Preprocessing Imprecise Points for Furthest Distance Queries

Vahideh Keikha* Sepehr Moradi† Ali Mohades‡

Abstract

Given is a set of regions in $\mathbb{R}^d$, in the region-based uncertainty model. We show here how to preprocess these regions so that if one point per region is specified with precise coordinates, in the query phase, the diameter of the query points can be computed faster than the scratch. We discuss a $(1+\epsilon)$-approximation algorithm with running time $O(\frac{n}{\epsilon^d})$ for answering such queries, for a set of pairwise disjoint unit balls, after spending $O(n \log n + \frac{n}{\epsilon^d})$ time for preprocessing.

1 Introduction

It is a common assumption in different areas of computational geometry that the input is a set of points. However, we usually face the problems at which the input data are not precise due to several resources, namely including bounded precision of measuring devices, rounding errors in calculations, etc. In some cases, we already know in which *region* each particular point would lie, however, the exact locations of the points are still unknown. One then may assume such a region as an *imprecise point*, that could be a disk, rectangle, line segment, etc.

In recent years, frequent analysis of uncertain data has been actively researched in computation modeling of real-world problems. There are numerous exact and approximation algorithms for processing uncertain data. Designing an exact algorithm that works for all possible instances may produce a big data structure and may need time-consuming calculations. As a result, these algorithms demand much time and space as their inputs are indeed superset compared to the standard algorithm, where the input is a set of points. There have been efforts to resolve this problem by careful analysis of the worst-case or the best-case behavior of the input instances, however, all cases are likely to happen. In some other scenarios, producing the lower and upper bound for feasible solutions may suffice. Another standpoint is preprocessing uncertain data for speeding-up the further computations on precise instances received later.

*Institute of Computer Science, Czech Academy of Sciences, Czech Republic keikha@cs.cas.cz

†Department of Computer Science, Amirkabir University of Technology semolation@gmail.com

‡Department of Computer Science, Amirkabir University of Technology mohades@aut.ac.ir

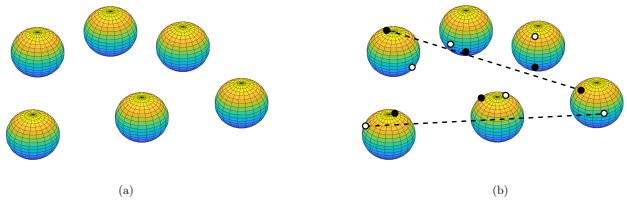

Figure 1: Problem definition: (a) A set D of 6 imprecise points modeled as unit balls, (b) and the diameter of two different realizations of D.

In this paper, we address the diameter problem in this setting.

Problem 1 (Diameter Query) *Let $D = \{d_1, \ldots, d_n\}$ be a set of balls in $\mathbb{R}^d$. For a given query point set $\mathcal{Q} = \{p_1, \ldots, p_n\}$, where $p_i \in d_i$, our objective is to find the diameter of $\mathcal{Q}$ in $o(dn^2)$ time, after preprocessing. We call the set $\mathcal{Q}$ a* realization *of D; see Figure 1.*

Related work. The region-based model of imprecision was introduced and extensively studied by Löffler and van Kreveld. Several models are already established for processing a set of imprecise points for (possibly) speeding up the sorting problem [17], computing an arbitrary triangulation [9, 18], the Delaunay triangulation [2, 11], and the convex hull of a query set in $\mathbb{R}^2$ [4]. In particular, it is previously shown that for a set of imprecise points modeled as convex polygons, with totally $O(n)$ vertices, an arbitrary triangulation of a query set with one point in each region can be computed in $O(n)$ time after spending $O(n \log n)$ for the preprocessing [18]. The same problem was also studied in [11] at which the same results also hold for computing a Delaunay triangulation. See also [2, 9]. For a set of imprecise points in the plane modeled as lines, it is shown that the preprocessing does not speed up the closest pair computation, the Delaunay triangulation, and the sorting problem on the realizations received later [4], where they lie on given lines known in advance. However, in the same paper, it is shown that preprocessing a set of lines, can speed up computing the convex hull of the points (on those lines) received later.

