Better Approximation Algorithm for Point-set Diameter

Mahdi Imanparast, and Seyed Naser Hashemi

***Abstract*—The problem of computing the diameter of a set of points on a dataset is one of the fundamental issues in computational geometry, which is used in many applications. Since precisely computing the diameter has required quadratic time, some approximation approaches to this problem is considered. In this paper, we propose a new $(1 + O(\varepsilon))$-approximation algorithm with $O(n + 1/\varepsilon^{(d-1)/2})$ running time for computing the diameter of a set of n points in the d-dimensional Euclidean space for a fixed dimension d, where $0 < \varepsilon \leqslant 1$. This result provides some improvements in the running time of this problem in comparison with previous algorithms.**

***Index Terms*—Computational geometry, approximation algorithms, diameter, point-set, fixed dimensions.**

I. Introduction

COMPUTING the diameter of a point-set has a long history and is used in various fields, such as database, data mining, clustering, vision [1], and interconnection networks [2]. For example, in a database containing a set of images of the same size, it is possible to consider each image with d pixels as a point in the d-dimensional space for a fixed dimension d. In this case, computing the diameter in this dataset means finding two most different images. For a finite set of n points, the purpose of computing the diameter of a point-set is to find two points with maximum distance in the point-set. A trivial brute-force solution for the diameter problem is to calculate the distance between each pair of points and then select the maximum distance which takes $O(dn^2)$ time. By reducing from the set disjointness problem, it can be shown that computing the diameter of n points in $\mathbb{R}^d$ requires $\Omega(n \log n)$ operations in the algebraic computation-tree model [3].

There are well-known solutions to this problem in two and three dimensions. This problem can be solved at the optimal time $O(n \log n)$ in the plane, but in three dimensions, it is a bit more difficult. Several attempts have been made to solve the diameter problem at the optimal time in three dimensions, and finally, Ramos [4] presented an optimal deterministic $O(n \log n)$-time algorithm in $\mathbb{R}^3$. In the case of higher dimensions (for $d > 3$), the brute-force algorithm needs $O(dn^2)$ time, which is too slow for large-scale datasets that appear in the fields. Hence, it seems to be necessary to use approximation algorithms for computing the diameter in these dimensions. A 2-approximation algorithm with $O(dn)$ time in d-dimensional space can easily be obtained by choosing a point from the point-set, and then finding the farthest point of it by brute-force manner. The first non-trivial approximation algorithm for the diameter is presented by Egecioglu and Kalantari [5], which calculates a $\sqrt{3}$-approximation with $O(dn)$ time. They also provided an iterative algorithm with $O(tdn)$ time and an approximation factor $\sqrt{5 - 2\sqrt{3}}$, where t is the number of iterations of the algorithm. Agarwal et al. [6] proposed the first $(1+\varepsilon)$-approximation algorithm for computing the diameter in $\mathbb{R}^d$ with $O(n/\varepsilon^{(d-1)/2})$ time by projection to directions. Subsequently, several approximation algorithms have been proposed in [7], [8], [9], [10], [11], each of which improved the required time to solve this problem. Recently, Chan [12] has presented an approximation algorithm based on the Chebyshev polynomials with the running time $O((n/\sqrt{\varepsilon}+1/\varepsilon^{\frac{d}{2}+1})(\log \frac{1}{\varepsilon})^{O(1)})$, for computing a $(1+O(\varepsilon))$-approximation of the diameter, and Arya et al. [13] have also shown that by applying an efficient decomposition of a convex body using a hierarchy of Macbeath regions, it is possible to have an approximation for the diameter of a set of points in time $O(n \log \frac{1}{\varepsilon} + 1/\varepsilon^{\frac{(d-1)}{2}+\alpha})$, where α is a small positive constant. Table I provides a summary overview of non-constant approximation algorithms for computing the diameter of a set of points.

TABLE I
A SUMMARY OF THE COMPLEXITY OF SOME UTILIZABLE NON-CONSTANT APPROXIMATION ALGORITHMS FOR COMPUTING THE DIAMETER OF A POINT-SET IN d-DIMENSIONAL EUCLIDEAN SPACE. OUR RESULT IS PRESENTED IN THE LAST ROW.

Ref.	Approx. Factor	Running Time	Year
[6]	$1+\varepsilon$	$O(\frac{n}{\varepsilon^{(d-1)/2}})$	1992
[7]	$1+\varepsilon$	$O(n+1/\varepsilon^{2(d-1)})$	2001
[8]	$1+\varepsilon$	$O((n+1/\varepsilon^{2d})\log\frac{1}{\varepsilon})$	2001
[9]	$1+\varepsilon$	$O(n+1/\varepsilon^{\frac{3(d-1)}{2}})$	2002
[11]	$1+\varepsilon$	$O(n+1/\varepsilon^{d-1})$	2018
[9]	$1+O(\varepsilon)$	$O(n+1/\varepsilon^{d-\frac{1}{2}})$	2002
[10]	$1+O(\varepsilon)$	$O(n+1/\varepsilon^{d-\frac{3}{2}})$	2006
[11]	$1+O(\varepsilon)$	$O(n+1/\varepsilon^{\frac{2d}{3}-\frac{1}{3}})$	2018
[12]	$1+O(\varepsilon)$	$O((n/\sqrt{\varepsilon}+1/\varepsilon^{\frac{d}{2}+1})(\log\frac{1}{\varepsilon})^{O(1)})$	2017
[13]	$1+O(\varepsilon)$	$O(n\log\frac{1}{\varepsilon}+1/\varepsilon^{\frac{(d-1)}{2}+\alpha})$	2017
Ours	$1+O(\varepsilon)$	$O(n+1/\varepsilon^{\frac{(d-1)}{2}})$	2019

