Approximate range closest-pair search

Jie Xue* Yuan Li† Ravi Janardan ‡

Abstract

The range closest-pair (RCP) problem, as a range-search version of the classical closest-pair problem, aims to store a dataset of points in some data structure such that whenever a query range Q is given, the closest-pair inside Q can be reported efficiently. This paper studies an approximate version of the RCP problem in which the answer pair is allowed to be "approximately" contained in the query range. A general reduction from the approximate RCP problem to the range-minimum and range-reporting problems is given, which works for a general class of query spaces. The reduction is applied to obtain efficient approximate RCP data structures for disk queries in $\mathbb{R}^2$ and ball queries in higher dimensions. Finally, the paper also shows that for orthogonal queries, the approximate RCP problem is (asymptotically) at least as hard as the orthogonal range-minimum problem.

1 Introduction

The closest-pair problem, as one of the most fundamental problems in Computational Geomtry, finds many real-world applications in similarity search and collision detection, etc. In some scenarios, instead of finding the global closest-pair, users are interested in computing the closest-pair inside some specified query range. This results in the so-called *range closest-pair* (RCP) problem, which aims to store a dataset of n points in some data structure such that whenever a query range Q is given, the closest-pair inside Q can be reported efficiently. The RCP problem has been the subject of some recent papers [1, 4, 5, 6, 7, 9].

Unlike most traditional range-search problems, the RCP problem is *non-decomposable.* That is, even if a query range Q can be written as $Q = Q_1 \cup Q_2$, the answer for Q cannot be obtained efficiently from the answers for Q_1 and Q_2. Due to this non-decomposability, many traditional range-search techniques are inapplicable to the RCP problem, which makes the problem quite challenging. Even for very simple query types in $\mathbb{R}^2$ (e.g., quadrants, strips, etc.), the RCP problem is nontrivial. In higher dimensions, it is even not clear how to build efficient data structures for answering RCP queries.

When handling such a difficult range-search problem, approximation could be helpful. In this paper, we study an approximate version of the RCP problem, where the approximation is defined with respect to the query ranges. Specifically, we allow the returned point-pair to be *approximately* (instead of strictly) contained in the query range Q in the sense that one point of the pair can be slightly outside Q but still within a small expansion of Q. For example, consider the disk query in the plane. Given a query disk Q and an approximation factor $\varepsilon > 0$ (which is part of the query), the data structure should return a pair (a, b) of points in the dataset which satisfies the following conditions:
(i) the distance between a and b is at most the distance of the closest-pair in Q.
(ii) $a \in Q$ and $b \in (1+\varepsilon)Q$, where $(1+\varepsilon)Q$ is the disk obtained by expanding Q by a factor $1+\varepsilon$.
Such an approximation can be useful in many real-world applications where the underlying data and/or query is not known precisely anyway. We are interested in how to build efficient data structures for this kind of approximate RCP search.

1.1 Related work

The RCP problem has received attention in recent years [1, 4, 5, 6, 7, 9]. The problem was for the first time introduced in the work [6]. The papers [4, 5, 7] mainly studied the RCP problem in $\mathbb{R}^2$ for orthogonal queries, while the paper [1] considered halfplane queries. Very recently, the previous results were all improved in [9]. In the table below, we summarize the best known bounds for various query types in $\mathbb{R}^2$ (Space refers to the space cost of the data structure and Qtime refers to the query time). In higher dimensions, the RCP problem is quite open. To our best knowledge, even in $\mathbb{R}^3$, no RCP data structure with guaranteed worst-case performance is known currently.

1.2 Our contributions

As mentioned before, in this paper, we study the problem of building efficient data structures for approximate RCP search. Throughout the paper, the query ranges under consideration are always convex bodies (i.e., convex compact subsets) in $\mathbb{R}^d$. Let $\mathcal{Q}$ be a collection of

*University of Minnesota, Twin Cities, xuexx193@umn.edu
†Facebook Inc., lydxlx@fb.com
‡University of Minnesota, Twin Cities, janardan@umn.edu

Query	Source	Space	Qtime
Quadrant	[9]	$O(n)$	$O(\log n)$
Strip	[9]	$O(n \log n)$	$O(\log n)$
Rectangle	[9]	$O(n \log^2 n)$	$O(\log^2 n)$
Halfplane	[9]	$O(n)$	$O(\log n)$

Table 1: Summary of the best known bounds for the RCP problem in $\mathbb{R}^2$.

convex bodies in $\mathbb{R}^d$, called the *query space*. An *approximate* $\mathcal{Q}$-RCP data structure built on a dataset S in $\mathbb{R}^d$ can return, for a specified query (Q, ε) where $Q \in \mathcal{Q}$ is the query range and $\varepsilon > 0$ is the (user-specified) approximation factor, a pair $\phi = (a, b)$ of points in S such that **(i)** $\|b - a\|_2$ is at most the distance of the closest-pair in $S \cap Q$ and **(ii)** $a \in Q$, $b \in (1+\varepsilon)Q$, where $(1+\varepsilon)Q$ is the $(1+\varepsilon)$-expansion of Q (see Section 2 for a precise definition).

