion given in Lemma 7, $W_{p'}$ is the smallest W-translate containing $\{a', b'\}$. Now we have $p \in W_{p'}$, which implies $W_p \subseteq W_{p'}$ and $a, b, a', b' \in W_{p'}$. Since the segments $[a, b]$ and $[a', b']$ cross, one of a and b must lie in the triangle $\triangle p'a'b'$, say $a \in \triangle p'a'b'$. Note that $\phi' = (a', b')$ is the closest-pair in $W_{p'}$, thus $\mathrm{dist}(a', b') < \mathrm{dist}(a, a')$ and $\mathrm{dist}(a', b') < \mathrm{dist}(a, b')$. It follows that $\angle a'ab' < \angle ab'a'$ and $\angle a'ab' < \angle aa'b'$. We further observe that $\angle aa'b' < \angle p'a'b'$ and $\angle ab'a' < \angle p'b'a'$, and hence $\angle a'ab' > \angle a'p'b'$. Thus, we have $\angle a'p'b' < \angle p'a'b'$ and $\angle a'pb' < \angle p'b'a'$, i.e., $\angle a'p'b'$ is the smallest angle of the triangle $\triangle p'a'b'$. As a result, ϕ' is steep. ◀

Lemma 9 implies that the flat candidate pairs in $\Phi(S, \mathcal{L}_W)$ do not cross each other. Therefore, the segments corresponding to the flat candidate pairs are edges of a planar graph with vertices in S, which gives a linear upper bound for the number of flat candidate pairs.

It now suffices to bound the number of steep candidate pairs in $\Phi(S, \mathcal{L}_W)$. Unfortunately, two steep candidate pairs (or even one steep candidate pair and one flat candidate pair) can cross, making the above non-crossing argument fail. Therefore, we need some new ideas.

▶ **Definition 10.** *Two pairs $\phi, \phi' \in \Phi(S, \mathcal{L}_W)$ are **adjacent** if we can write $\phi = (a, b)$ and $\phi' = (a, b')$ such that $b \neq b'$; we call $\angle bab'$ the **angle** between ϕ and ϕ', denoted by $\mathrm{ang}(\phi, \phi')$.*



(a) When $b' \in \triangle pab$. **(b)** When $b' \notin \triangle pab$.

Figure 4 Illustrating Lemma 11.

▶ **Lemma 11.** *For adjacent $\phi, \phi' \in \Phi(S, \mathcal{L}_W)$, if ϕ and ϕ' are both steep, then* $\text{ang}(\phi, \phi') \geq \theta$.

Proof sketch. Suppose $\phi = (a, b)$ and $\phi' = (a, b')$. Let $p \in \mathbb{R}^2$ be the point such that W_p is the smallest W-translate containing $\{a, b, b'\}$ described in Lemma 7. We denote by r_p and r'_p the r-branch and r'-branch of W_p, respectively. Using Lemma 7 and the fact that ϕ and ϕ' are steep, we can deduce $p \notin \{a, b, b'\}$; see the complete proof in [16] for details. We distinguish two cases: a is on the boundary of W_p or a is in the interior of W_p. Here we only discuss the first case. The second one is deferred to the complete proof.

Assume a is on the boundary of W_p. Since $p \notin \{a, b, b'\}$, we have $a \neq p$. Thus a must lie on exactly one of r_p and r'_p, say $a \in r_p$. Because W_p is the smallest W-translate containing $\{a, b, b'\}$, one of b and b' must lie on r'_p by the criterion given in Lemma 7. Without loss of generality, assume $b \in r'_p$. Using the criterion in Lemma 7 again, we see that W_p is also the smallest W-translate containing $\{a, b\}$. Thus, ϕ is the closest-pair in $S \cap W_p$ and in particular we have $\text{dist}(a, b) < \text{dist}(b, b')$. It follows that $\angle ab'b < \angle bab' = \text{ang}(\phi, \phi')$. If $b' \in \triangle pab$, we are done, because in this case $\angle ab'b \geq \angle apb = \theta$ and thus $\text{ang}(\phi, \phi') = \angle bab' > \angle ab'b \geq \angle apb = \theta$. See Figure 4a for an illustration of this case. Next, assume $b' \notin \triangle pab$. This case is presented in Figure 4b. Let $p' \in \mathbb{R}^2$ be the point such that $W_{p'}$ is the smallest W-translate containing $\{a, b'\}$; thus, we have $W_{p'} \subseteq W_p$ and in particular $p' \in W_p$. Furthermore, p' must lie on the segment $[p, a]$, as $a \in W_{p'}$. Since $\phi' = (a, b')$ is steep, a and b' lie on the two branches of $W_{p'}$ respectively, and $\angle p'b'a > \angle ap'b' = \theta$. Clearly, b' lies on the r'-branch of $W_{p'}$, which we denote by $r'_{p'}$. Let c be the intersection point of $r'_{p'}$ and the segment $[a, b]$. See Figure 4b. We then have $\angle ab'b = \angle ab'c + \angle cb'b \geq \angle ab'c = \angle pb'a > \angle ap'b' = \theta$. Using the fact $\angle ab'b < \angle bab' = \text{ang}(\phi, \phi')$ obtained before, we conclude $\text{ang}(\phi, \phi') \geq \theta$. ◀