A. Our result

In this paper, we propose a simple $(1 + O(\varepsilon))$-approximation algorithm for computing the diameter of a set $\mathcal{S}$ of n points in $\mathbb{R}^d$ with $O(n + 1/\varepsilon^{(d-1)/2})$ time and $O(n)$ space, where $0 < \varepsilon \leqslant 1$. As stated in Table I, two new results have been recently presented for the diameter problem in [12] and [13]. It should be noted that our algorithm is completely different in terms of computational technique.

Manuscript received April 28, 2019; revised July 30, 2019.

M. Imanparast is with the Department of Mathematics and Computer Science, Amirkabir University of Technology, Tehran, Iran (e-mail: m.imanparast@aut.ac.ir)

S. N. Hashemi is with the Department of Mathematics and Computer Science, Amirkabir University of Technology, Tehran, Iran (e-mail: nhashemi@aut.ac.ir)

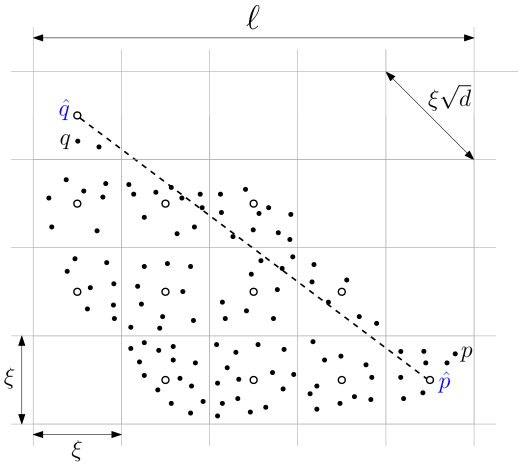


Fig. 1. Rounding the points of the set $\mathcal{S}$ to their central-cell points in a ξ-grid. Two points $\hat{p}$ and $\hat{q}$ are the corresponding central-cell points for two points p and q which are the diametrical pair $(p, q) \in \mathcal{S}$.

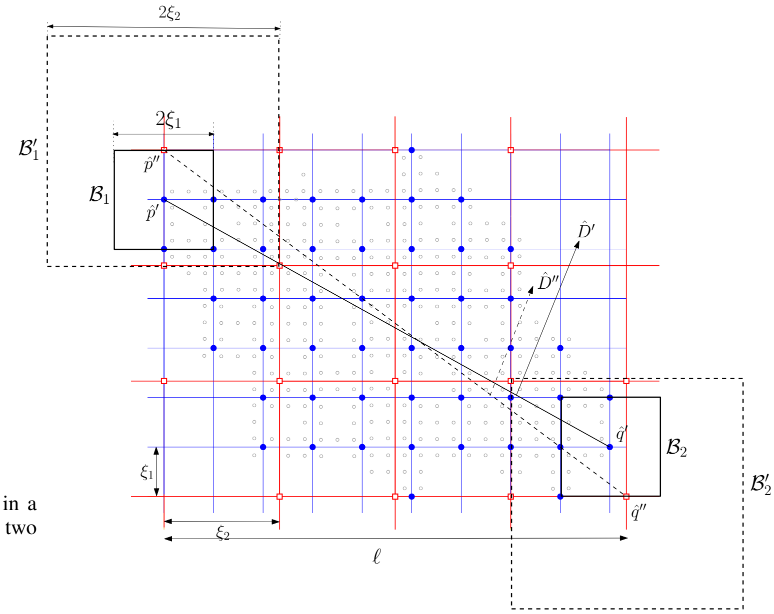


Fig. 2. An example of rounding the points of the rounded point-set $\hat{\mathcal{S}}$ to a ξ_1-grid, and then rounding the points of the set $\hat{\mathcal{S}}'$ to a ξ_2-grid.

The polynomial technique provided by Chan [12] is based on using Chebyshev polynomials and discrete upper envelope subroutine [10], and the method presented by Arya et al. [13] requires the use of complex data structures to approximately answer queries for polytope membership, directional width, and nearest-neighbor. While our algorithm in comparison with these algorithms is simpler in terms of understanding and data structure. The rest of this paper is organized as follows. The proposed algorithm is introduced in section II. Analysis of the algorithm is described in section III. And we conclude our paper in section VI.