Our main contribution is a general reduction from the approximate RCP problem to the range-reporting and range-minimum problems for the same query space (Theorem 1). Our reduction works for any query space $\mathcal{Q}$ (consisting of convex bodies) whose elements have width-diameter ratio lower-bounded by a positive constant (this ratio will be explained Section 2). As concrete applications of the reduction, we obtain efficient approximate RCP data structures for disk queries in $\mathbb{R}^2$ (Corollary 3) and ball queries in higher dimensions (Corollary 4). These query types have not been considered in previous work. Finally, we give a hardness result which shows that, for orthogonal queries, the approximate RCP problem is (asymptotically) at least as hard as the orthogonal range-minimum problem (Theorem 5).

The rest of the paper is organized as follows. Section 2 presents some preliminaries. (We suggest the reader reads this section carefully before moving on.) The general reduction is given in Section 3, while its applications are given in Section 4. In Section 5, we present the hardness result.

2 Preliminaries

Point-pairs and closest-pair. For a pair $\phi = (a, b)$ of points in $\mathbb{R}^d$, the *length* of ϕ, denoted by $|\phi|$, is referred to the distance between a and b, i.e., $|\phi| = \|a-b\|_2$. The *closest-pair* (in a set of points) is the pair of (distinct) points with minimum length. For a point-set S, we denote by $\kappa(S)$ the *closest-pair distance* of S, i.e., the length of the closest-pair in S.

Slabs. A *slab* in $\mathbb{R}^d$ is a closed region bounded by two distinct parallel hyperplanes in $\mathbb{R}^d$. The *thickness* of a slab L, denoted by $\mathrm{thk}(L)$, is the distance between its two bounding hyperplanes. Note that for any slab in $\mathbb{R}^d$, we can always write the equations of its two bounding hyperplanes as $\sum_{i=1}^d a_i x_i + b = -1$ and $\sum_{i=1}^d a_i x_i + b = 1$ for some $a_1, \dots, a_d, b \in \mathbb{R}$.

Diameter, width, and directional width. Let X be a convex body in $\mathbb{R}^d$, and $\mathcal{L}_X$ be the collection of all minimal (with respect to the partial order of "$\subseteq$") slabs enclosing X. For a unit vector $\mathbf{u}$ in $\mathbb{R}^d$, the *directional width* of X in the direction $\mathbf{u}$, denoted by $\mathrm{wid}_{\mathbf{u}}(X)$, is defined as

$$\mathrm{wid}_{\mathbf{u}}(X) = \sup_{x \in X} \langle \mathbf{u}, x \rangle - \inf_{x \in X} \langle \mathbf{u}, x \rangle,$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product. Equivalently, $\mathrm{wid}_{\mathbf{u}}(X)$ is the thickness of the slab $L \in \mathcal{L}_X$ whose two bounding hyperplanes are perpendicular to $\mathbf{u}$. The *diameter* $\mathrm{diam}(X)$ of X is defined as $\mathrm{diam}(X) = \sup_{\mathbf{u}}(\mathrm{wid}_{\mathbf{u}}(X))$ for $\mathbf{u}$ taken over all unit vectors in $\mathbb{R}^d$, while the *width* $\mathrm{wid}(X)$ of X is defined as $\mathrm{wid}(X) = \inf_{\mathbf{u}}(\mathrm{wid}_{\mathbf{u}}(X))$. Equivalently, we can also define the diameter and width as $\mathrm{diam}(X) = \sup_{L \in \mathcal{L}_X}(\mathrm{thk}(L))$ and $\mathrm{wid}(X) = \inf_{L \in \mathcal{L}_X}(\mathrm{thk}(L))$. The *width-diameter ratio* of X, denoted by $\gamma(X)$, is defined as $\gamma(X) = \mathrm{wid}(X)/\mathrm{diam}(X)$. If $\mathcal{X}$ is a collection of convex bodies in $\mathbb{R}^d$, we define the *width-diameter ratio* of $\mathcal{X}$ as $\gamma(\mathcal{X}) = \inf_{X \in \mathcal{X}} \gamma(X)$.

Expansion of a convex body. In order to introduce our result, we need to formally define what we mean by "expanding" a convex body. Let X be a convex body in $\mathbb{R}^d$. If X is a ball, then expanding X (by a factor $\delta \geq 1$) can be simply defined as scaling X with respect to the center of X by a factor of δ. This definition can be naturally generalized to a general convex body as follows. For a slab L bounded by two hyperplanes $\sum_{i=1}^d a_i x_i + b = -1$ and $\sum_{i=1}^d a_i x_i + b = 1$, we define the *$\delta$-expansion* of L (for $\delta \geq 1$), denoted by δL, as the slab bounded by the two hyperplanes $\sum_{i=1}^d a_i x_i + b = -\delta$ and $\sum_{i=1}^d a_i x_i + b = \delta$. Let $\mathcal{L}_X$ be the collection of all minimal (with respect to the partial order of "$\subseteq$") slabs enclosing X. Then we define the *δ-expansion* of X, denoted by δX, as $\delta X = \bigcap_{L \in \mathcal{L}_X} \delta L$. Under this definition, the 1-expansion of X is X itself (since $X = \bigcap_{L \in \mathcal{L}_X} L$). Furthermore, as one can easily verify, $\mathrm{wid}_{\mathbf{u}}(\delta X) = \delta \mathrm{wid}_{\mathbf{u}}(X)$ for any unit vector $\mathbf{u}$.