The diameter of a set of points is the maximum pairwise distance between the points in the set. Computing the diameter of a set of points has a long history. It is shown that computing the diameter in $\mathbb{R}^d$ needs

$\Omega(n \log n)$ time in any algebraic decision tree, by a reduction from the set disjointedness problem. But the best-known algorithm for computing the diameter in $\mathbb{R}^d$ takes $O(\min\{nd \log n, n^2 \log^{s-2} n, n^2 d^{s-2}\})$ time, where $s \approx 2.376$ [5, 13]. However, this running time can be improved for specific values of $d > 2$ [5]. For $d = 2$, near linear approximation algorithm exists for the diameter problem [8]. We refer the reader to [5, 12] for a complete list of algorithms for the diameter problem in different dimensions. We note that the diameter problem has extensively used as a black box in database queries. See, e.g., [6].

Contribution. We show there exists a $(1 + \epsilon)$-approximation for approximating the diameter queries on pairwise disjoint unit disks, that takes $O(\frac{n}{\epsilon^d})$ time, after spending $O(n \log n + \frac{n}{\epsilon^d})$ time for preprocessing (Sec 3.2).

2 Preliminaries

For a set $\mathcal{Q}$ of points in $\mathbb{R}^d$, let $diam(S)$ denote the diameter of $\mathcal{Q}$. In the following, we recall the definitions we use from the literature.

Let $G = (S, E)$ be a geometric graph on $\mathcal{Q}$. Let $d_G(p, q)$ denote the geodesic distance between any pair $p, q \in \mathcal{Q}$, that is defined as the length of the shortest path between these two points in G. The graph G is called a *t-spanner* for some $t \geq 1$, if for any two points $p, q \in \mathcal{Q}$ we have $d_G(p, q) \leq t|pq|$, where $|pq|$ is the Euclidiean distance between p and q. The parameter t is refereed to as the stretch factor.

2.1 Well Separated Pair Decomposition (WSPD)

Let $\mathcal{Q}$ be a set of points in $\mathbb{R}^d$. Two sets $P_i, Q_i \subseteq \mathcal{Q}$ of points are s-well separated if they can be enclosed within balls of radius r such that the closest distance between these balls is at least sr. An s-well separated pair decomposition (s-WSPD) of size m for a point set $\mathcal{Q}$ is a set of *s-well-separated* pairs of subsets $\{(P_1, Q_1), \ldots, (P_m, Q_m)\}$, where each $(P_i, Q_i) \subset 2^{\mathcal{Q}} \times 2^{\mathcal{Q}}$, and for any pair of points $p, q \in \mathcal{Q}\, (p \neq q)$ there is a unique index i for which $p \in P_i, q \in Q_i$. See Figure 2. Moreover, for any s-well separated pair (P_i, Q_i), for a sufficiently large separation parameter s, we have approximately equal distances between any two points, where one lies in P_i and the other lies in Q_i. Furthermore, each pair P_i, Q_i has two *representatives* $p_i \in P_i$ and $q_i \in Q_i$, where p_i, q_i gives an approximation for distances between any two points from P_i to Q_i. It has been shown that an s-WSPD of $O(s^d n)$ pairs can be computed in $O(n \log n + s^d n)$ [3].

We start stating our results with a related question: Given is a set D of imprecise points modeled by disjoint unit balls. The question is determining whether there exists an spanner G for an arbitrary realization $\mathcal{Q}$ of D such that for any other realization $\mathcal{Q}'$ of D where $\mathcal{Q}' \neq \mathcal{Q}$, the graph G remains an spanner for $\mathcal{Q}'$ with the same stretch factor. Abam et al. in [1] answered this question positively by introducing a method for computing the WSPD with respect to a separation ratio s' on the center of the balls. They proved that the computed WSPD remains valid for any realization $\mathcal{Q}$ of D, where the separation ratio s of the WSPD on instances is calculated according to $s = \frac{s'-2}{2}$. Then they create a spanner that is valid for any realization.

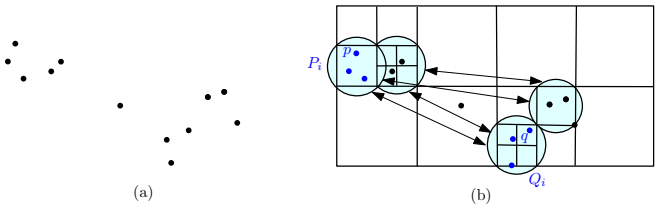

Figure 2: (a) Illustration of a point set and (b) a well-separated pair decomposition of it with 4 pairs (computed from the quadtree [14]).