II. The Proposed Algorithm

In this section, we describe our new approximation algorithm to compute the diameter of a set $\mathcal{S}$ of n points in $\mathbb{R}^d$. The main idea of our algorithm is based on rounding the points to the grids. We first round the points to their central-cell points on a grid and then in two phases to their nearest grid-points for different grid cell sizes. The most important advantage of these sequential rounding by increasing the grid cell size is that it will not only reduce the number of points we examine, but also it will reduce the search domain for computing the diameter at each rounding phase. In fact, by computing the diameter in the set of rounded points in the third phases, we divide the problem into a set of subproblems that helps us in solving subproblems in the previous rounded point-sets and ultimately leads to finding a $(1 + O(\varepsilon))$-approximation of the true diameter.

We first find the extreme points in each coordinate and compute the axis-parallel bounding box of $\mathcal{S}$, which is denoted by $B(\mathcal{S})$. We use the largest length side ℓ of $B(\mathcal{S})$ to impose grids on the point-set. In first rounding phase, we decompose $B(\mathcal{S})$ to a grid of regular hypercubes with side length ξ, where $\xi = \varepsilon\ell/2\sqrt{d}$. We call each hypercube a cell. See Fig. 1. On the other hand, for the diametrical pair points p and q and their corresponding approximate diametrical pair points $\hat{p}$ and $\hat{q}$, the true diameter and the approximate diameter will be $D = ||p-q||$ and $\hat{D} = ||\hat{p}-\hat{q}||$. This means that the first rounding provides a $(1+\varepsilon)$-approximation for the diametrical pair $(p, q) \in \mathcal{S}$ (for more details see [11]). Then, we repeat the same rounding process twice for grids with side length $\xi_1 \leftarrow \sqrt{\varepsilon}\ell/2\sqrt{d}$ and $\xi_2 \leftarrow \varepsilon^{\frac{1}{4}}\ell/2\sqrt{d}$. We assume that $\mathcal{B}_\delta(p)$ denotes a hypercube with side length δ and central-point p. We also use notation $Diam(\mathcal{B}_1, \mathcal{B}_2)$ for the process of computing the diameter of the point-set $\mathcal{B}_1 \cup \mathcal{B}_2$. In the following, we present our algorithm in more details in Algorithm 1.

Algorithm 1: Approximate Diameter $(\mathcal{S}, \varepsilon)$

Input: a set $\mathcal{S}$ of n points in $\mathbb{R}^d$ and an error parameter ε.

Output: approximate diameter $\hat{D}$.

Compute the axis-parallel bounding box $B(\mathcal{S})$ for the point-set $\mathcal{S}$.
$\ell \leftarrow$ Find the length of the largest side in $B(\mathcal{S})$.
Set $\xi \leftarrow \varepsilon\ell/2\sqrt{d}$, $\xi_1 \leftarrow \sqrt{\varepsilon}\ell/2\sqrt{d}$ and $\xi_2 \leftarrow \varepsilon^{\frac{1}{4}}\ell/2\sqrt{d}$.
$\hat{\mathcal{S}} \leftarrow$ Round each point of $\mathcal{S}$ to its central-cell point in a ξ-grid.
$\hat{\mathcal{S}}' \leftarrow$ Round each point of $\hat{\mathcal{S}}$ to its nearest grid-point in a ξ_1-grid.
$\hat{\mathcal{S}}'' \leftarrow$ Round each point of $\hat{\mathcal{S}}'$ to its nearest grid-point in a ξ_2-grid.
$\hat{D}'' \leftarrow$ Compute the diameter of the point-set $\hat{\mathcal{S}}''$ by brute-force manner.
Compute a list of the diametrical pairs $(\hat{p}'', \hat{q}'')$, such that $\hat{D}'' = ||\hat{p}'' - \hat{q}''||$.
Corresponding to each diametrical pair $(\hat{p}'', \hat{q}'') \in \hat{\mathcal{S}}''$, find points of $\hat{\mathcal{S}}'$ which are inside two hypercubes $\mathcal{B}_{2\xi_2}(\hat{p}'')$ and $\mathcal{B}_{2\xi_2}(\hat{q}'')$, and store them in two point-sets $\mathcal{B}'_1$ and $\mathcal{B}'_2$.
$\hat{D}' \leftarrow$ For each pair $(\mathcal{B}'_1, \mathcal{B}'_2)$ corresponding to each diametrical pair $(\hat{p}'', \hat{q}'') \in \hat{\mathcal{S}}''$, compute $Diam(\mathcal{B}'_1, \mathcal{B}'_2)$, by brute-force manner and return the maximum value between them.
For the computed diameter $\hat{D}'$, compute a list of the diametrical pairs $(\hat{p}', \hat{q}')$, such that $\hat{D}' = ||\hat{p}' - \hat{q}'||$.
Corresponding to each diametrical pair $(\hat{p}', \hat{q}') \in \hat{\mathcal{S}}'$, find points of $\hat{\mathcal{S}}$ which are inside two hypercubes $\mathcal{B}_{2\xi_1}(\hat{p}')$ and $\mathcal{B}_{2\xi_1}(\hat{q}')$, and store them in two point-sets $\mathcal{B}_1$ and $\mathcal{B}_2$.
$\hat{D} \leftarrow$ Compute $Diam(\mathcal{B}_1, \mathcal{B}_2)$, corresponding to each diametrical pair $(\hat{p}', \hat{q}') \in \hat{\mathcal{S}}'$ by Chan's [9] recursive approach and return the maximum value between them.
Output $\hat{D}$.