3 A general reduction

Let $\mathcal{Q}$ be a collection of convex bodies in $\mathbb{R}^d$. Recall that an *approximate* $\mathcal{Q}$-RCP data structure built on a dataset S in $\mathbb{R}^d$ can return, for a specified query (Q, ε) where $Q \in \mathcal{Q}$ is the query range and $\varepsilon > 0$ is the approximation factor, a pair $\phi = (a, b)$ of points in S such that **(i)** $\|b-a\|_2 \leq \kappa(S \cap Q)$ and **(ii)** $a \in Q$, $b \in (1+\varepsilon)Q$. Note that here ε is specified in the query and needs not to be known beforehand.

Our main result is a reduction from the approximate RCP problem to the range-reporting and range-minimum problems for the same query space. This reduction works for any query space $\mathcal{Q}$ (consisting of convex bodies) satisfying $\gamma(\mathcal{Q}) > 0$.

Theorem 1 *Let $\mathcal{Q}$ be a fixed collection of convex bodies in $\mathbb{R}^d$ satisfying $\gamma(\mathcal{Q}) > 0$. Given a range-minimum data structure $\mathcal{D}_1$ and a range-reporting data structure $\mathcal{D}_2$ for query space $\mathcal{Q}$, one can build an approximate $\mathcal{Q}$-RCP data structure $\mathcal{D}$ such that*

- *If the space of $\mathcal{D}_1$ is $s_1(n)$ and the space of $\mathcal{D}_2$ is $s_2(n)$, then the space of $\mathcal{D}$ is $O(s_1(n) + s_2(n))$.*
- *If the query time of $\mathcal{D}_1$ is $q_1(n)$ and the query time of $\mathcal{D}_2$ is $q_2(n, k)$ where k is the number of points to be reported, then the query time of $\mathcal{D}$ is $O(q_1(n)+q_2(n, \varepsilon^{-d})+\varepsilon^{-d}\log(1/\varepsilon))$ where ε is the parameter specified in the query.*
- *If the preprocessing time of $\mathcal{D}_1$ is $p_1(n)$ and the preprocessing time of $\mathcal{D}_2$ is $p_2(n)$, then the preprocessing time of $\mathcal{D}$ is $O(p_1(n) + p_2(n) + n \log n)$.*

The rest of this section is dedicated to proving the above result. To this end, we first describe the construction of the desired data structure in Theorem 1, and then analyze its space, query time, and preprocessing time. Let S be the given dataset in $\mathbb{R}^d$ of size n. We want to build an approximate $\mathcal{Q}$-RCP data structure $\mathcal{D}$ on S, given the range-minimum data structure $\mathcal{D}_1$ and the range-reporting data structure $\mathcal{D}_2$.

Data structure. For a point $a \in S$, let $\mathsf{nn}(a) \in S$ denote the nearest neighbor of a in $S\backslash\{a\}$. We associate the information of $\mathsf{nn}(a)$ with the point a for all $a \in S$. Define a weight function $w : S \to \mathbb{R}$ as $w(a) = \|\mathsf{nn}(a) - a\|_2$. This gives us a weighted dataset $\mathcal{S} = (S, w)$. We build on $\mathcal{S}$ the range-minimum data structure $\mathcal{D}_1$. Also, we build on S the range-reporting data structure $\mathcal{D}_2$. Then our approximate $\mathcal{Q}$-RCP data structure $\mathcal{D}$ (built on S) simply consists of $\mathcal{D}_1$ and $\mathcal{D}_2$.

Query algorithm. Let (Q, ε) be a query, where $Q \in \mathcal{Q}$ is the query range and $\varepsilon > 0$ is the approximation factor. Our query algorithm consists of two phases. In the first phase, we use the range-minimum data structure $\mathcal{D}_1$ to find the point $a^* \in S \cap Q$ with the minimum weight. If $w(a^*) \leq \varepsilon \mathrm{wid}(Q)$, then we report the pair $(a^*, \mathsf{nn}(a^*))$ and terminate the query process. Otherwise, we proceed to the second phase. In the second phase, we use the range-reporting data structure $\mathcal{D}_2$ to report all the points in $S \cap Q$. Then we simply run the standard divide-and-conquer closest-pair algorithm on $S \cap Q$ to find the closest-pair and report it as the answer.