Let D be a set of n unit balls, and let s' be the separation ratio, for which one make a WSPD on the centers. The following result exist:

Lemma 1 (Lemma 1 [1]) *Let D be a set of disjoint unit balls, and let $\{(P_i, Q_i)|1 \leq i \leq m\}$ be a WSPD for the set $\{c_1, \ldots, c_n\}$ as the centers of the balls in D, with respect to $s' = 2s + 2$. Let $\mathcal{Q} = \{p_1, \ldots, p_n\}$ be a set of points, where $p_j \in D_j$, for $1 \leq j \leq n$. For $1 \leq i \leq m$, let $P'_i = \{p_j | c_j \in P_i\}$ and $Q'_i = \{p_j | c_j \in Q_i\}$. Then $\{(P'_i, Q'_i)|1 \leq i \leq m\}$ is a WSPD for $\mathcal{Q}$ with respect to s.*

2.2 Point Set Diameter Approximation

A $(1 + \epsilon)$-approximation algorithm already exists for approximating the diameter of a point set in $\mathbb{R}^d$ using WSPD [7] (Chapter 3, Lemma 3.14). Let $\mathcal{Q}$ be a set of n points in $\mathbb{R}^d$. For a given $0 \leq \epsilon \leq 1$, the objective is computing a pair $p_u, p_v \in \mathcal{Q}$ such that $\frac{diam(\mathcal{Q})}{1+\epsilon} \leq \|p_u p_v\| \leq diam(\mathcal{Q})$. In the following, we recall the algorithm.

Algorithm: Approximating the Diameter [7]. We first compute an s-WSPD for a point set $\mathcal{Q}$, where $s = 4/\epsilon$. For each WSPD pair (P_i, Q_i), associate a pair of points as representative points $p_u \in P_i, p_v \in Q_i$ and compute the distance between them. See Figure 3. We then remember the maximum distance among all representative points and return it in the end. This would give a $(1 + \epsilon)$-approximation for the diameter of $\mathcal{Q}$ [7]. It is shown that the running time of this algorithm is $O(n \log n + s^d n)$ as the WSPD needs to be computed. Although, the number of candidate pairs realizing the diameter is only $O(s^d n)$.

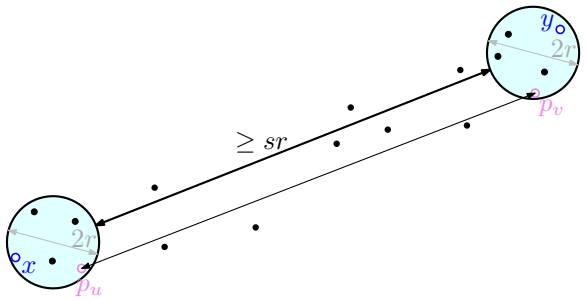

Figure 3: Diameter approximation using WSPD. The points x, y determine the diameter, and p_u, p_v approximate the diameter within a factor $1+\epsilon$.

Our method is in fact a combination of Abam et al. [1] technique in the preprocessing phase for computing a *persistent* WSPD which is computed on the disk centers, and the diameter approximation algorithm [7] in the query phase, using the computed WSPD in the preprocessing step.

3 Computing the Diameter after Preprocessing

Observe that for any set D of n imprecise points in $\mathbb{R}^2$, there is no preprocessing with running time $o(n \log n)$ on D to speed-up answering the diameter queries on D to $o(n \log n)$ time. That is because all such problems simulate the point set case, and it is known that there is a lower bound $\Omega(n \log n)$ for the diameter problem in any algebraic decision tree [12]. In other words, if the preprocessing takes $o(n \log n)$ time, this would result in an $o(n \log n)$ time algorithm for the diameter of a set of points in the plane. As another variant consider the input regions as a set of parallel lines in the plane. If the $2D$ points are sorted in just a single direction, one cannot compute their diameter in less than $\Omega(n \log n)$ time [15]. Because, if D is a set of parallel lines, e.g., along the x-axis, we can only anticipate the x-order of the points (received later), from which the lower bound follows.

For a set of unit disks in $\mathbb{R}^2$, the diameter query problem can be solved in $O(n)$ time after spending $O(n \log n)$ time for preprocessing. Let D be a set of unit disks in the plane. It is known that the Delaunay triangulation of a realization of D, as the query set, can be computed in $O(n)$ time after spending $O(n \log n)$ time for preprocessing. Hence, the convex hull can be extracted in $O(n)$ time. Having the convex hull, the diameter also can be computed in $O(n)$ time, as all the antipodal pairs of a convex polygon can be computed in $O(n)$ time and the diameter is among them [16].