In Fig. 2, we illustrate an example of rounding the points of the rounding point-set $\hat{\mathcal{S}}$ to a ξ_1-grid, and then rounding points of the new point-set $\hat{\mathcal{S}}'$ to a ξ_2-grid. Points of the rounded point-set $\hat{\mathcal{S}}$ are shown by circle points ($\circ$) and their corresponding nearest grid-points in point-set $\hat{\mathcal{S}}'$ are shown by blue points ($\bullet$). Moreover, the points of the point-set $\hat{\mathcal{S}}''$ are shown by small squares ($\square$).

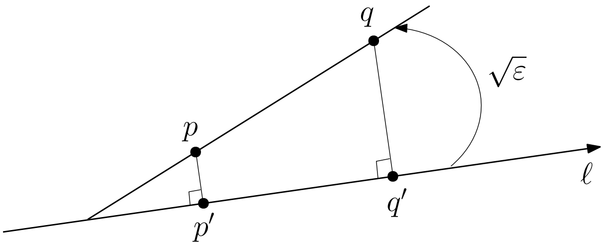


Fig. 3. Projecting two points p and q on direction ℓ.

The searching domain for finding the diameter of the point-set $\hat{\mathcal{S}}$ is reduced into two point-sets $\mathcal{B}_1$ and $\mathcal{B}_2$. It should be noted that we use a hypercube $\mathcal{B}_{2\xi_1}(\hat{p}')$ of side length $2\xi_1$ to make sure that we do not lose any points of the rounded point-set $\hat{\mathcal{S}}$ around a grid-point $\hat{p}' \in \hat{\mathcal{S}}'$ (see Lemma 1). As can be seen in Fig. 2, starting from the point-set $\hat{\mathcal{S}}''$, and finding the points of the previous set $\hat{\mathcal{S}}'$ contained within two hypercubes $\mathcal{B}_1'$ and $\mathcal{B}_2'$, and repeating the same process for the point-set $\hat{\mathcal{S}}'$, finally an approximate diameter for the point-set $\hat{\mathcal{S}}$ is computed.

What needs to be further explained is the last step of the algorithm in which Chan's [9] recursive approach is used to compute the approximate diameter. As previously mentioned, Agarwal et al. [6] introduced an approximation algorithm that in $O(n/\varepsilon^{(d-1)/2})$ time could find a $(1+\varepsilon)$-approximation for the diameter of a set of n points. Their result is based on the following simple observation. Let denote the projection of a point-set $\mathcal{S}$ on the line l by $\mathcal{S}_l$. See Fig. 3. Then, if two points p and q are the diametrical pair of the point-set $\mathcal{S}$, and p' and q' be their projection on the line l, such that the angle between pq and $p'q'$ be at most $\sqrt{\varepsilon}$, we have

$$||p'-q'|| \leqslant ||p-q|| \leqslant (1+\varepsilon)||p'-q'||. \quad (1)$$

This means that $||p'-q'||$ is a $(1+\varepsilon)$-approximation of $||p-q||$.

Now, we can find $O(1/\varepsilon^{(d-1)/2})$ numbers of directions in $\mathbb{R}^d$, for example by constructing a uniform grid on a unit sphere, such that for each vector $x \in \mathbb{R}^d$, there is a direction that the angle between this direction and vector x be at most $\sqrt{\varepsilon}$ [6]. So, it is sufficient to project the point-set $\mathcal{S}$ on each of these directions and compute their diameter. Then, we can consider the maximum value of the computed diameters as a $(1+\varepsilon)$-approximate diameter for the point-set $\mathcal{S}$.

Chan [9] observed that instead of running their method on n points, the points can be first rounded to a grid, in which case, the number of points would be reduced from n to $m = O(1/\varepsilon^{(d-1)})$. Then, by applying Agarwal et al. [6] method on this rounded point-set, we need $O((1/\varepsilon^{d-1})/\varepsilon^{(d-1)/2}) = O(1/\varepsilon^{3(d-1)/2})$ time to compute the maximum diameter over all $O(1/\varepsilon^{(d-1)/2})$ directions. Taking into account $O(n)$ time we spend for rounding to a grid, this new approach computes a $(1+\varepsilon)$-approximation for the diameter of a set of n points in $O(n + 1/\varepsilon^{3(d-1)/2})$ time.