Correctness. Let ϕ be the answer returned by our query algorithm. If ϕ is reported in the second phase, then it is in fact the closest-pair in $S \cap Q$ and hence our algorithm is clearly correct (as $|\phi| = \kappa(S \cap Q)$ and both points of ϕ are contained in Q). Suppose ϕ is reported in the first phase. Then $\phi = (a^*, \mathsf{nn}(a^*))$ where a^* is the point in $S \cap Q$ with the minimum weight. Assume (s, t) is the closest-pair in $S \cap Q$. We see $\|\mathsf{nn}(a^*) - a^*\|_2 = w(a^*) \leq w(s) = \|\mathsf{nn}(s) - s\|_2 \leq \|t - s\|_2 = \kappa(S \cap Q)$. Next, we show that $a^* \in Q$ and $\mathsf{nn}(a^*) \in (1 + \varepsilon)Q$, whence the correctness of our algorithm is verified. We have $a^* \in Q$ by the definition of a^*. To see $\mathsf{nn}(a^*) \in (1 + \varepsilon)Q$, let L be any minimal (with respect to the partial order of "$\subseteq$") slab enclosing Q. The thickness $\mathrm{thk}(L)$ of L is at least $\mathrm{wid}(Q)$. Furthermore, we have $\mathrm{dist}(\mathsf{nn}(a^*), L) \leq \mathrm{dist}(\mathsf{nn}(a^*), Q) \leq \|\mathsf{nn}(a^*) - a^*\|_2 \leq \varepsilon \mathrm{wid}(Q) \leq \varepsilon \mathrm{thk}(L)$, where $\mathrm{dist}(\mathsf{nn}(a^*), L)$ (resp., $\mathrm{dist}(\mathsf{nn}(a^*), Q)$) denotes the minimum distance between $\mathsf{nn}(a^*)$ and a point in L (resp., Q), which is zero when $\mathsf{nn}(a^*) \in L$ (resp., $\mathsf{nn}(a^*) \in Q$). Hence, we have $\mathsf{nn}(a^*) \in (1 + \varepsilon)L$, which implies $\mathsf{nn}(a^*) \in (1 + \varepsilon)Q$ (as the slab L is arbitrarily chosen). See Figure 1 for an intuitive illustration. Since $\|\mathsf{nn}(a^*)-a^*\|_2 \leq \kappa(S\cap Q)$ and $a^* \in Q$, $\mathsf{nn}(a^*) \in (1+\varepsilon)Q$, our algorithm is correct.

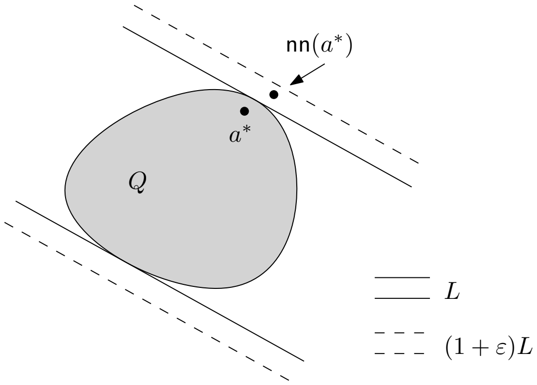

Figure 1: $\mathsf{nn}(a^*) \in (1 + \varepsilon)L$ because $\|\mathsf{nn}(a^*) - a^*\|_2 \leq \varepsilon \mathrm{wid}(Q) \leq \varepsilon \mathrm{thk}(L)$.

Analysis. We now show that the space, query time, and preprocessing time of our data structure $\mathcal{D}$ satisfy the requirements in Theorem 1. The space of $\mathcal{D}$ is clearly $O(s_1(n) + s_2(n))$, as it just consists of $\mathcal{D}_1$ and $\mathcal{D}_2$. To analyze the query time of $\mathcal{D}$, we observe that there are not too many points reported in the second phase.

Lemma 2 *The number of the points reported in the second phase is bounded by $O(\varepsilon^{-d})$.*

Proof. Recall that in the query algorithm, we proceed to the second phase only if $w(a^*) > \varepsilon \mathrm{wid}(Q)$. Since a^* is the point in $S \cap Q$ with the minimum weight, we have $w(a) > \varepsilon \mathrm{wid}(Q)$ for all $a \in S \cap Q$, i.e., $\|\mathsf{nn}(a) - a\|_2 > \varepsilon \mathrm{wid}(Q)$ for all $a \in S \cap Q$. It follows that $\|b - a\|_2 > \varepsilon \mathrm{wid}(Q)$ for any $a, b \in S \cap Q$ unless $a = b$. Now we can show $|S \cap Q| = O(\varepsilon^{-d})$ using the Pigeonhole Principle. Indeed, $\mathrm{diam}(Q) = \mathrm{wid}(Q)/\gamma(Q) \leq \mathrm{wid}(Q)/\gamma(\mathcal{Q})$, hence $\|b - a\|_2 \leq \mathrm{diam}(Q) \leq \mathrm{wid}(Q)/\gamma(\mathcal{Q})$ for any $a, b \in S \cap Q$. So there exists a hyper-cube of side-length

