A simple approximation algorithm for the diameter of a set of points in an Euclidean plane

Jieying Hong¹, Zhipeng Wang², Wei Niu²,³*

¹ ESSEC Asia-Pacific, ESSEC Business School, Singapore, Singapore, ² École Centrale Pékin, Beihang University, Beijing, China, ³ Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing, China

* wei.niu@buaa.edu.cn

Abstract

Approximation algorithms with linear complexities are required in the treatments of big data, however, present algorithms cannot output the diameter of a set of points with arbitrary accuracy and near-linear complexity. By introducing the partition technique, we introduce a very simple approximation algorithm with arbitrary accuracy \( \varepsilon \) and a complexity of \( O(N + \varepsilon^{-1} \log \varepsilon^{-1}) \) for the cases that all points are located in an Euclidean plane. The error bounds are proved strictly, and are verified by numerical tests. This complexity is better than existing algorithms, and the present algorithm is also very simple to be implemented in applications.

Introduction

Given a finite set of points \( T \) in a 2D Euclidean plane \( \mathbb{R}^2 \), its diameter, denoted by \( d_T \), is defined as the maximum distance between two points of \( T \). Computing the diameter of a point set is a fundamental problem in computer science. It has been proved that in an Euclidean plane, finding the accurate diameter of a set of \( N \) points can be reduced to formulating the convex hull of them, with a lower bound of complexity \( O(N \log N) \) [1–4].

In the science of big data, this classical problem encounters new challenges. For big data, the number of points \( N \) can be huge, and one usually expects linear or sub-linear algorithms to replace the \( O(N \log N) \) complexity. Clearly, as \( O(N \log N) \) is the lower bound, they will certainly be approximate algorithms. In the present paper we only consider the algorithms without pre-processing. For these cases, no sub-linear algorithm can guarantee the accuracy of the approximate diameter, as listing the points will require a minimum \( O(N) \) complexity. Therefore, if we want to obtain an estimable approximate diameter, a linear complexity should be the lower bound.

As an introduction, we here show an easiest approximate algorithm. Given an arbitrary point \( p \), this algorithm simply reports its maximum distance to other points, i.e.,

\[
\max_{i,j \in \{1, \ldots, N\}, i \neq j} T_i T_j \quad \text{with} \quad T_i T_j \quad \text{the distance between points} \quad T_i, T_j \in T, \quad \text{as the approximate diameter} \quad d_a. \n\]

It is simply to show that \( 1 \leq d_T/d_a \leq 2 \), implying a very low accuracy of the approximation.
Approximation for the diameter of a set of points

There exist two references for improving this approximation to higher accuracy. Egecioglu and Kalantari designed an algorithm that in \( m \) iterations the reported approximate diameter \( d_m \) satisfies that \( 1 \leq d_m / d_o \leq \sqrt{5 - 2\sqrt{3}} \approx 1.24 \) [5]. Recently, Alipour et al. improved this algorithm to allow fewer iterations, however, the accuracy of the approximation did not change [6].

Another type of approximation problems allows introducing an arbitrary positive number \( 0 < \varepsilon < 1 \) aims at outputting an approximate diameter \( d_o \) in linear time, such that

\[
1 - \varepsilon \leq \frac{d_o}{d_o} \leq 1 + \varepsilon.
\]

Note that this is equivalent to the description that \( 1 \leq \frac{d_o - \varepsilon d_o}{d_o} \leq \frac{1 + \varepsilon}{1 + \varepsilon} \), indicating that if Eq (1) is satisfied, we can output \( d_o = (1 - \varepsilon)d_o \) instead of \( d_o \) to satisfy that

\[
1 \leq \frac{d_o}{d_o} \leq 1 + \frac{\varepsilon}{1 - \varepsilon},
\]

which is formally consistent to literature. These problems are usually named as \((1 + \varepsilon)\)-approximations. In two dimensions, lots of approximation algorithms with various near-linear complexities have been developed. Table 1 gives a comparison of different \((1 + \varepsilon)\)-approximation algorithms to compute the diameter of \( T \), in chronological order. However, we remark that most of these algorithms are difficult to be implemented in practices since they require complicated calculations and designs in computational geometry. In this paper, we will introduce a very simple algorithm to approach a near-linear complexity of \( O(N + \varepsilon^{-1} \log \varepsilon^{-1}) \) which is much simpler to be implemented in applications.