Chan [10] also observed that in Agarwal et al. [6] method, instead of projecting rounded points on $O(1/\varepsilon^{(d-1)/2})$ d-dimensional unit vectors, one can project $m = O(1/\varepsilon^{(d-1)})$ rounded points on a set of $O(1/\sqrt{\varepsilon})$ 2-dimensional unit vectors to reduce the problem to $O(1/\sqrt{\varepsilon})$ numbers of $(d-1)$-dimensional subproblems which can be solved recursively. Let us denote the required time for computing the diameter of m points in a d-dimensional space with $t_d(m)$, then for a rounded point-set on a grid with $m = O(1/\varepsilon^{d-1})$ points, this recursive approach breaks the problem into $O(1/\sqrt{\varepsilon})$ subproblems in a $(d-1)$ dimensional space. Hence, we have a recurrence

$$t_d(m) = O(m + 1/\sqrt{\varepsilon}\, t_{d-1}(O(1/\varepsilon^{d-1}))). \quad (2)$$

By assuming $E = 1/\varepsilon$, we can rewrite the recurrence as:

$$t_d(m) = O(m + E^{\frac{1}{2}} t_{d-1}(O(E^{d-1}))). \quad (3)$$

This can be solved to $t_d(m) = O(m + E^{d-\frac{1}{2}})$. In this case, $m = O(1/\varepsilon^{d-1})$, so, this recursion takes $O(1/\varepsilon^{d-\frac{1}{2}})$ time. Taking into account $O(n)$ time, we spent for rounding to a grid at the first, Chan's recursive approach computes a $(1+O(\varepsilon))$-approximation for the diameter of a set of n points in $O(n + 1/\varepsilon^{d-\frac{1}{2}})$ time [9]. For more details on the Agarwal et al. and Chan's recursive methods, the readers are referred to [9] and [11].

III. Analysis

In this section, we analyze the proposed algorithm. For this purpose, we first show in the following lemma that it is sufficient to consider a hypercube of side length of twice of the grid cell size at diametrical pairs in rounded point-sets.

Lemma 1. *Considering the points inside hypercubes with side length of* $2\xi_1$ *around the rounded points* $\hat{p}_1$ *and* $\hat{q}_1$ *are sufficient to find an approximate diameter.*

Proof: Consider a special case of Fig. 4, in which the defining points of the true diameter are located at the upper boundary of the hypercube $\mathcal{B}_1$ and the lower boundary of the hypercube $\mathcal{B}_2$. In this case, the true diameter between two points v and w is $D = Diam(S) = ||v-w||$. On the other hand, $\hat{D}_1 = Diam(\hat{S}') = ||\hat{p_1}' - \hat{q_1}'||$. For the approximate diameter, we will have two points a and d, which means that $\hat{D} = Diam(\hat{S}) = Diam(\mathcal{B}_1, \mathcal{B}_2) = ||a-d||$. Now, suppose the size of the sides of the hypercubes $\mathcal{B}_1$ and $\mathcal{B}_2$ be less than $2\xi_1$. In this case, two points b and c are considered for the approximate diameter. Then, $\hat{D}' = ||b-c||$ is the approximate diameter in this case.

On the other hand, according to Fig. 1, for the true diameter and the approximate diameter, we have:

$$D - \xi\sqrt{d} \leqslant \hat{D} \leqslant D + \xi\sqrt{d}. \quad (4)$$

Given that $\xi = \varepsilon\ell/2\sqrt{d}$ and $\ell \leqslant D$, we have:

$$D \leqslant \hat{D} + \varepsilon\ell/2 \leqslant D + \varepsilon\ell. \quad (5)$$

If we assume that $\tilde{D} = \hat{D} + \varepsilon\ell/2$, we have:

$$D \leqslant \tilde{D} \leqslant (1+\varepsilon)D. \quad (6)$$

Therefore, the final approximate diameter is $\tilde{D} = \hat{D} + \varepsilon\ell/2$ (for more details see [11]). So, suppose $\varepsilon = 1/36$. Since $\ell = 102$. In this case, we will have $\xi = 1$, $\xi_1 = 6$, and we have:

$$D = ||v-w|| = 102.313, \quad (7)$$

and

$$\tilde{D} = \hat{D} + \varepsilon\ell/2,$$
$$\tilde{D} = ||a-d|| + \varepsilon\ell/2 = 101.815 + \frac{1}{36}\cdot\frac{102}{2} = 103.232, \quad (8)$$

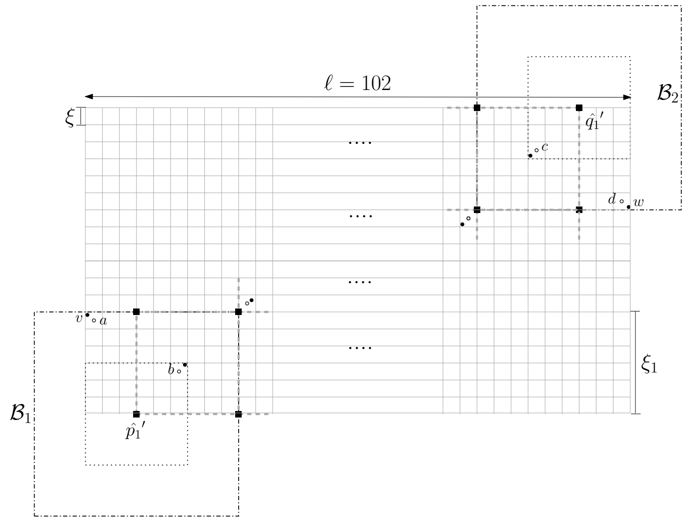


Fig. 4. A case in proof of the Lemma 1.

and

$$\tilde{D}' = \hat{D}' + \varepsilon\ell/2,$$

$$\tilde{D}' = ||b - c|| + \varepsilon\ell/2 = 91.924 + \frac{1}{36} \cdot \frac{102}{2} = 93.341, \quad (9)$$

Finally, we can see that the approximate diameter $\tilde{D}$ satisfy the relation $D \leqslant \tilde{D} \leqslant (1+\varepsilon)D$, and for the approximate diameter $\tilde{D}'$, the relation $D \leqslant \tilde{D}' \leqslant (1+\varepsilon)D$ is not established. So, this indicates that the considering points inside a hypercube of side length $2\xi_1$ is a good choice, and this is sufficient to select the points. ■

Now, we can prove the following theorem about the computed diameter in Algorithm 1.