$\mathrm{wid}(Q)/\gamma(\mathcal{Q})$ that contains $S \cap Q$. Because the pairwise distances of the points in $S \cap Q$ are greater than $\varepsilon\mathrm{wid}(Q)$, we have $|S \cap Q| = O((\gamma(\mathcal{Q}) \cdot \varepsilon)^{-d}) = O(\varepsilon^{-d})$ by the Pigeonhole Principle. (The Pigeonhole Principle implies that a set of points in a hyper-cube with side-length α whose pairwise distances are greater than β has size $O((\alpha/\beta)^d)$.) □

The time cost of the first phase is $O(q_1(n))$. By Lemma 2, the time cost for range reporting in the second phase is $O(q_2(n, \varepsilon^{-d}))$. The standard closest-pair algorithm on m points runs in $O(m \log m)$ time, therefore to compute the closest-pair among the reported points takes $O(\varepsilon^{-d} \log(1/\varepsilon))$ time. Thus, the total query time of $\mathcal{D}$ is $O(q_1(n) + q_2(n, \varepsilon^{-d}) + \varepsilon^{-d} \log(1/\varepsilon))$. Finally, we analyze the preprocessing time of $\mathcal{D}$. To build $\mathcal{D}$, we need to compute $\mathsf{nn}(a)$ for all $a \in S$. This can be done using the well-known all-nearest-neighbor algorithm [3], which takes $O(n \log n)$ time. After all $\mathsf{nn}(a)$ are computed, we build $\mathcal{D}_1$ and $\mathcal{D}_2$ directly. Thus, the overall preprocessing time is $O(p_1(n) + p_2(n) + n \log n)$. This completes the proof of Theorem 1.

Discussion. We now briefly discuss which kinds of concrete query spaces our reduction is applicable to. The condition in Theorem 1 for the query space $\mathcal{Q}$ is $\gamma(\mathcal{Q}) > 0$. If $\mathcal{Q}$ is the collection of all balls in $\mathbb{R}^d$ (e.g., all disks in $\mathbb{R}^2$), then $\gamma(\mathcal{Q}) = 1$, and our reduction is applicable; we will discuss this in detail in the next section. More generally, let C be a convex body with nonempty interior in $\mathbb{R}^d$ called *base shape* (note that $\gamma(C) > 0$ in this case). Define $\mathcal{Q}_C$ as the collection of all convex bodies that can be obtained by applying rotation, isotropic scaling, and translation on the base shape C. Then $\gamma(\mathcal{Q}_C) = \gamma(C) > 0$, and our reduction is applicable. We remark that our reduction is inapplicable to the axis-parallel box query, since $\gamma(\mathcal{B}) = 0$ where $\mathcal{B}$ is the collection of all axis-parallel boxes in $\mathbb{R}^d$ (indeed, the width-diameter ratio of a box can be arbitrarily small). However, if we consider a sub-collection $\mathcal{B}_\eta \subseteq \mathcal{B}$ consisting of the boxes in which the ratio of the length of the shortest edge to the length of the longest edge is at least η (where $\eta > 0$), then $\gamma(\mathcal{B}_\eta) \geq \eta/\sqrt{d} > 0$, and our reduction is applicable.

4 Applications

In this section, we apply our general reduction to some specific query spaces to build efficient approximate RCP data structures.

First, we consider the disk query. Let $\mathcal{O}$ be the collection of all disks in $\mathbb{R}^2$. Clearly $\gamma(\mathcal{O}) = 1$ and thus the reduction in Section 3 applies to the query space $\mathcal{O}$. Therefore, to build an approximate $\mathcal{O}$-RCP data structure, it suffices to have the disk range-minimum and range-reporting data structures. It is well-known that the disk range-minimum (resp., range-reporting) problem can be reduced (via lifting) to the halfspace range-minimum (resp., range-reporting) problem in $\mathbb{R}^3$. Halfspace range-reporting in $\mathbb{R}^3$ can be solved optimally (i.e., with $O(n)$ space, $O(\log n + k)$ query time, and $O(n \log n)$ preprocessing time), using the data structure given in [2]. Note that this data structure can also be used to answer halfspace range-emptiness queries in $O(\log n)$ time (i.e., decide whether a given halfspace contains no points in the dataset). By taking advantage of this range-emptiness data structure, we can easily build in $O(n \log^2 n)$ time a halfspace range-minimum data structure in $\mathbb{R}^3$ with $O(n \log n)$ space and $O(\log^2 n)$ query time; we defer the details to Section 4.1. As such, Theorem 1 implies the following corollary.