### Approximation algorithm for the diameter in an Euclidean plane

For the finite set of points \( T \) in \( \mathbb{R}^2 \), we choose a point \( O \in T \) arbitrarily as the origin, and then divide the plane into \( 6n \) same regions with \( n \in \mathbb{Z}^+ \). In each region \( S_i \) \((i = 1, \ldots, 6n)\), we can find a farthest point from the origin, and let \( r_i \) denote the distance between the farthest point of \( S_i \) and the origin. By using the origin \( O \) as the the center of a circle and \( r_i \) as the radius, we can obtain \( 6n \) sector regions, as Fig. 1. We remark that the number of the regions, \( 6n \), will allow in the following parts to provide analytical error estimation. Let \( p_i \) \((i = 1, \ldots, 6n)\) be the midpoint of the arc of each sector region, and compute the largest distance \( d_p \) of these \( 6n \) midpoints \( p_i \' s \). Then we propose the following main theorem of this paper which shows the relationship between the diameter \( d_T \) of the point set \( T \) and the largest distance \( d_p \). Here we note that the virtual points \( p_i \) can be different with the real points in \( T \).

### Table 1. Comparison of different \((1 + \varepsilon)\) – approximation algorithms to compute the diameter of \( T \).

| Author       | complexity       | main method                      |
|--------------|-----------------|----------------------------------|
| Agarwal 1992 | \( O(\varepsilon^{-2} N) \) | Random – sampling               |
| Barequet 1999 | \( O(N + \varepsilon^{-3}) \) | Rounding to Grid                |
| Chan 2000    | \( O(N + \varepsilon^{-3}) \) | Rounding to Grid + Cones        |
| Chan 2000    | \( O(N + \varepsilon^{-2}) \) | Grid + Cones + Dimension Reduction |
| Arya 2014    | \( O(N + \varepsilon^{-3} N^{3/32}) \) | \( \varepsilon \) – Dependencies |
| Our algorithm| \( O(N + \varepsilon^{-1} \log \varepsilon^{-1}) \) | Partition technique            |

https://doi.org/10.1371/journal.pone.0211201.t001
Theorem 1. \( d_T \) is the diameter of a finite set of points \( T \) in \( \mathbb{R}^2 \), and \( d_p \) is the largest distance of the 6n virtual midpoints \( p_i \) as defined above, then the following statement holds:

\[
\frac{1}{\sqrt{2\left(1 + \cos\left(\frac{2n-1 \pi}{3n}\right)\right)}} \leq \frac{d_T}{d_p} \leq \sqrt{2\left(1 + \cos\left(\frac{2n-1 \pi}{3n}\right)\right)}. \tag{3}
\]

In the following we will give the proof of Theorem 1 in two parts: to prove the upper and lower bounds of \( \frac{d_T}{d_p} \) respectively.

**Lower bound of \( \frac{d_T}{d_p} \)**

Without loss of generality, suppose that an endpoint of the line segment \( d_p \) is in the region \( S \), and then we denote the opposite angle region by \( S_0 \) and denote the other regions clockwise by \( S_1, \ldots, S_{6n-1} \). Note that in this way the region \( S \) is exactly the region \( S_{3n} \). Let the line passing through the origin \( O \) and the midpoint of the arc of the region \( S \) be the x-axis, then we can set up the Cartesian coordinate system in the plane, as Fig 2. The coordinate of the midpoint \( p_i \) of the arc in each region \( S_i \) is \((-r_i \cos\left(\frac{m \pi}{3n}\right), r_i \sin\left(\frac{m \pi}{3n}\right))\), where \( i = 1, \ldots, 6n - 1 \), and \( r_i \) is the radius of
the sector region $S_i$. Before giving the proof on the lower bound of $d_T$, we bring out the following lemmas.

**Lemma 1.** If an endpoint of the line segment $d_p$ is in the region $S$ (i.e., $S_{3n}$) as we supposed above, then the other endpoint of $d_p$ cannot be obtained in the regions $S_{2n+1}, \ldots, S_{3n-1}$.