For a point $a \in S$, consider the subset $\Psi_a \subseteq \Phi(S, \mathcal{L}_W)$ consisting of all steep candidate pairs having a as one point. We claim $|\Psi_a| = O(1)$. Suppose $\Psi_a = \{\psi_1, \dots, \psi_r\}$ where $\psi_i = (a, b_i)$ and $b_1, \dots, b_r$ are sorted in polar-angle order around a. By Lemma 11, $\text{ang}(\psi_i, \psi_j) \geq \theta$ for any distinct $i, j \in \{1, \dots, r\}$. Since $\sum_{i=1}^{r-1} \text{ang}(\psi_i, \psi_{i+1}) \leq 2\pi$, we have $r \leq 2\pi/\theta + 1 = O(1)$. As such, $\sum_{a \in S} |\Psi_a| = O(n)$, implying that the number of steep candidate pairs is linear. As the numbers of flat and steep candidate pairs are both linear, we conclude the following.

▶ **Lemma 12.** *$|\Phi(S, \mathcal{L}_W)| = O(n)$, where $n = |S|$.*

Suppose $\Phi(S, \mathcal{L}_W) = \{\phi_1, \dots, \phi_m\}$ where $\phi_i = (a_i, b_i)$ and $\phi_1, \dots, \phi_m$ are sorted in increasing order of their lengths. We have $m = O(n)$ by Lemma 12. Now we only need a data structure which can report, for a query $W_q \in \mathcal{L}_W$, the smallest i such that $a_i, b_i \in W_q$ (note that ϕ_i is the closest-pair in $S \cap W_q$). We design this data structure as follows. Let $\tilde{W} = \{(x, y) : (-x, -y) \in W\}$, which is a wedge obtained by rotating W around the origin with angle π. For a point $p \in \mathbb{R}^2$, it is clear that $a_i, b_i \in W_p$ iff $p \in \tilde{W}_{a_i} \cap \tilde{W}_{b_i}$. Since the intersection of finitely many $\tilde{W}$-translates is a $\tilde{W}$-translate, we may write $\tilde{W}_{a_i} \cap \tilde{W}_{b_i} = \tilde{W}_{c_i}$ for some $c_i \in \mathbb{R}^2$. It follows that ϕ_i is contained in W_p iff $p \in \tilde{W}_{c_i}$. By successively overlaying $\tilde{W}_{c_1}, \dots, \tilde{W}_{c_m}$, we obtain a planar subdivision whose cells are $\Sigma_1, \dots, \Sigma_m$ where $\Sigma_i = \tilde{W}_{c_i} \backslash \bigcup_{j=1}^{i-1} \tilde{W}_{c_j}$. This subdivision has $O(m)$ complexity as overlaying a new $\tilde{W}$-translate can create at most two new vertices. The answer for a query W_q is i iff $q \in \Sigma_i$. Therefore, the problem can be solved by building on the subdivision an $O(m)$-space point-location data structure with $O(\log m)$ query time. Since $m = O(n)$, we have the following conclusion.

▶ **Theorem 13.** *There is an $O(n)$-space $\mathcal{L}_W$-RCP data structure with $O(\log n)$ query time.*

2.3 Handling co-wedge translation queries

Let C be a fixed co-wedge in $\mathbb{R}^2$ and W be the *complementary* wedge of C, i.e., the closure of $\mathbb{R}^2 \backslash C$. We denote by r and r' the two branches of C (and also of W). For convenience, assume $r = \{(t, 0) : t \geq 0\}$ and $r' = \{(\alpha t, t) : t \geq 0\}$ for some $\alpha \in \mathbb{R}$. With this assumption, the vertex of C (resp., W) is the origin and the vertex of C_p (resp., W_p) is p for all $p \in \mathbb{R}^2$. In this section, we present an $O(n)$-space $\mathcal{L}_C$-RCP data structure with $O(\log n)$ query time.

Similar to the wedge case, the key step here is to establish a linear upper bound for $|\Phi(S, \mathcal{L}_C)|$. However, the techniques used here are very different. First of all, we exclude from $\Phi(S, \mathcal{L}_C)$ the candidate pairs with respect to halfplanes. Let $\mathcal{H}$ be the collection of halfplanes, and $\Phi^* = \Phi(S, \mathcal{L}_C) \backslash \Phi(S, \mathcal{H})$. It was shown in [1] that $|\Phi(S, \mathcal{H})| = O(n)$. Therefore, it suffices to prove that $|\Phi^*| = O(n)$. For a pair $\phi = (a, b) \in \Phi^*$, define its *associated* C-translate, $\mathrm{Ass}(\phi)$, as the smallest W-translate containing $\{a, b\}$ (Lemma 7). The pairs in Φ^* and their associated C-translates has the following property.