Theorem 2. *Algorithm 1 computes a* $(1+O(\varepsilon))$*-approximate diameter for a set* $\mathcal{S}$ *of* n *points in* $\mathbb{R}^d$ *in* $O(n+1/\varepsilon^{(d-1)/2})$ *time and* $O(n)$ *space, where* $0 < \varepsilon \leqslant 1$.

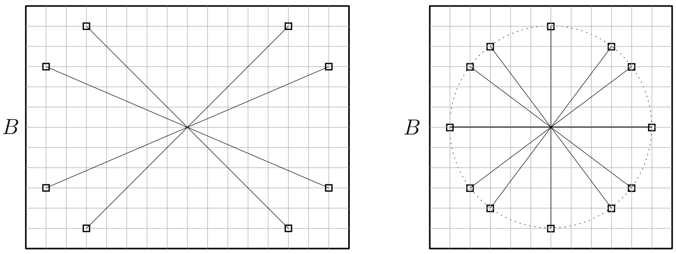
Proof: Finding the extreme points in all coordinates and finding the largest side of $B(\mathcal{S})$ can be done in $O(dn)$ time. The rounding step takes $O(d)$ time for each point, and for all of them takes $O(dn)$ time. But for computing the diameter over the rounded point-set $\hat{\mathcal{S}}''$, in line 7 of the Algorithm 1, we need to know the number of points in the set $\hat{\mathcal{S}}''$. We know that the largest side of the bounding box $B(\mathcal{S})$ has length ℓ and the side length of each cell in the ξ_2-grid is $\xi_2 = \varepsilon^{\frac{1}{4}}\ell/2\sqrt{d}$. On the other hand, the volume of a hypercube of side length L in d-dimensional space is L^d. Since, corresponding to each point in the point-set $\hat{\mathcal{S}}''$, we can take a hypercube of side length ξ_2. Therefore, in order to count the maximum number of points inside the point-set $\hat{\mathcal{S}}''$, it is sufficient to calculate the number of hypercubes of side length ξ_2 in a hypercube (bounding box) with side length $\ell + \xi_2$. See Fig. 2. This means that the number of grid-points in an imposed ξ_2-grid to the bounding box $B(\mathcal{S})$ is at most

$$\frac{(\ell+\xi_2)^d}{(\xi_2)^d} = \left(\frac{\ell}{\varepsilon^{\frac{1}{4}}\ell/2\sqrt{d}}+1\right)^d = \left(\frac{2\sqrt{d}}{\varepsilon^{\frac{1}{4}}}+1\right)^d = O\left(\frac{(2\sqrt{d})^d}{\varepsilon^{\frac{d}{4}}}\right). \quad (10)$$

So, the number of points in $\hat{\mathcal{S}}''$ is at most $O((2\sqrt{d})^d/\varepsilon^{\frac{d}{4}})$. This can be reduced to $O((2\sqrt{d})^d/\varepsilon^{\frac{d}{4}-\frac{1}{4}})$ by discarding some internal points which do not have any potential to be the diametrical pairs in rounded point-set $\hat{\mathcal{S}}''$. This can be done by considering all the points in the ξ_2-grid, which are same in their $(d-1)$ coordinates and keep only highest and lowest. Hence, by the brute-force quadratic algorithm, we need $O(((2\sqrt{d})^d/\varepsilon^{\frac{d}{4}-\frac{1}{4}})^2) = O((2\sqrt{d})^{2d}/\varepsilon^{\frac{d}{2}-\frac{1}{2}})$ time for computing all distances between points of the point-set $\hat{\mathcal{S}}''$, and its diametrical pair list. Then, for a diametrical pair $(\hat{p}'', \hat{q}'')$ in point-set $\hat{\mathcal{S}}''$, we compute two sets $\mathcal{B}_1'$ and $\mathcal{B}_2'$ in $O(dn)$ time. They include points of $\hat{\mathcal{S}}'$ which are inside two hypercubes $\mathcal{B}_{2\xi_2}(\hat{p}'')$ and $\mathcal{B}_{2\xi_2}(\hat{q}'')$, respectively. In addition, for computing the diameter of the point-set $\mathcal{B}_1' \cup \mathcal{B}_2'$, we need to know the number of points in each of two point-sets $\mathcal{B}_1'$ and $\mathcal{B}_2'$. On the other hand, the number of points in two sets $\mathcal{B}_1'$ or $\mathcal{B}_2'$ is equivalent to the number of hypercubes of size $\mathcal{B}_{\xi_1}$ that can be in a hypercube of size $\mathcal{B}_{2\xi_2}$, which is at most

$$\frac{Vol(\mathcal{B}_{2\xi_2})}{Vol(\mathcal{B}_{\xi_1})} = \frac{(2\varepsilon^{\frac{1}{4}}\ell/2\sqrt{d})^d}{(\sqrt{\varepsilon}\ell/2\sqrt{d})^d} = \frac{(2\varepsilon^{\frac{1}{4}})^d}{(\varepsilon^{\frac{1}{2}})^d} = \frac{(2)^d}{\varepsilon^{\frac{d}{4}}}. \quad (11)$$