**Proof.** Denote $R = \max(r_0, \ldots, r_{6n-1})$, then the relationship $R \leq d_p$ is obvious. For point $p_i(-r, \cos \frac{\pi}{3n}, r_1 \sin \frac{\pi}{3n})$ in the region $S_{2n+1}, \ldots, S_{3n-1}$ (i.e., $i \in [2n+1, 3n-1]$), and the point $p_{3n}(-r, 0)$ in the region $S$, we can compute the distance between these two points:

$$p_ip_{3n} = r^2 + r_1^2 + 2rr_1 \cos \frac{\pi}{3n}.$$ (4)

Obviously, $p_ip_{3n} < r^2 + r_1^2 - rr_1$.

Let $f(r) = r^2 + r_1^2 - rr_1$, then $f(r) = 2r - r_1$ can be easily obtained. Since $r, r_1 \in [0, R]$, then we have that $f(r) < 0$ when $r \in [0, r_1/2)$, and $f(r) > 0$ when $r \in (r_1, R]$. Thus $f(r)_{\text{max}} = f(0)$ or $f(R)$.

When $r = 0$, $f(r)_{\text{max}} = f(0) = r_1^2 \leq R^2$; and when $r = R$, $f(r)_{\text{max}} = f(R) = R^2 + r_1^2 - Rr_1$.

Let $g(r_i) = f(r_i) = R^2 + r_1^2 - Rr_1$, since $r_i \in [0, R]$, we can obtain the maximum of $g(r_i)$:

$g(r_i)_{\text{max}} = g(0)$ or $g(R)$, and $g(0) = g(R) = R^2$. Thus $g(r_i) \leq R^2$. 

Fig 2. Diagram of the $6n$ regions for the point set and the Cartesian coordinate system. 

https://doi.org/10.1371/journal.pone.0211201.g002
Therefore, \( \overline{pq}_3^2 < r^2 + r_i^2 - rr_i \leq R_i^2 \), and the equality arrives when \( (r = R, r_i = 0) \) or \( (r = R, r_i = R) \) or \( (r = R, r = 0) \). In this way, we can prove that \( \overline{pq}_3^2 \leq R_i^2 \leq d_p^2 \). That means the distance \( \overline{pq}_3^2 \) \( (i = 2n + 1, \ldots , 3n - 1) \), is always less than \( d_p \).

According to the symmetry of the regions, the following lemma can be easily obtained.

**Lemma 2.** If an endpoint of the line segment \( d_p \) is in the region \( S_i \) (i.e., \( S_{3n} \)), then the other endpoint of \( d_p \) cannot be obtained in the regions \( S_{3n+1}, \ldots , S_{4n-1} \).

Moreover, the cases for an endpoint of \( d_p \) in the regions \( S_0, S_6n \), \( \ldots , S_4n \) are equivalent to those in the regions \( S_0, S_1, \ldots , S_{2n} \). Therefore, if we suppose that an endpoint of the line segment \( d_p \) is in the region \( S_i \), then from Lemma 1 and 2, we only need to consider the \( 2n + 1 \) cases where the other endpoint of \( d_p \) is in the regions \( S_0, \ldots , S_{2n} \). In what follows we will consider two cases in order to compute the lower bound of \( \frac{d_p}{d_1} \).

**Case 1:** \( i \in [0, 2n - 1] \)

As we supposed above, an endpoint of \( d_p \) is in the region \( S_i \), then there certainly exists a point \( q_1 \) on the arc of the region \( S_i \). The coordinate of \( q_1 \) is \( (r \cos \theta, r \sin \theta) \), where \( \theta \in \left[-\frac{\pi}{6n}, \frac{\pi}{6n}\right] \) and \( r \) is the radius of the sector region \( S_i \). If the other endpoint of \( d_p \) is in the region \( S_{i+1} \) or \( S_{i-1} \), then there certainly exists a point \( q_2 \) on the arc of the region \( S_{i} \) and the coordinate of \( q_2 \) is \( (-r_i \cos \theta_i, r_i \sin \theta_i) \), where \( \theta_i \in \left[ \frac{\pi}{3n}, \frac{2\pi}{3n}, \frac{4\pi}{3n} \right] \) and \( r_i \) is the radius of the sector region \( S_i \).