▶ **Lemma 14.** *Let $\phi = (a, b) \in \Phi^*$ and $C_p = \mathrm{Ass}(\phi) \in \mathcal{L}_C$. Then $p \notin \{a, b\}$ and a, b lie on the two branches of C_p respectively. Furthermore, ϕ is the closest-pair in $S \cap C_p$.*

Consider a pair $\phi \in \Phi^*$ and its associated C-translate $C_p = \mathrm{Ass}(\phi)$. By Lemma 14, one point of ϕ lies on the r-branch of C_p and the other lies on the r'-branch of C_p; we call them the *r-point* and *r'-point* of ϕ, respectively. Let $R \subseteq S$ (resp., $R' \subseteq S$) be the subset consisting of all the r-points (resp., r'-points) of the pairs in Φ^*.

▶ **Lemma 15.** *We have $R \cap R' = \emptyset$, and thus the graph $G = (S, \Phi^*)$ is bipartite.*

For a pair $\phi = (a, b) \in \Phi^*$ where $a \in R$ and $b \in R'$, we define a vector $\mathbf{v}_\phi = \overrightarrow{ab}$. Our key lemma is the following. Let $\mathrm{ang}(\cdot, \cdot)$ denote the angle between two vectors.

▶ **Lemma 16.** *Let $\Psi \subseteq \Phi^*$ be a subset such that $\mathrm{ang}(\mathbf{v}_\psi, \mathbf{v}_{\psi'}) \leq \pi/4$ for all $\psi, \psi' \in \Psi$. Then the graph $G_\Psi = (S, \Psi)$ is acyclic, and in particular $|\Psi| = O(n)$.*

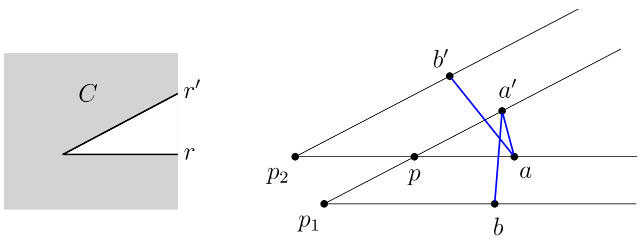


Figure 5 Illustrating Lemma 16.

Proof sketch. Suppose there is a cycle in G_Ψ. Let $\psi = (a, a')$ be the *shortest* edge in the cycle where $a \in R$ and $a' \in R'$. Let $\psi_1 = (b, a') \in \Psi$ and $\psi_2 = (a, b') \in \Psi$ be the two adjacent edges of ψ in the cycle (so $b \in R$ and $b' \in R'$). Let C_p, C_{p_1}, C_{p_2} be the associated C-translates of ψ, ψ_1, ψ_2, respectively. See Figure 5 for an example. We deduce that $a', b, b' \in C_{p_1}$, $C_{p_2} \subseteq C_p$, and $a, a', b, b' \in C_p$; see the complete proof in [16] for an argument. Since $a, a', b, b' \in C_p$ and ϕ is the closest-pair in C_p by Lemma 14, we have $|\phi| < \mathrm{dist}(a, b)$ and hence $\angle aba' < \angle aa'b = \mathrm{ang}(\mathbf{v}_\psi, \mathbf{v}_{\psi_1}) \leq \pi/4$. It follows that

$$\mathrm{ang}(\overrightarrow{ba}, \mathbf{v}_\psi) \leq \mathrm{ang}(\overrightarrow{ba}, \mathbf{v}_{\psi_1}) + \mathrm{ang}(\mathbf{v}_\psi, \mathbf{v}_{\psi_1}) = \angle aba' + \mathrm{ang}(\mathbf{v}_\psi, \mathbf{v}_{\psi_1}) < \pi/2.$$

From this we can further deduce that

$$\mathrm{ang}(\overrightarrow{bb'}, \mathbf{v}_\psi) = \mathrm{ang}(\overrightarrow{ba} + \mathbf{v}_{\psi_2}, \mathbf{v}_\psi) \leq \max\{\mathrm{ang}(\overrightarrow{ba}, \mathbf{v}_\psi), \mathrm{ang}(\mathbf{v}_{\psi_2}, \mathbf{v}_\psi)\} < \pi/2.$$

Next, we establish an inequality that contradicts the above inequality. Let l be the bisector of $[b, b']$. Since $a', b, b' \in C_{p_1}$ and ψ_1 is the closest-pair in C_{p_1}, we have $\mathrm{dist}(a', b') > \mathrm{dist}(b, a')$, i.e., a' is on the same side of l as b. Using the same argument symmetrically, we can deduce that a is on the same side of l as b'. As l is the bisector of $[b, b']$, this implies $\mathrm{ang}(\overrightarrow{bb'}, \mathbf{v}_\psi) = \mathrm{ang}(\overrightarrow{bb'}, \overrightarrow{aa'}) > \pi/2$, which is a contradiction. Thus, G_Ψ is acyclic. ◀

With the above lemma in hand, it is quite straightforward to prove $|\Phi^*| = O(n)$. We evenly separate the plane into 8 sectors $K_1, \dots, K_8$ around the origin. Define $\Psi_i = \{\phi \in \Phi^* : \mathbf{v}_\phi \in K_i\}$ for $i \in \{1, \dots, 8\}$. Now each Ψ_i satisfies the condition in Lemma 16 and thus $|\Psi_i| = O(n)$. Since $\Phi^* = \bigcup_{i=1}^{8} \Psi_i$, we have $|\Phi^*| = O(n)$. Therefore, we conclude the following.