This means that the number of points in two point-sets $\mathcal{B}_1'$ and $\mathcal{B}_2'$ is at most $O((2)^d/\varepsilon^{\frac{d}{4}})$, which can be also reduced to $O(2^d/\varepsilon^{\frac{d}{4}-\frac{1}{4}})$. Hence, for computing $Diam(\mathcal{B}_1', \mathcal{B}_2')$, we need $O(((2)^d/\varepsilon^{\frac{d}{4}-\frac{1}{4}})^2) = O((2)^{2d}/\varepsilon^{\frac{d}{2}-\frac{1}{2}})$ time by quadratic brute-force manner. On the other hand, we might have more than one diametrical pair $(\mathcal{B}_1', \mathcal{B}_2')$. Since the point-set $\hat{\mathcal{S}}''$ is a set of grid-points on a regular grid, so we could have in the worst-case $O(2^d)$ different diametrical pairs $(\mathcal{B}_1', \mathcal{B}_2')$ in the point-set $\hat{\mathcal{S}}''$. Two examples of the diametrical pairs on regular girds in the two-dimensional space are shown in Fig. 4. Therefore, this step takes at most $O(2^d \cdot (2)^{2d}/\varepsilon^{\frac{d}{2}-\frac{1}{2}}) = O((2\sqrt{2})^{2d}/\varepsilon^{\frac{d}{2}-\frac{1}{2}})$ time.



(a) (b)

Fig. 5. The maximum number of diametrical pairs on a regular grid in two dimensional space. The squares (□) denote the rounded points on the grid.

In the next step, for each diametrical pair $(\hat{p}', \hat{q}') \in \hat{\mathcal{S}}'$, we compute two sets $\mathcal{B}_1$ and $\mathcal{B}_2$ which include points of set $\hat{\mathcal{S}}$ which are inside two hypercubes $\mathcal{B}_{2\xi 1}(\hat{p}')$ and $\mathcal{B}_{2\xi 1}(\hat{q}')$, respectively. Moreover, the number of points in two sets $\mathcal{B}_1$ or $\mathcal{B}_2$ is at most

$$\frac{Vol(\mathcal{B}_{2\xi_1})}{Vol(\mathcal{B}_{\xi})} = \frac{(2\sqrt{\varepsilon}\ell/2\sqrt{d})^d}{(\varepsilon\ell/2\sqrt{d})^d} = \frac{(2\varepsilon^{\frac{1}{2}})^d}{\varepsilon^d} = \frac{(2)^d}{\varepsilon^{\frac{d}{2}}}. \quad (12)$$

This can be reduced to $O((2)^d/\varepsilon^{\frac{d}{2}-\frac{1}{2}})$, by keeping only highest and lowest points which are same in their $(d-1)$ coordinates. Now, for computing $Diam(\mathcal{B}_1, \mathcal{B}_2)$, we use Chan's [9] recursive approach instead of using the quadratic brute-force algorithm on the point-set $\mathcal{B}_1 \cup \mathcal{B}_2$. On the other hand, computing the diameter of a set of $O(1/\varepsilon^{\frac{d}{2}-\frac{1}{2}})$ points using Chan's recursive approach takes the following recurrence based on the relation (2): $t_d(m) = O(m +$

$1/\sqrt{\varepsilon}t_{d-1}(O(1/\varepsilon^{\frac{d}{2}-\frac{1}{2}})))$. By assuming $E = 1/\varepsilon$, we can rewrite the recurrence as:

$$t_d(m) = O(m + E^{\frac{1}{2}}t_{d-1}(O(E^{\frac{d}{2}-\frac{1}{2}}))). \quad (13)$$

This can be solved to $t_d(m) = O(m + E^{\frac{d}{2}})$. In this case, $m = O(E^{\frac{d}{2}-\frac{1}{2}})$, so, this recursion takes $O(E^{\frac{d}{2}} + E^{\frac{d}{2}-\frac{1}{2}}) = O(1/\varepsilon^{\frac{d}{2}-\frac{1}{2}})$ time. Moreover, if we have more than one diametrical pair $(\hat{p}', \hat{q}')$ in point-set $\hat{\mathcal{S}}'$, then this step takes at most $O((2^d)(2)^d/\varepsilon^{\frac{d}{2}-\frac{1}{2}}) = O(2^{2d}/\varepsilon^{\frac{d}{2}-\frac{1}{2}})$ time. Therefore, we can write the total complexity time of the algorithm as follows, which includes required time for three times rounding the point-sets to the grids with $O(dn)$ time, and the required time to find points inside the hypercubes $(\mathcal{B}_1, \mathcal{B}_2)$ or $(\mathcal{B}'_1, \mathcal{B}'_2)$, which can take at most $O(2^d dn)$ time, plus the time of calculating the diameter in each of three sets of rounded points:

$$\begin{aligned} T_d(n) &= O(dn) + O(dn) + O(dn) + O(2^d dn) + O(\frac{(2\sqrt{d})^{2d}}{\varepsilon^{\frac{d}{2}-\frac{1}{2}}}) \\ &\quad + O(2^d dn) + O(\frac{(2\sqrt{2})^{2d}}{\varepsilon^{\frac{d}{2}-\frac{1}{2}}}) + O(\frac{2^{2d}}{\varepsilon^{\frac{d}{2}-\frac{1}{2}}}), \\ &\leqslant O(2^d dn + \frac{(2\sqrt{d})^{2d}}{\varepsilon^{\frac{d}{2}-\frac{1}{2}}}). \end{aligned} \quad (14)$$

Since d is fixed, we have:

$$T_d(n) = O(n + \frac{1}{\varepsilon^{\frac{(d-1)}{2}}}). \quad (15)$$

On the other hand, we know that according to Lemma 1, the points of the set $\hat{\mathcal{S}}$ provide a $(1+\varepsilon)$-approximate for the true diameter of the point-set $\mathcal{S}$, and also the Chan's recursive approach computes a $(1+O(\varepsilon))$-approximate for the diameter of the point-set $\hat{\mathcal{S}}$. So, the proposed algorithm provides a $(1+O(\varepsilon))$-approximation diameter for the diameter of the initial point-set $\mathcal{S}$. About the required space, we only need $O(n)$ space for storing required point-sets. So, this completes the proof. ■

IV. Conclusion and Future Work

In this paper, we have presented a simple $(1+O(\varepsilon))$-approximation algorithm to compute the diameter of a point-set $\mathcal{S}$ of n points in $\mathbb{R}^d$ for a fixed dimension d with $O(n + 1/\varepsilon^{(d-1)/2})$ time, where $0 < \varepsilon \leqslant 1$. As an open problem, one can ask whether it is possible that the point-set diameter is computed in $O(n + 1/\varepsilon^{\alpha})$ time such that $\alpha < d/2$.

References

[1] P. Techa-angkoon and S. Rattanaudomsawat, "Time-optimal algorithm for computing the diameter of a point set on a completely overlapping network,"*In Proceedings of the International Multi Conference of Engineers and Computer Scientists (IMECS 2009)*, vol. 1, 2009.

[2] A. Ganesan, "An Efficient Algorithm for the Diameter of Cayley Graphs Generated by Transposition Trees," *IAENG International Journal of Applied Mathematics*, vol. 42, no. 4, pp. 214-223, 2012.

[3] F. P. Preparata and M. I. Shamos, "Computational Geometry: an Introduction," New York, Springer-Verlag, pp. 176-182, 1985.

[4] E. A. Ramos, "An optimal deterministic algorithm for computing the diameter of a three-dimensional point set," *Discrete and Computational Geometry*, vol. 26, pp. 245-265, 2001.

[5] O. Egecioglu and B. Kalantari, "Approximating the diameter of a set of points in the Euclidean space," *Information Processing Letters*, vol. 32, no. 4, pp. 205-211, 1989.

[6] P. K. Agarwal, J. Matousek, and S. Suri, "Farthest neighbors maximum spanning trees and related problems in higher dimensions," *Computational Geometry: Theory and Applications*, vol. 1, pp. 189-201, 1992.

[7] G. Barequet and S. Har-Peled, "Efficiently approximating the minimum-volume bounding box of a point set in three dimensions," *Journal of Algorithms*, vol. 38, pp. 91-109, 2001.

[8] S. Har-Peled, "A practical approach for computing the diameter of a point set," *In Proceedings of the 17th annual Symposium on Computational Geometry (SoCG'01)*, pp. 177-186, 2001.

[9] T. M. Chan, "Approximating the diameter, width, smallest enclosing cylinder, and minimum-width annulus," *International Journal of Computational Geometry and Applications*, pp. 67-85, 2002.

[10] T. M. Chan, "Faster core-set constructions and data stream algorithms in fixed dimensions," *Computational Geometry: Theory and Applications*, vol. 35, pp. 20-35, 2006.

[11] M. Imanparast, S. N. Hashemi, and A. Mohades, "An Efficient Approximation for Point-set Diameter in Higher Dimensions," *In Proceedings of the 30th Canadian Conference on Computational Geometry (CCCG'18)*, pp. 72-77, 2018.

[12] T. M. Chan, "Applications of Chebyshev polynomials to low-dimensional computational geometry," *In Proceedings of the 33rd International Symposium on Computational Geometry (SoCG'17)*, pages 26:1-15, 2017.

[13] S. Arya, G. D. da Fonseca, and D. M. Mount, "Near-optimal ε-kernel construction and related problems," *In Proceedings of the 33rd International Symposium on Computational Geometry (SoCG'17)*, pages 10:1-15, 2017.