The distance between the two points \( q_1 \) and \( q_2 \) can be computed as

\[
\overline{q_1q_2}^2 = (r \cos \theta_i + r \cos \theta)^2 + (r \sin \theta_i - r \sin \theta)^2
\]

\[
= r_i^2 + r^2 + 2rr \cos (\theta_i + \theta),
\]

where \( \theta_i + \theta \in \left[ \frac{(i-1)\pi}{3n}, \frac{(i+1)\pi}{3n} \right] \), for \( i = 0, 1, \ldots , 2n - 1 \). Since \( \cos x \) is monotone decreasing in \( \left[ \frac{2\pi}{3n}, \frac{4\pi}{3n} \right] \), we have \( \cos (\theta_i + \theta) \geq \cos \left( \frac{(i+1)\pi}{3n} \right) \). Thus

\[
\overline{q_1q_2}^2 \geq r_i^2 + r^2 + 2rr \cos \left( \frac{(i+1)\pi}{3n} \right).
\]

From the definition of \( d_1 \), we know that

\[
d_1 \geq \overline{q_1q_2} \geq \sqrt{r_i^2 + r^2 + 2rr \cos \left( \frac{(i+1)\pi}{3n} \right)}.
\]

Since \( d_p = \overline{pq}_3 = \sqrt{r^2 + r_i^2 + 2rr \cos \frac{\pi}{3n}} \), the following relationship can be obtained.

\[
\frac{d_1}{d_p} \geq \sqrt{\frac{r^2 + r_i^2 + 2rr \cos \left( \frac{(i+1)\pi}{3n} \right)}{r^2 + r_i^2 + 2rr \cos \frac{\pi}{3n}}}.
\]

From (8), we have

\[
\frac{d_1}{d_p} \geq \sqrt{1 + \frac{2rr \left( \cos \frac{(i+1)\pi}{3n} - \cos \frac{\pi}{3n} \right)}{r^2 + r_i^2 + 2rr \cos \frac{\pi}{3n}}} = \sqrt{1 + \frac{2 \left( \cos \frac{(i+1)\pi}{3n} - \cos \frac{\pi}{3n} \right)}{r + r_i + 2 \cos \frac{\pi}{3n}}.
\]

When \( x \in [0, 2n - 1] \), the function \( \cos \frac{\pi x}{3n} \) is monotone decreasing, thus \( \cos \frac{(i+1)\pi}{3n} - \cos \frac{i\pi}{3n} < 0 \).
In this subsection, we will prove the upper bound of $d_T$. Then we can obtain that

$$
\frac{d_T}{d_p} \geq \sqrt{1 + \frac{2}{1 + \cos \frac{(i+1)\pi}{3n}}} \geq \sqrt{1 + \cos \frac{(i+1)\pi}{3n}}.
$$

(10)

Let $f(x) = \frac{1 + \cos \frac{(i+1)\pi}{3n}}{1 + \cos \frac{2\pi}{3n}}$ and $x \in [0, 2n - 1]$. We consider the monotonicity of $f(x)$ and compute its derivative for this purpose.

$$
f'(x) = \frac{\frac{1}{\sqrt{3}} \left[ - \sin \frac{(i+1)\pi}{3n} \left( 1 + \cos \frac{2\pi}{3n} \right) + \sin \frac{\pi}{3n} \left( 1 + \cos \frac{(i+1)\pi}{3n} \right) \right]}{(1 + \cos \frac{2\pi}{3n})^2}.
$$

(11)

Since the denominator of $f'(x)$ is always greater than 0, we only consider the sign of its numerator. Let $g(x)$ be the numerator of $f'(x)$ divided by $\pi/(3n)$.

$$
g(x) = -\sin \frac{(i+1)\pi}{3n} \left( 1 + \cos \frac{2\pi}{3n} \right) + \sin \frac{\pi}{3n} \left( 1 + \cos \frac{(i+1)\pi}{3n} \right)
= \sin \frac{\pi}{3n} - \sin \frac{(i+1)\pi}{3n} + \sin \left( \frac{\pi}{3n} - \frac{(i+1)\pi}{3n} \right)
= \sin \frac{\pi}{3n} - \sin \frac{(i+1)\pi}{3n} - \sin \frac{\pi}{3n}.
$$

(12)

The derivative of $g(x)$ is $\frac{\pi}{3n} \left( \cos \frac{(i+1)\pi}{3n} - \cos \frac{(i+1)\pi}{3n} \right)$. In the case $x \in [0, 2n - 1]$, it is obviously that $g'(x) > 0$, and we can get that $g(0) < 0$ and $g(2n - 1) < 0$ by computation. Thus for any $x \in [0, 2n - 1]$, $g(x) < 0$ and also $f'(x) < 0$. Therefore we know that $f(x)$ is monotone decreasing in the interval $[0, 2n - 1]$.