Corollary 3 *There exists an approximate $\mathcal{O}$-RCP data structure with $O(n \log n)$ space and $O(\log^2 n + \varepsilon^{-2} \log(1/\varepsilon))$ query time, which can be built in $O(n \log^2 n)$ time.*

More generally, we consider the ball query in $\mathbb{R}^d$. Let $\mathcal{O}_d$ be the collection of all disks in $\mathbb{R}^d$ where $d \geq 3$. Again, we have $\gamma(\mathcal{O}_d) = 1$ and thus the reduction in Section 3 works. Similar to the disk case, the range-minimum (resp., range-reporting) problem for query space $\mathcal{O}_d$ can be reduced to the halfspace range-minimum (resp., range-reporting) problem in $\mathbb{R}^{d+1}$. By reducing the halfspace range-minimum problem to the halfspace range-emptiness problem, we can obtain a halfspace range-minimum data structure in $\mathbb{R}^{d+1}$ with $O(n^{\lceil d/2 \rceil})$ space and $O(\log^2 n)$ query time; we defer the details to Section 4.1. The halfspace range-reporting in $\mathbb{R}^{d+1}$ can be solved with $O(n^{\lceil d/2 \rceil} \log^c n)$ space and $O(\log n + k)$ query time [8], where c is a sufficiently large constant. Therefore, Theorem 1 implies the following corollary.

Corollary 4 *There exists an approximate $\mathcal{O}_d$-RCP data structure with $O(n^{\lceil d/2 \rceil} \log^c n)$ space and $O(\log^2 n + \varepsilon^{-d} \log(1/\varepsilon))$ query time.*

While our reduction can be applied to obtain efficient approximate RCP data structures for disk and ball queries, it is unfortunately inapplicable to orthogonal queries (i.e., axis-parallel box queries). In Section 5, we will consider the approximate RCP problem for orthogonal queries and show that it is (asymptotically) at least as hard as the orthogonal range-minimum problem.

4.1 Halfspace range-minimum data structures

We show how to solve the halfspace range-minimum problem via halfspace range-emptiness queries. Suppose there is a halfspace range-emptiness data structure $\mathcal{D}_0$ in $\mathbb{R}^d$, whose space is $s_0(n)$, query time is $q_0(n)$, and preprocessing time is $p_0(n)$. We build a halfspace range-minimum data structure $\mathcal{D}$ in $\mathbb{R}^d$ as follows.

Let $\mathcal{S} = (S, w)$ be a weighted dataset in $\mathbb{R}^d$. Assume $S = \{a_1, \dots, a_n\}$ where $w(a_1) \leq \cdots \leq w(a_n)$. The data structure $\mathcal{D}$ built on $\mathcal{S}$, denoted by $\mathcal{D}(\mathcal{S})$, is constructed recursively. If $n = 1$, then $\mathcal{D}(\mathcal{S})$ is the trivial data structure. Otherwise, let $S_1 = \{a_1, \dots, a_{n/2}\}$ and $S_2 = \{a_{n/2+1}, \dots, a_n\}$. We recursively build $\mathcal{D}(\mathcal{S}_1)$ and $\mathcal{D}(\mathcal{S}_2)$, where $\mathcal{S}_i = (S_i, w_{|S_i})$ ($w_{|S_i}$ denotes the restriction of the weight function w to S_i). Furthermore, we build the halfspace range-emptiness data structure $\mathcal{D}_0(S_1)$. Then $\mathcal{D}(\mathcal{S})$ is the combination of $\mathcal{D}(\mathcal{S}_1)$, $\mathcal{D}(\mathcal{S}_2)$, and $\mathcal{D}_0(S_1)$. If we write the space of $\mathcal{D}$ as $s(n)$ and the preprocessing time (excluding the time for sorting the points by their weights) of $\mathcal{D}$ as $p(n)$, we have the recurrences $s(n) = 2s(n/2) + s_0(n/2)$ and $p(n) = 2p(n/2) + p_0(n/2)$.

To answer a halfspace range-minimum query H using $\mathcal{D}(\mathcal{S})$, we first query $\mathcal{D}_0(S_1)$ to see whether $S_1 \cap H$ is empty. If $S_1 \cap H$ is nonempty, then the answer should be some point in $\mathcal{S}_1$, and thus we can recursively query $\mathcal{D}(\mathcal{S}_1)$ to find it. If $S_1 \cap H$ is empty, the answer should be in $\mathcal{S}_2$, and we can query $\mathcal{D}(\mathcal{S}_2)$ to find it. If we write the query time of $\mathcal{D}$ as $q(n)$, we have the recurrence $q(n) = q(n/2) + q_0(n/2)$.

In $\mathbb{R}^3$, the optimal halfspace range-reporting data structure [2] gives us a halfspace range-emptiness data structure with $s_0(n) = O(n)$ space, $q_0(n) = O(\log n)$ query time, and $p_0(n) = O(n \log n)$ preprocessing time. Thus the above recurrences solve to $s(n) = O(n \log n)$, $q(n) = O(\log^2 n)$, and $p(n) = O(n \log^2 n)$.

In $\mathbb{R}^{d+1}$ for $d \geq 3$, there exists a halfspace range-emptiness data structure with $s_0(n) = O(n^{\lceil d/2 \rceil})$ space and $q_0(n) = O(\log n)$ query time [8]. Thus the above recurrences solve to $s(n) = O(n^{\lceil d/2 \rceil})$ and $q(n) = O(\log^2 n)$.