▶ **Lemma 17.** *$|\Phi(S, \mathcal{L}_C)| = O(n)$, where $n = |S|$.*

Suppose $\Phi(S, \mathcal{L}_C) = \{\phi_1, \dots, \phi_m\}$ where $m = O(n)$ and $\phi_1, \dots, \phi_m$ are sorted in increasing order of their lengths. Now we only need a data structure which can report, for a query $C_q \in \mathcal{L}_C$, the smallest i such that ϕ_i is contained in C_q. Similar to the wedge case, we obtain such a data structure with $O(m)$ space and $O(\log m)$ query time (see [16]).

▶ **Theorem 18.** *There is an $O(n)$-space $\mathcal{L}_C$-RCP data structure with $O(\log n)$ query time.*

Theorem 3 now follows from Theorem 6, 13, and 18.

3 Translation RCP queries for smooth convex bodies

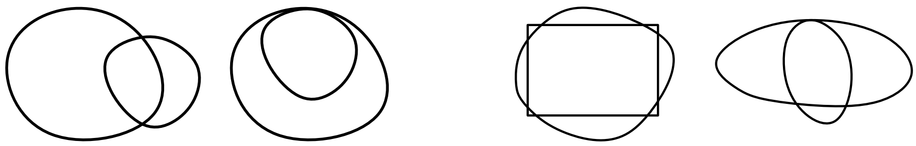
Let Γ be a fixed convex body whose boundary is *smooth* (or *smooth convex body*), that is, through each point on the boundary there is a unique tangent line to Γ. Assume we can compute in $O(1)$ time, for any line l in $\mathbb{R}^2$, the segment $\Gamma \cap l$. We investigate the $\mathcal{L}_\Gamma$-RCP problem (under the Euclidean metric). Throughout this section, $O(\cdot)$ hides constants depending on Γ. Our main result is the following, to prove which is the goal of this section.

▶ **Theorem 19.** *Let Γ be a fixed smooth convex body in $\mathbb{R}^2$. Then there is an $O(n \log n)$-space $\mathcal{L}_\Gamma$-RCP data structure with $O(\log^2 n)$ query time.*

Let S be the given dataset in $\mathbb{R}^2$ of size n. Suppose for convenience that the pairwise distances of the points in S are distinct (so that the closest-pair in any subset of S is unique). Also, suppose that no three points in S are collinear. Our data structure is based on the two technical results presented below, both of which are of geometric interest. The first result states that sufficiently short candidate pairs do not cross when Γ is a smooth convex body.

▶ **Theorem 20.** *Let Γ be a smooth convex body. Then there exists a constant $\tau > 0$ (depending on Γ only) such that if $(a, b), (c, d) \in \Phi(S, \mathcal{L}_\Gamma)$ are two pairs whose lengths are both at most τ, then the segments $[a, b]$ and $[c, d]$ do not cross.*

To introduce the second result, we need an important notion. For two convex bodies C, D in $\mathbb{R}^2$ such that $C \cap D \neq \emptyset$, we say C and D *intersect plainly* if $\partial C \cap D$ and $\partial D \cap C$ are both connected; see Figure 6 for an illustration. (The reader can intuitively understand this as "the boundaries of C and D cross each other at most twice", but it is in fact stronger.) Note that a collection of convex bodies in $\mathbb{R}^2$ in which any two are disjoint or intersect plainly form a family of pseudo-discs [2]. Our second result is the following theorem.



(a) Intersect plainly. **(b)** Not intersect plainly.

Figure 6 An illustration the concept of "intersect plainly".

▶ **Theorem 21.** *Let C be a convex body in $\mathbb{R}^2$, and $p_1, p_2, p_1', p_2' \in \mathbb{R}^2$ be four points (not necessarily distinct) such that $I^\circ \neq \emptyset$ and $I'^\circ \neq \emptyset$, where $I = C_{p_1} \cap C_{p_2}$ and $I' = C_{p_1'} \cap C_{p_2'}$. Suppose that any three of p_1, p_2, p_1', p_2' are not collinear unless two of them coincide. If the segments $[p_1, p_2], [p_1', p_2']$ do not cross and $I \cap I' \neq \emptyset$, then I and I' intersect plainly.*

Figure 7 gives an illustration of Theorem 21 in the case where C is a disc. Note that, even for the disc-case, without the condition that $[p_1, p_2], [p_1', p_2']$ do not cross, one can easily construct a counterexample in which I and I' do not intersect plainly.