All this leads up to the following inequality:

$$
\frac{d_T}{d_p} \geq \sqrt{1 + \cos \frac{(i+1)\pi}{3n}} \geq \sqrt{2(2n - 1)} = \frac{1}{\sqrt{2(1 + \cos \frac{(2n-1)\pi}{3n})}}.
$$

(13)

**Case II: $i = 2n$**

In this case $d_p = \overline{p_{2n}p_{3n}} = \sqrt{r^2 + r_i^2 - rr_i}$, and from the proof of Lemma 1 we know that $\sqrt{r^2 + r_i^2 - rr_i} \leq R$. Moreover, since $d_T \geq R$, we have

$$
\frac{d_T}{d_p} \geq \frac{R}{\sqrt{r^2 + r_i^2 - rr_i}} \geq R = 1 > \frac{1}{\sqrt{2(1 + \cos \frac{(2n-1)\pi}{3n})}}.
$$

(14)

Concluding the two cases, we can obtain the lower bound of $\frac{d_T}{d_p}$:

$$
\frac{d_T}{d_p} \geq \frac{1}{\sqrt{2(1 + \cos \frac{(2n-1)\pi}{3n})}}.
$$

(15)

**Upper bound of $\frac{d_T}{d_p}$**

In this subsection, we will prove the upper bound of $\frac{d_T}{d_p}$. Similar to the approach for the proof of the lower bound, suppose that an endpoint of the line segment $d_T$ is in the region $S$, and then we only need to consider the cases that the other endpoint of $d_T$ is in the region $S_i$ for $i \in [0, 2n]$. 
**Case 1: \( i \in [1, 2n] \)**

As we supposed above, an endpoint of the distance between the two points is in the region \( S_i \) and the other endpoint is in the region \( S_j \) respectively. The coordinates of \( m_1 \) and \( m_2 \) are \((\hat{r} \cos \theta, \hat{r} \sin \theta)\) and \((\tilde{r} \cos \theta_i, \tilde{r} \sin \theta_i)\) respectively, where \( \hat{r} \in [0, r_i], \tilde{r} \in [0, r_j], \theta \in \left[ -\frac{\pi}{2m}, \frac{\pi}{2m} \right] \) and \( \theta_i \in \left[ -\frac{\pi}{2m}, \frac{\pi}{2m} \right] \). The distance between the two points \( \overline{m_i m_j} \), is exactly \( d_T \) and thus \( \overline{m_i m_j} \geq R \).

Furthermore,
\[
\overline{m_i m_j}^2 = (\hat{r}_i \cos \theta_i + \hat{r} \cos \theta)^2 + (\hat{r}_i \sin \theta_i - \hat{r}_i \sin \theta)^2 \]
\[
= \hat{r}^2 + \hat{r}_i^2 + 2\hat{r}_i \hat{r} \cos (\theta_i + \theta),
\]
where \( \theta_i + \theta \in \left[ \left( -\frac{i-1}{3m} \right), \left( \frac{i+1}{3m} \right) \right] \), for \( i = 1, \ldots, 2n \). Since \( \cos x \) is monotone decreasing in \([0, \pi] \), we have \( \cos (\theta_i + \theta) \leq \cos \left( \frac{(i-1)\pi}{3m} \right) \). Thus
\[
\overline{m_i m_j}^2 \leq \hat{r}^2 + \hat{r}_i^2 + 2\hat{r}_i \hat{r} \cos \left( \frac{(i-1)\pi}{3m} \right).
\]

Let \( h(\hat{r}) = \hat{r}^2 + \hat{r}_i^2 + 2a\hat{r}_i \hat{r} \), where \( a = \cos \left( \frac{(i-1)\pi}{3m} \right) \in \left( -\frac{1}{2}, 1 \right] \).