5 Hardness result for orthogonal queries

In this section, we show that, for orthogonal queries, the approximate RCP data structure is (asymptotically) at least as hard as the range-minimum problem. We use $\mathcal{B}$ to denote the collection of all axis-parallel boxes in $\mathbb{R}^d$. An orthogonal range-minimum (resp., RCP) data structure in $\mathbb{R}^d$ refers to a range-minimum (resp., RCP) data structure for query space $\mathcal{B}$.

Theorem 5 *Given an approximate orthogonal RCP data structure $\mathcal{D}_0$ in $\mathbb{R}^d$, one can build an orthogonal range-minimum data structure $\mathcal{D}$ in $\mathcal{R}^d$ such that*

- *If the space of $\mathcal{D}_0$ is $s(n)$, then the space of $\mathcal{D}$ is $O(s(2n) + n)$.*
- *If the query time of $\mathcal{D}_0$ is $q(n, \varepsilon)$, then the query time of $\mathcal{D}$ is $O(q(2n, 1) + \log n)$.*
- *If the preprocessing time of $\mathcal{D}_0$ is $p(n)$, then the preprocessing time of $\mathcal{D}$ is $O(p(2n) + n \log n)$.*

Proof. Let $\mathcal{S} = (S, w)$ be a weighted dataset in $\mathbb{R}^d$ of size n. We show how to build the desired orthogonal range-minimum data structure $\mathcal{D}$ on $\mathcal{S}$, with $\mathcal{D}_0$ in hand. Suppose $S = \{a_1, \dots, a_n\}$. For convenience, assume $a_1, \dots, a_n$ have distinct coordinates in each dimension and have distinct weights. We keep d sorted lists $\Gamma_1, \dots, \Gamma_d$ of $\{a_1, \dots, a_n\}$, where Γ_j is sorted by the j-th coordinates of the points. For $i \in \{1, \dots, n\}$ and $j \in \{1, \dots, d\}$, we write $t_{i,j}$ as the *rank* of the j-th coordinate of a_i in S, i.e., $t_{i,j} = k$ if the index of a_i in Γ_j is k. Set $\alpha = 1/(100\sqrt{d})$. Define for each $i \in \{1, \dots, n\}$ a hypercube $C_i = \prod_{j=1}^{d} [t_{i,j}, t_{i,j} + \alpha]$. We now choose $2n$ points $b_1, \dots, b_n, b'_1, \dots, b'_n$ such that **(i)** $b_i, b'_i \in C_i$ for all $i \in \{1, \dots, n\}$ and **(ii)** $\|b_i - b'_i\|_2 < \|b_j - b'_j\|_2$ if $w(a_i) < w(a_j)$. See Figure 2 for an intuitive illustration. Our construction above guarantees that $\|b_i - b'_i\|_2 \leq 1/100$ for all $i \in \{1, \dots, n\}$. Furthermore, for distinct $i, j \in \{1, \dots, n\}$, the distance between a point in C_i and a point in C_j is at least 0.5. We let $S' = \{b_1, \dots, b_n, b'_1, \dots, b'_n\}$, and build the orthogonal RCP data structure $\mathcal{D}_0$ on S'. Then the desired data structure $\mathcal{D}$ is just $\mathcal{D}_0$ and the sorted lists $\Gamma_1, \dots, \Gamma_d$.

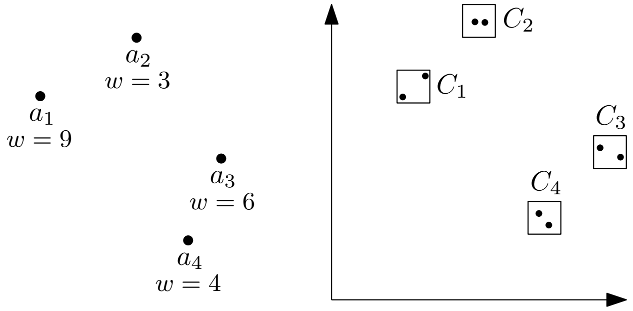

Figure 2: Illustration of the hyper-cubes $C_1, \dots, C_n$. The two points contained in each C_i are b_i and b'_i.