In $\mathbb{R}^d$, we focus on approximation algorithms. An obvious constant factor approximation algorithm for this problem is the smallest enclosing ball (SEB) of a set of points that approximates the diameter of the points within a factor $\frac{\sqrt{3}}{3}$. Consider the configuration at which four points on the boundary of the SEB forms an equilateral triangular-based pyramid, and the side length of each triangular face determines the diameter. If one translates any pair of these points on the boundary of the SEB, to get closer, the diameter enlarges between at least one pair. Hence, a $(\frac{\sqrt{3}}{3}+\epsilon)$-approximation of the diameter of any set of points in $\mathbb{R}^d$ can be computed in $O(dnz/\epsilon^{O(1)})$ time by using the randomized $O(1+\epsilon)$-approximation algorithm in [10] for computing the SEB of a set of points in $\mathbb{R}^d$, at which z is a parameter depending on the input [1]. Next, we discuss a $(1+\epsilon)$-approximation algorithm.

3.1 Preprocessing

In this section, our objective is to preprocess the regions, such that when the exact position of points are given, one can compute and return an approximation of the diameter in $o(n \log n)$ time. To solve the problem, in our algorithm we use the aforementioned technique that returns a $(1+\epsilon)$-approximation of diameter using WSPD on the point set in $O(n/\epsilon^2)$. However, we need to compute the WSPD on the point set according to a specific separation factor $s = \frac{4}{\epsilon}$, but it takes $O(n \log n + s^2 n)$ time and makes the algorithm useless. Therefore, in the case where the input is a set of disks, we use Abam et al. [1] technique for computing a WSPD on the center points of the disks with the separation parameter $s' = 2s + 2$, which has been proved that would be valid for any realization according to separation factor s. Hence, we do not need to compute the WSPD on each instance, and the WSPD is computed only once in the preprocessing phase.

Lemma 2 *Let $\{(A_i, B_i) | 1 \leq i \leq m\}$ be a WSPD on the set $\{c_1, \ldots, c_n\}$ of given disjoint unit disks with respect to $s' = \frac{8}{\epsilon} + 2$. Let $\mathcal{Q} = \{p_1, \ldots, p_n\}$ be a set of points, where $p_j \in D_j$, for $1 \leq j \leq n$. For $1 \leq i \leq m$, let $A'_i = \{p_j | c_j \in A_i\}$ and $B'_i = \{p_j | c_j \in B_i\}$. Then $\{(A'_i, B'_i) | 1 \leq i \leq m\}$ is a WSPD for $\mathcal{Q}$ with respect to $s = \frac{4}{\epsilon}$.*

Proof. According to Lemma 1 the $\{(A'_i, B'_i) | 1 \leq i \leq m\}$ would be a valid WSPD for any instance with respect to separate factor $s = \frac{s'-2}{2}$. We assumed the separate factor of the WSPD on the center points is $s' = \frac{8}{\epsilon} + 2$, so we have: $s = \frac{s'-2}{2} = \frac{(\frac{8}{\epsilon}+2)-2}{2} = \frac{\frac{8}{\epsilon}}{2} = \frac{4}{\epsilon}$. □

3.2 Query Phase

Now, when we are given a realization of the balls, we wish to compute a $(1+\epsilon)$-approximation of the diameter in $O(n/\epsilon^d)$ time. It follows from Lemma 2 that we can

[1] We note this is the best-known algorithm for computing the SEB, that has a linear dependency on d.

do this by having a WSPD on the center points with respect to separation factor $s = \frac{4}{\epsilon}$. In addition, our WSPD is valid for any other realization.

Theorem 3 *For any given set $D = \{D_1, \ldots, D_n\}$ of n imprecise points modeled as the same size balls which are pairwise disjoint, a $(1+\epsilon)$-approximation of the diameter of a realization $\mathcal{Q}$ of D can be computed in $O(\frac{n}{\epsilon^d})$ time, after $O(n \log n + \frac{n}{\epsilon^d})$ preprocessing time.*

Proof. Let $s = \frac{4}{\epsilon}$ and $s' = 2s + 2 = \frac{8}{\epsilon} + 2$ and $\mathcal{Q} = \{p_1, \ldots, p_n\}$ be the set of precise points. Let $\{(A_i, B_i)\}_{i=1,\ldots,m}$ be an s'-WSPD for the center points, of size $m = O(s'^2 n)$, and let $A'_i = \{p_j | c_j \in A_i\}$, $B'_i = \{p_j | c_j \in B_i\}$. It follows from Lemma 1 that $\{(A'_i, B'_i) | 1 \leq i \leq m\}$ is a WSPD for $\mathcal{Q}$ with respect to separation parameter $s = \frac{4}{\epsilon}$.