1. In the case \( a = \cos \left( \frac{(i-1)\pi}{3m} \right) \in [0, 1] \), \( h(\hat{r}) = \hat{r}^2 + \hat{r}_i^2 + 2a\hat{r}_i \hat{r} \leq \hat{r}^2 + \hat{r}_i^2 + 2ar \). And since \( d_p \geq \overline{m_i m_j} = \sqrt{r^2 + r_i^2 + 2rr \cos \frac{i\pi}{3m}} \), we know that
\[
\frac{d_x}{d_p} \leq \frac{\sqrt{r^2 + r_i^2 + 2rr \cos \frac{(i-1)\pi}{3m}}}{\sqrt{r^2 + r_i^2 + 2rr \cos \frac{i\pi}{3m}}}.
\]

2. In the case \( a = \cos \left( \frac{(i-1)\pi}{3m} \right) \in \left( -\frac{1}{2}, 0 \right) \), we can compute that \( h'(\hat{r}) = 2(\hat{r} + a\hat{r}_i) \).

a. If \( \hat{r}_i \geq 2r \), then \( h'(\hat{r}) \leq 0 \) when \( \hat{r} \in [0, r] \). And we can get the maximum of \( h(\hat{r}) \):
\[
h(\hat{r})_{\text{max}} = h(0) = \hat{r}_i^2 \leq R^2.
\]
Thus \( d_x = \overline{m_i m_j} \leq R \), and as we mentioned above, \( \overline{m_i m_j} \geq R \), those lead that
\[
d_x = \overline{m_i m_j} = R \text{ and } r_i = R, r = 0.
\]

In this moment,
\[
d_p \geq \overline{m_i m_j} = \sqrt{r^2 + r_i^2 + 2rr \cos \frac{i\pi}{3m}} = R.
\]
Therefore we have
\[
\frac{d_x}{d_p} \leq \frac{R}{R} = 1 < \frac{\sqrt{r^2 + r_i^2 + 2rr \cos \frac{(i-1)\pi}{3m}}}{\sqrt{r^2 + r_i^2 + 2rr \cos \frac{i\pi}{3m}}}.
\]

b. If \( 0 \leq \hat{r}_i < 2r \), then \( h'(\hat{r}) \leq 0 \) when \( \hat{r} \in [0, -a\hat{r}_i] \) and \( h'(\hat{r}) \geq 0 \) when \( \hat{r} \in [-a\hat{r}_i, r] \). In this case, \( h(\hat{r})_{\text{max}} = h(0) \) or \( h(\hat{r}) \).
If $h(\hat{r})_{\text{max}} = h(0)$, same as the case (a),

$$\frac{d_{r}}{d_{p}} \leq 1.$$  \hspace{1cm} (22)

If $h(\hat{r})_{\text{max}} = h(r) = \hat{r}^{2} + r^{2} + 2a \hat{r} r$, then let $t(\hat{r}) = \hat{r}^{2} + r^{2} + 2ar \hat{r}$, where $\hat{r}_{i} \in [0, r_{i}]$. The derivative of $t(\hat{r})$ is $2(\hat{r} + ar)$. If $\hat{r}_{i} < -ar$, then $t(\hat{r}_{i}) < 0$, and $t(\hat{r}_{i})_{\text{max}} = t(0) = r^{2}$. In this case, $d_{r} = \overline{m_{1}m_{2}} \leq \sqrt{h(r)} \leq \sqrt{t(0)} = r \leq R$, and then similar to the case (a),

$$\frac{d_{r}}{d_{p}} \leq 1.$$  \hspace{1cm} (23)

If $\hat{r}_{i} \geq -ar$, then $t'(\hat{r}_{i}) \geq 0$, and $t'(\hat{r}_{i})_{\text{max}} = t(r_{i}) = r^{2} + r_{i}^{2} + 2ar r_{i}$. Here we have $d_{r} \leq \sqrt{h(r)} \leq \sqrt{t(r_{i})} = r^{2} + r_{i}^{2} + 2ar r_{i}$, and thus

$$\frac{d_{r}}{d_{p}} \leq \frac{\sqrt{r^{2} + r_{i}^{2} + 2ar r_{i}}} {\sqrt{r^{2} + r_{i}^{2} + 2ar r_{i}}} \cos \frac{(l-1)\pi}{3n}. \hspace{1cm} (24)$$

Since $\frac{\sqrt{r^{2} + r_{i}^{2} + 2ar r_{i}} \cos \frac{(l-1)\pi}{3n}} {\sqrt{r^{2} + r_{i}^{2} + 2ar r_{i}}} > 1$, we can summarize all the cases in Case I and get

$$\frac{d_{r}}{d_{p}} \leq \frac{\sqrt{r^{2} + r_{i}^{2} + 2ar r_{i}} \cos \frac{(l-1)\pi}{3n}} {\sqrt{r^{2} + r_{i}^{2} + 2ar r_{i}}} = 1 + \frac{2}{\cos \frac{(l-1)\pi}{3n}} \left( \frac{\cos \frac{(l-1)\pi}{3n} - \cos \frac{\pi}{3n}} {\cos \frac{\pi}{3n}} \right). \hspace{1cm} (25)$$