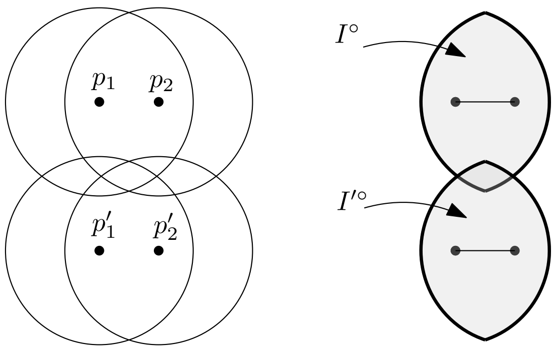


Figure 7 Illustration of Theorem 21 when C is a disc.

The proof of Theorem 20 is presented later in Section 3.3. The proof of Theorem 21 is more involved and is presented the full version [16]. Next, we first assume the correctness of the two theorems and present our $\mathcal{L}_\Gamma$-RCP data structure. Our data structure consists of two parts $\mathcal{D}_1$ and $\mathcal{D}_2$ where $\mathcal{D}_1$ handles the queries for which the length of the answer (closest-pair) is "short" and $\mathcal{D}_2$ handles the queries for which the answer is "long".

3.1 Handling short-answer queries

We describe the first part $\mathcal{D}_1$ of our data structure. Let $\tau > 0$ be the constant in Theorem 20 such that any two candidate pairs of lengths at most τ do not cross. For a query $\Gamma_q \in \mathcal{L}_\Gamma$, $\mathcal{D}_1$ reports the closest-pair in $S \cap \Gamma_q$ if its length is at most τ, and reports nothing otherwise.

Let $\Phi_{\leq\tau} \subseteq \Phi(S, \mathcal{L}_\Gamma)$ be the sub-collection consisting of the candidate pairs of lengths at most τ, and suppose $m = |\Phi_{\leq\tau}|$. We have $m = O(n)$, because the graph $G = (S, \Phi_{\leq\tau})$ is planar by Theorem 20. Define $\tilde{\Gamma} = \{(x, y) \in \mathbb{R}^2 : (-x, -y) \in \Gamma\}$. For a pair $\theta = (a, b) \in \Phi_{\leq\tau}$, we write $I^\theta = \tilde{\Gamma}_a \cap \tilde{\Gamma}_b$. Then θ is contained in a query range $\Gamma_q \in \mathcal{L}_\Gamma$ iff $q \in I^\theta$.

In order to design $\mathcal{D}_1$, we first introduce a so-called *membership* data structure (MDS). Let $\Psi = \{\theta_1, \ldots, \theta_r\} \subseteq \Phi_{\leq\tau}$ and $U = \bigcup_{\theta\in\Psi} I^\theta$. An MDS on Ψ can decide, for a given $\Gamma_q \in \mathcal{L}_\Gamma$, whether Γ_q contains a pair in Ψ or not. As argued before, Γ_q contains a pair in Ψ iff $q \in U$. Thus, to have an MDS on Ψ, it suffices to have a data structure that can decide if a given point is in U. By Theorem 20, the segments corresponding to any two pairs θ_i, θ_j do not cross each other. Also, no three points in S are collinear by assumption. Thus, by Theorem 21, I^{θ_i} and I^{θ_j} intersect plainly for any $i, j \in \{1, \ldots, r\}$ such that $I^{\theta_i} \cap I^{\theta_j} \neq \emptyset$. It follows that $\{I^{\theta_1}, \ldots, I^{\theta_r}\}$ is a family of pseudo-discs [2], and hence the complexity of their union U is $O(r)$ by [8]. As such, optimal point location data structures (e.g., [5, 9]) can be applied to decide whether a point is contained in U in $O(\log r)$ time, using $O(r)$ space. We remark that, although the edges defining the boundary of U are not line-segments (existing point-location results we know of work on polygonal subdivisions), each edge is a connected portion of $\partial\tilde{\Gamma}$ and hence can be decomposed into constant number of "fragments" that are both x-monotone and y-monotone. Recall our assumption that we can compute (in constant time) $\Gamma \cap l$ for any line l in $\mathbb{R}^2$, and thus also $\tilde{\Gamma} \cap l$ for any line l. With this assumption, the existing data structures [5, 9] can be generalized straightforwardly for our purpose. Thus, we have an MDS on Ψ with $O(r)$ space and $O(\log r)$ query time, which we denote by $\mathcal{M}(\Psi)$.