Then, we associate one point to each set as the representative point, let $p_a \in A'_i$ and $p_b \in B'_i$ be the representative points of the sets A'_i and B'_i respectively. From the presented approximation algorithm for the diameter in [7], by calculating the distance between representative points of each pair and computing the maximum distance among all, we have a $(1+\epsilon)$-approximation of the diameter of any realization in $O(s'^d n)$ time. □

4 Discussion

The main open question is finding an algorithm for the general version, as our approach cannot be extended to overlapping disks or disks of arbitrary size. We note that since the WSPD gives the nearest neighbour pairs in a set of points, our results give also an $O(1)$-approximation algorithm for the nearest neighbour query for a realization of D in high dimensions.

Acknowledgement The authors would like to thank an anonymous reviewer for pointing out the efficient exact algorithm for the 2-d case. V.K is Supported by the Czech Science Foundation, grant number GJ19-06792Y, and by institutional support RVO: 67985807.

References

[1] M. A. Abam, P. Carmi, M. Farshi, and M. Smid. On the power of the semi-separated pair decomposition. *Comput. Geom.*, 46(6):631–639, 2013.

[2] K. Buchin, M. Löffler, P. Morin, and W. Mulzer. Delaunay triangulation of imprecise points simplified and extended. In *WADS*, pages 131–143. Springer, 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. *J. ACM*, 42(1):67–90, 1995.

[4] E. Ezra and W. Mulzer. Convex hull of points lying on lines in $o(n \log n)$ time after preprocessing. *Comput. Geom.*, 46(4):417–434, 2013.

[5] D. V. Finocchiaro and M. Pellegrini. On computing the diameter of a point set in high dimensional euclidean space. *Theoretical Computer Science*, 287(2):501–514, 2002.

[6] X. Guo, X. Yang, D. Chen, and C. Chen. Diameter-aware extreme group queries. *IEEE Access*, 6:58687–58701, 2018.

[7] S. Har-Peled. *Geometric approximation algorithms*. Number 173. American Mathematical Soc., 2011.

[8] J. Hong, Z. Wang, and W. Niu. A simple approximation algorithm for the diameter of a set of points in an euclidean plane. *Plos one*, 14(2):e0211201, 2019.

[9] V. Keikha, A. Mohades, and M. D. Monfared. On the triangulation of non-fat imprecise points. In *CCCG*, pages 114–121, 2016.

[10] A. Krivošija. Probabilistic smallest enclosing ball in high dimensions. *Technical report for Collaborative Research Center SFB 876 Providing Information by Resource-Constrained Data Analysis*, page 13, 2019.

[11] M. Löffler and J. Snoeyink. Delaunay triangulation of imprecise points in linear time after preprocessing. *Comput. Geom.*, 43(3):234–242, 2010.

[12] G. Malandain and J.-D. Boissonnat. Computing the diameter of a point set. *Internat. J. Comput. Geom. Appl.*, 12(06):489–509, 2002.

[13] F. P. Preparata and M. I. Shamos. Computational Geometry: an Introduction. pages 95–149. Springer, 1985.

[14] H. Samet. The quadtree and related hierarchical data structures. *ACM Computing Surveys (CSUR)*, 16(2):187–260, 1984.

[15] R. Seidel. A method for proving lower bounds for certain geometric problems. In *Machine Intelligence and Pattern Recognition*, volume 2, pages 319–334. Elsevier, 1985.

[16] M. I. Shamos. Computational geometry. Ph.D. Thesis. 1978.

[17] I. van der Hoog, I. Kostitsyna, M. Löffler, and B. Speckmann. Preprocessing ambiguous imprecise points. *arXiv preprint arXiv:1903.08280*, 2019.

[18] M. Van Kreveld, M. Löffler, and J. S. Mitchell. Preprocessing imprecise points and splitting triangulations. *SIAM J. Comput.*, 39(7):2990–3000, 2010.