Then by using the similar approach for computing the lower bound of $\frac{d_{r}}{d_{p}}$, where $i \in [1, 2n]$, we can deduce the following inequality

$$\frac{d_{r}}{d_{p}} \leq \sqrt{\frac{1 + \cos \frac{(2n-1)\pi}{3n}} {1 + \cos \frac{\pi}{3n}}} = \sqrt{2(1 + \cos \frac{(2n-1)\pi}{3n}). \hspace{1cm} (26)$$

**Case II: $i = 0$**

In this case, $d_{p} \geq \overline{m_{1}m_{\text{in}}} = r_{i} + r$. Moreover,

$$d_{r} = \overline{m_{1}m_{2}} = \sqrt{\hat{r}_{i}^{2} + \hat{r}^{2} + 2r_{i}\hat{r} \cos (\theta_{i} + \theta)} \left( \text{where } \theta_{i} + \theta \in \left[ -\frac{\pi}{3n}, \frac{\pi}{3n} \right] \right)$$

$$\leq \sqrt{\hat{r}_{i}^{2} + \hat{r}^{2} + 2r_{i}\hat{r} \hat{r}} = \hat{r}_{i} + \hat{r} \leq r_{i} + r. \hspace{1cm} (27)$$

Therefore,

$$\frac{d_{r}}{d_{p}} \leq \frac{r_{i} + r}{r_{i} + r} = 1. \hspace{1cm} (28)$$

From Case I and II, the upper bound of $\frac{d_{r}}{d_{p}}$ can be concluded:

$$\frac{d_{r}}{d_{p}} \leq \sqrt{2(1 + \cos \frac{(2n-1)\pi}{3n}). \hspace{1cm} (29)}$$
In this way, we have proved the Theorem 1, and the relationship in this theorem can be also written as

$$\frac{1}{\sqrt{2(1 + \cos \frac{(2n-1)\pi}{3n})}} \leq \frac{d_e}{d_p} \leq \sqrt{2(1 + \cos \frac{(2n-1)\pi}{3n})}. \quad (30)$$

This theorem therefore provides a fast approximation for the diameter of the point set $T$. 

**Remarks**

We remark that the complexities of visiting all $N$ points in set $T$, calculating their polar coordinates and renewing the values of $r_i$ are all linearly $O(N)$. The complexity of calculating the diameter of 6$n$ virtual points is $O(n \log n)$ as introduced in section 1. Indeed, even if we compare all pairs among these points via brute force, the complexity will be at most $O(n^2)$, which will be negligible by comparing to $O(N)$ if $N$ is huge. Therefore, we conclude that our approximation algorithm has a complexity of $O(N + n \log n)$, which is deterministically linear complexity when $N \gg n$.

In addition, recalling the problem descriptions (1) and (2), we can also formulate the outputted diameter and calculate the necessary number of regions $n$ for an arbitrary accuracy $0 < \epsilon < 1$. It is easy to verify that when

$$n > \frac{2\pi}{2\pi - 3 \cos^{-1}\left(\frac{-(1+4\epsilon)(2\epsilon^2)}{2(\epsilon-1)^2}\right)},$$

$$d_o = \frac{d_p}{(1-\epsilon)\sqrt{2(1 + \cos \frac{(2n-1)\pi}{3n})}},$$

$$d_o^\epsilon = \frac{d_p}{\sqrt{2(1 + \cos \frac{(2n-1)\pi}{3n})}}. \quad (31)$$

Eqs (1) and (2) are both satisfied. For small $\epsilon$ Taylor expansion shows that

$$\frac{2\pi}{2\pi - 3 \cos^{-1}\left(\frac{-(1+4\epsilon)(2\epsilon^2)}{2(\epsilon-1)^2}\right)} \approx \frac{\pi}{\sqrt{3}} \epsilon^{-1}, \quad (32)$$

which indicates that

$$n \sim \epsilon^{-1}, \quad (33)$$

and the complexity writes

$$O(N + n \log n) \sim O(N + \epsilon^{-1} \log \epsilon^{-1}). \quad (34)$$

We therefore remark that all values of $d_p$, $d_o$ and $d_o^\epsilon$ can be used as approximate diameters, depending on the accuracy interval one requires.