With the MDS in hand, we can now design $\mathcal{D}_1$. For a sub-collection $\Psi = \{\theta_1, \ldots, \theta_r\} \subseteq \Phi_{\leq\tau}$ where $\theta_1, \ldots, \theta_r$ are sorted in increasing order of their lengths, let $\mathcal{D}_1(\Psi)$ be a data structure defined as follows. If $r = 1$, then $\mathcal{D}_1(\Psi)$ simply stores the only pair $\theta_1 \in \Psi$. If $r > 1$, let $\Psi_1 = \{\theta_1, \ldots, \theta_{\lfloor r/2\rfloor}\}$ and $\Psi_2 = \{\theta_{\lfloor r/2\rfloor+1}, \ldots, \theta_r\}$. Then $\mathcal{D}_1(\Psi)$ consists of three parts: $\mathcal{D}_1(\Psi_1)$, $\mathcal{D}_1(\Psi_2)$, and $\mathcal{M}_{\Psi_1}$, where $\mathcal{D}_1(\Psi_1)$ and $\mathcal{D}_1(\Psi_2)$ are defined recursively. We show that we can use $\mathcal{D}_1(\Psi)$ to find, for a query $\Gamma_q \in \mathcal{L}_\Gamma$, the shortest pair $\theta^* \in \Psi$ contained in Γ_q. We first query $\mathcal{M}(\Psi_1)$ to see if Γ_q contains a pair in Ψ_1. If so, θ^* must be in Ψ_1, so we recursively query $\mathcal{D}_1(\Psi_1)$ to find it. Otherwise, we recursively query $\mathcal{D}_1(\Psi_2)$. In this way, we can eventually find θ^*. Now we simply define $\mathcal{D}_1 = \mathcal{D}_1(\Phi_{\leq\tau})$. A direct analysis shows that the space of $\mathcal{D}_1$ is $O(m \log m)$ and the query time of $\mathcal{D}_1$ is $O(\log^2 m)$ (see [16]).

3.2 Handling long-answer queries

If $\mathcal{D}_1$ fails to answer the query Γ_q, then the length of the closest-pair in $S \cap \Gamma_q$ is greater than τ. In this case, we shall use the second part $\mathcal{D}_2$ of our data structure to answer the query. $\mathcal{D}_2$ simply reports all the points in $S \cap \Gamma_q$ and computes the closest-pair by brute-force. Since the length of the closest-pair in $S \cap \Gamma_q$ is greater than τ, we have $|S \cap \Gamma_q| = O(1)$ by Lemma 1 and hence computing the closest-pair takes $O(1)$ time. In order to do reporting, we consider the problem in the dual setting. Again, define $\tilde{\Gamma} = \{(x, y) \in \mathbb{R}^2 : (-x, -y) \in \Gamma\}$. Clearly, for any $a \in \mathbb{R}^2$, $a \in \Gamma_q$ iff $q \in \tilde{\Gamma}_a$. Thus, the problem is equivalent to reporting the ranges in $\mathcal{S} = \{\tilde{\Gamma}_a : a \in S\}$ that contain q. Define the *depth*, $\mathrm{dep}(p)$, of a point $p \in \mathbb{R}^2$ as the number of the ranges in $\mathcal{S}$ containing p. Let $\mathcal{A}$ be the arrangement of the ranges in $\mathcal{S}$, and k be a sufficiently large constant. The $\leq k$-level of $\mathcal{A}$, denoted by $\mathcal{A}_{\leq k}$, is the sub-arrangement of $\mathcal{A}$ contained in the region $R_{\leq k} = \{p \in \mathbb{R}^2 : \mathrm{dep}(p) \leq k\}$. By Theorem 21, any two ranges $\tilde{\Gamma}_a, \tilde{\Gamma}_b \in \mathcal{S}$ intersect plainly if they intersect (setting $p_1 = p_2 = a$ and $p'_1 = p'_2 = b$ when applying Theorem 21), which implies that $\mathcal{S}$ is a family of n pseudo-discs and $\mathcal{A}$ is a pseudo-disc arrangement. So we have the following well-known lemma.

▶ **Lemma 22.** *The complexity of $\mathcal{A}_{\leq k}$ is $O(n)$ for a constant k.*

By the above lemma, we can build a point-location data structure on $\mathcal{A}_{\leq k}$ with $O(n)$ space and $O(\log n)$ query time. Also, we associate to each cell Δ of $\mathcal{A}_{\leq k}$ the (at most k) ranges in $\mathcal{S}$ containing Δ. Now we can report the ranges in $\mathcal{S}$ containing q as follows. Since

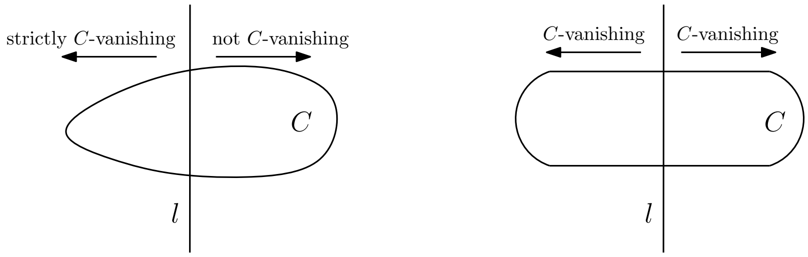
$|S \cap \Gamma_q| = O(1)$ and k is sufficiently large, we have $|S \cap \Gamma_q| \leq k$ and hence q is in $\mathcal{A}_{\leq k}$. Using the point-location data structure, we find in $O(\log n)$ time the cell Δ of $\mathcal{A}_{\leq k}$ containing q. Then the ranges associated to Δ are exactly those containing q. Together with our argument above, this gives us the desired data structure $\mathcal{D}_2$ with $O(n)$ space and $O(\log n)$ time. Combining $\mathcal{D}_2$ with the data structure $\mathcal{D}_1$ in the last section, Theorem 19 is proved.