To answer an orthogonal range-minimum query $Q = \prod_{j=1}^{d} [p_j, q_j]$ on $\mathcal{S}$ using $\mathcal{D}$, we first create another box Q' as follows. For all $j \in \{1, \dots, d\}$, let L_j be the slab bounded by the two hyperplanes $x_j = p_j$ and $x_j = q_j$, and define $u_j = \min\{t_{i,j} : a_i \in L_j\}$ and $v_j = \max\{t_{i,j} : a_i \in L_j\}$. Then we set $Q' = \prod_{j=1}^{d} [u_j, v_j + \alpha]$, and query the data structure $\mathcal{D}_0$ with (Q', ε) for $\varepsilon = 1$. (Actually, any $\varepsilon > 0$ works here.) Let ϕ be the pair returned by $\mathcal{D}_0$. Assume that a_k is the point in $S \cap Q$ with the minimum weight, i.e., the answer to the range-minimum query Q. We claim that $\phi = (b_k, b'_k)$. By the construction of Q', for all $i \in \{1, \dots, n\}$, $b_i, b'_i \in Q'$ if $a_i \in Q$ and $b_i, b'_i \notin Q'$ otherwise. Therefore, according to the locations of $b_1, \dots, b_n, b'_1, \dots, b'_n$, we observe that **(i)** the closest-pair in $S' \cap Q'$ is (b_k, b'_k) and **(ii)** the distance between a point in $S' \cap Q'$ and a point in $S' \backslash (S' \cap Q')$ is at least 0.5 (as the two points must be contained in different C_i's). Note that $\|b_k - b'_k\|_2 \leq \sqrt{d}\alpha = 1/100$. It follows that $\|b_k - b'_k\|_2 < \|f - g\|_2$ for any distinct points $f \in S' \cap Q'$ and $g \in S'$. Since $|\phi| \leq \|b_k - b'_k\|_2$ and one point of ϕ must be contained in Q', we have $\phi = (b_k, b'_k)$. As such, with ϕ in hand, we can know a_k and answer the query Q.

Now we analyze the performance of $\mathcal{D}$. The space of $\mathcal{D}$ is clearly $O(s(2n)+n)$, as it consists of $\mathcal{D}_0$ and the sorted lists $\varGamma_1, \dots, \varGamma_d$. The query time of $\mathcal{D}$ consists of the time for constructing Q' and querying $\mathcal{D}_0$. To construct Q', it suffices to compute $u_1, \dots, u_d, v_1, \dots, v_d$, which can be done in $O(\log n)$ time using binary search in $\varGamma_1, \dots, \varGamma_d$. Querying $\mathcal{D}_0$ takes $O(q(2n,1))$ time. Thus the query time of $\mathcal{D}$ is $O(q(2n,1) + \log n)$. The preprocessing of $\mathcal{D}$ can be done in $O(p(2n) + n \log n)$ time. Indeed, we use $O(n \log n)$ time to create the sorted lists $\varGamma_1, \dots, \varGamma_d$. With the lists in hand, we can compute $t_{i,j}$ and S' in linear time. Finally, constructing $\mathcal{D}_0$ takes $O(p(2n))$ time. □

Theorem 5 implies that for orthogonal query ranges, the range minimum problem is (asymptotically) no harder than the approximate RCP problem considered here (or equivalently, the approximate RCP problem is asymptotically at least as hard as the range minimum problem), assuming that $s(n)$ is $\Omega(n)$, $p(n)$ is $\Omega(n \log n)$, and the part of $q(n, \varepsilon)$ that depends on n is $\Omega(\log n)$.

6 Conclusion and future work

In this paper, we studied an approximate version of the RCP problem in which one point of the answer pair is allowed to be slightly outside the query range. We gave a general reduction from the approximate RCP problem to the range-minimum and range-reporting problems, which works for any query space consisting of convex bodies whose width-diameter ratio is lower bounded by a positive constant. By applying our reduction, we obtained efficient approximate RCP data structures for disk and ball queries. Finally, we showed that the approximate RCP problem for orthogonal queries is (asymptotically) at least as hard as the orthogonal range-minimum problem.

Next, we raise an open question for future work. The approximation used in this paper is with respect to the query range. Perhaps, the most natural approximation model is with respect to the quality of the answer. That is, for a specified query range Q, we want to report a $(1+\varepsilon)$-approximate closest-pair in Q, i.e., a pair of points (strictly contained in Q) whose distance is at most $(1+\varepsilon) \cdot \kappa(S \cap Q)$. How to design efficient RCP data structures for this approximation model is an interesting direction for future study.

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. Afshani and T. M. Chan. Optimal halfspace range reporting in three dimensions. In *Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms*, pages 180–186. Society for Industrial and Applied Mathematics, 2009.

[3] P. B. Callahan and S. R. Kosaraju. A decomposition of multidimensional point sets with applications to k-nearest-neighbors and n-body potential fields. *Journal of the ACM (JACM)*, 42(1):67–90, 1995.

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

[5] P. Gupta, R. Janardan, Y. Kumar, and M. Smid. Data structures for range-aggregate extent queries. *Computational Geometry: Theory and Applications*, 2(47):329–347, 2014.

[6] J. Shan, D. Zhang, and B. Salzberg. On spatial-range closest-pair query. In *International Symposium on Spatial and Temporal Databases*, pages 252–269. Springer, 2003.

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

[8] C. D. Toth, J. O'Rourke, and J. E. Goodman. *Handbook of discrete and computational geometry.* Chapman and Hall/CRC, 2017.

[9] J. Xue, Y. Li, S. Rahul, and R. Janardan. New bounds for range closest-pair problems. In *Proceedings of the 34th Symposium on Computational Geometry.* Schloss Dagstuhl-Leibniz-Zentrum fur Informatik GmbH, Dagstuhl Publishing, 2018.