3.3 Proof of Theorem 20

We begin with introducing some basic notions and geometric results regarding convex bodies in $\mathbb{R}^2$. Let C be a convex body in $\mathbb{R}^2$. For a line l in $\mathbb{R}^2$, we denote by $\mathrm{len}_C(l)$ the length of the segment $C \cap l$. Suppose l is $ax + by + c = 0$, then it cuts $\mathbb{R}^2$ into two halfplanes, $ax + by + c \geq 0$ and $ax + by + c \leq 0$ (called the two *sides* of l hereafter). Let H be one side of l. For a number $t \geq 0$, let l_t denote the (unique) line parallel to l satisfying $l_t \subseteq H$ and $\mathrm{dist}(l, l_t) = t$. Set $\lambda = \sup\{t \geq 0 : C \cap l_t \neq \emptyset\}$ (if $\{t \geq 0 : C \cap l_t \neq \emptyset\} = \emptyset$, set $\lambda = 0$).

▶ **Definition 23.** *We say H is a C-**vanishing** (resp., **strictly** C-**vanishing**) side of l, if $f(t) = \mathrm{len}_C(l_t)$ is a non-increasing (resp., decreasing) function in the domain $[0, \lambda]$.*

Figure 8 gives an intuitive illustration of the above definition. Note that for any convex body C and line l in $\mathbb{R}^2$, at least one side of l is C-vanishing due to the convexity of C.



(a) A line with a strictly C-vanishing side and a side that is not C-vanishing.

(b) A line with two C-vanishing sides.

Figure 8 An illustration of "C-vanishing".

In order to prove Theorem 20, we establish the following key observations.

▶ **Lemma 24.** *Let C be a convex body in $\mathbb{R}^2$, $p_1, p_2 \in \mathbb{R}^2$ be two points, and l be an arbitrary line in $\mathbb{R}^2$. Then we have the following facts.*
(1) If $C_{p_1} \cap l \subsetneq C_{p_2} \cap l$ and V is a C_{p_1}-vanishing side of l, then $C_{p_1} \cap V \subseteq C_{p_2}$.
(2) If $C_{p_1} \cap l$ is contained in the "interior" of $C_{p_2} \cap l$ (i.e., $C_{p_2} \cap l$ excluding both endpoints) and V is a C_{p_1}-vanishing side of l, then $C_{p_1} \cap (V \backslash l) \subseteq C_{p_2} \backslash \partial C_{p_2}$.

▶ **Lemma 25.** *Let C be a smooth convex body in $\mathbb{R}^2$. Then there exists a number $\tau > 0$ such that, for any line l with $0 < \mathrm{len}_C(l) < \tau$ and any point $r \in C$ on a C-vanishing side of l, the distance between r and an (arbitrary) endpoint of $C \cap l$ is less than $\mathrm{len}_C(l)$.*

Now we are able to prove Theorem 20. Suppose Γ is a smooth convex body in $\mathbb{R}^2$. Taking $C = \Gamma$, we can find a constant τ satisfying the condition in Lemma 25. We claim that τ also satisfies the condition in Theorem 20. Let $(a, b), (c, d) \in \Phi(S, \mathcal{L}_\Gamma)$ be two pairs of lengths at most τ. Assume that $[a, b]$ and $[c, d]$ cross. By Lemma 2, there exists $\Gamma_p \in \mathcal{L}_\Gamma$ such that either $\Gamma_p \cap \{a, b, c, d\} = \{a, b\}$ or $\Gamma_p \cap \{a, b, c, d\} = \{c, d\}$; assume $\Gamma_p \cap \{a, b, c, d\} = \{a, b\}$. Suppose (c, d) is the closest-pair in $\Gamma_q \in \mathcal{L}_\Gamma$. Let l be the line through c, d, and V be a

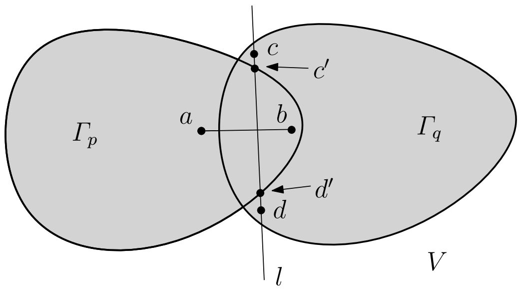


Figure 9 Illustrating the proof of Theorem 20.

Γ_p-vanishing side of l. See Figure 9 for an illustration of the notations. Since the segments $[a, b]$ and $[c, d]$ cross, $[c, d]$ must intersect Γ_p. But $c, d \notin \Gamma_p$, thus $\Gamma_p \cap l \subsetneq [c, d]$. Furthermore, because $[c, d] \subseteq \Gamma_q \cap l$, we have $\Gamma_p \cap l \subsetneq \Gamma_q \cap l$. This implies $\Gamma_p \cap V \subseteq \Gamma_q$ by (1) of Lemma 24. Note that one of a, b must be contained in $\Gamma_p \cap V$, say $b \in \Gamma_p \cap V$. Thus $b \in \Gamma_q$. We now show that $\text{dist}(b, c) < \text{dist}(c, d)$ and $\text{dist}(b, d) < \text{dist}(c, d)$. As $\Gamma_p \cap l \subsetneq [c, d]$ and the length of $[c, d]$ is at most τ, we have $\text{len}_{\Gamma_p}(l) < \tau$. We denote by c', d' the two endpoints of the segment $\Gamma_p \cap l$; see Figure 9. By Lemma 25, the distance from c' (or d') to any point in $\Gamma_p \cap V$ is less than $\text{len}_{\Gamma_p}(l) = \text{dist}(c', d')$. In particular, $\text{dist}(b, c') < \text{dist}(c', d')$ and $\text{dist}(b, d') < \text{dist}(c', d')$. Now $\text{dist}(b, c) \leq \text{dist}(b, c') + \text{dist}(c', c) < \text{dist}(c', d') + \text{dist}(c', c) = \text{dist}(c, d') < \text{dist}(c, d)$. For the same reason, we have $\text{dist}(b, d) < \text{dist}(c, d)$. Since $b, c, d \in \Gamma_q$, this contradicts the fact that (c, d) is the closest-pair in Γ_q. The proof of Theorem 20 is now complete.

References

1 M. A. Abam, P. Carmi, M. Farshi, and M. Smid. On the power of the semi-separated pair decomposition. In *Workshop on Algorithms and Data Structures*, pages 1–12. Springer, 2009.

2 P. K. Agarwal, J. Pach, and M. Sharir. State of the union (of geometric objects): A review. *Computational Geometry: Twenty Years Later. American Mathematical Society*, 2007.

3 Pankaj K Agarwal, Jeff Erickson, et al. Geometric range searching and its relatives. *Contemporary Mathematics*, 223:1–56, 1999.

4 Sang Won Bae and Michiel Smid. Closest-Pair Queries in Fat Rectangles. *arXiv preprint*, 2018. `arXiv:1809.10531`.

5 Herbert Edelsbrunner, Leonidas J Guibas, and Jorge Stolfi. Optimal point location in a monotone subdivision. *SIAM Journal on Computing*, 15(2):317–340, 1986.

6 Prosenjit Gupta. Range-aggregate query problems involving geometric aggregation operations. *Nordic Journal of Computing*, 13(4):294–308, 2006.

7 Prosenjit Gupta, Ravi Janardan, Yokesh Kumar, and Michiel Smid. Data structures for range-aggregate extent queries. *Computational Geometry: Theory and Applications*, 2(47):329–347, 2014.

8 K. Kedem, R. Livne, J. Pach, and M. Sharir. On the union of Jordan regions and collision-free translational motion amidst polygonal obstacles. *Discrete & Computational Geometry*, 1(1):59–71, 1986.

9 N. Sarnak and R. E. Tarjan. Planar point location using persistent search trees. *Communications of the ACM*, 29(7):669–679, 1986.

10 Jing Shan, Donghui Zhang, and Betty Salzberg. On spatial-range closest-pair query. In *International Symposium on Spatial and Temporal Databases*, pages 252–269. Springer, 2003.

11 R. Sharathkumar and P. Gupta. Range-aggregate proximity queries. *Technical Report IIIT/TR/2007/80. IIIT Hyderabad, Telangana*, 500032, 2007.

12 Michiel Smid. Closest-point problems in computational geometry. In *Handbook of computational geometry*, pages 877–935. Elsevier, 2000.

13 Jie Xue. Colored range closest-pair problem under general distance functions. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 373–390. SIAM, 2019.

14 Jie Xue, Yuan Li, and Ravi Janardan. Approximate range closest-pair search. In *Proceedings of the 30th Canadian Conference on Computational Geometry*, pages 282–287, 2018.

15 Jie Xue, Yuan Li, Saladi Rahul, and Ravi Janardan. New Bounds for Range Closest-Pair Problems. In *Proceedings of the 34th International Symposium on Computational Geometry*, pages 73:1–73:14, 2018.

16 Jie Xue, Yuan Li, Saladi Rahul, and Ravi Janardan. Searching for the closest-pair in a query translate. *arXiv preprint*, 2018. `arXiv:1807.09498`.