**Numerical tests**

As illustrated in the previous section, expressions (3), (1) and (2) describe the error bounds of $d_p$, $d_o$ and $d_o^\epsilon$ respectively. However, in practice these upper and lower bounds correspond to the worst cases, while for most situations the error will be even smaller. In this section we show by three different point sets this error distribution, respectively. In the first point set $T_{(1)}$,
the Cartesian coordinate of each point is \((x, y)\), where \(x\) and \(y\) are independent random variables uniformly distributed in \([0, 1)\), leading to a diameter close to the diagonal; in the second point set \(T(2)\), the polar coordinate of each point is \((r, \theta)\), where \(\theta\) is a random variable homogeneously distributed in \([0, \pi)\), and \(r\) is a random variable with Gaussian distribution \(N(0, 1)\) [11]; the third point set \(T(3)\) is chosen from a real database on the positions of fluid particles [12]. Both \(T(1)\) and \(T(2)\) have \(1 \times 10^6\) discrete points, while \(T(3)\) have \(4 \times 10^5\) discrete points. We simply use \(O(n^2)\) brute force method to calculate the diameter of the \(6n\) virtual points. Indeed, this does not yield any inconvenience in the calculation, while the computational time of the case \(n = 100\) is only 1.01 times of that of the case \(n = 1\). Therefore we can conclude that the calculations are of near-linear complexity.

Without loss of generality, here we use our algorithm to output the values of \(d_p\), and compare them with the theoretical error ranges (3), as shown in Fig 3. We randomly select 100 different origin points for each value of \(n\) respectively. Clearly, for all cases, most calculated diameters \(d_p\) are quite close to the real value \(d_T\), which are even quite better than the theoretically worst bounds (shown as dash-dotted lines in Fig 3). These results then show the effectiveness of the present algorithm.

We also present the CPU time in Fig 4. Points are generated similarly to the \(T(1)\) case, i.e., point coordinates are independent random variables uniformly distributed in \([0, 1)\). The partition parameter \(n\) is fixed as 2, 12, 100 and 300 respectively. Calculations are performed via single thread at Intel Core i5-6200U CPU 2.30GHz, interpreted by Python 2.5.1 in the IDLE.
software. Fig 4 shows that the CPU time is linear to the value of \( N \), illustrating that the present algorithm is of nearly linear complexity. In addition, although no optimization is implemented to accelerate the calculations, the real performance is acceptable since calculating the approximate diameter of \( 2 \times 10^6 \) points with \( n = 100 \) (corresponding to relative error \( \varepsilon = 9 \times 10^{-7} \)) only costs about 3 seconds. These evidences suggest the implementation of the present algorithm in real applications.

**Conclusion**

As a fundamental problem of big data, linear approximation algorithms for the diameter of a set of points will be potentially useful. By introducing the partition technique, we introduce an approximation algorithm with arbitrary accuracy and deterministically linear complexity. The implementation of this algorithm is very simple and does not require any complicated data structure. Note that the lower bound of the proposed algorithm is \( O(N + n \log n) \) with \( n \) of the order of \( \varepsilon^{-1} \), while a brute force visiting algorithm for virtual points will increase this to \( O(N + n^2) \). In practice \( n \) will be much smaller than \( N \), therefore \( O(n^2) \) will be negligible by comparing to \( O(N) \). In addition, increasing the number of partition \( n \) does not increase any multiple coefficient to \( O(N) \), which indicates the robustness of the near-linear complexity of our algorithm. Comparing to existing approximation algorithms, the present algorithm shows a lowest complexity \( O(N + \varepsilon^{-1} \log \varepsilon^{-1}) \). Also, another advantage of the present algorithm is that it is very simple to be implemented, which does not require any complicated data structure or geometry calculation.

Fig 4. CPU time with different \( N \) values. Points are generated similarly to the \( T(i) \) case. \( n = 2, 12, 100 \) and 300 respectively.

https://doi.org/10.1371/journal.pone.0211201.g004
The present contribution is a preliminary attempt in 2D plane. For higher dimensional cases, this method might also be extended, but a division of hyper-sphere [13–15] will be required. In those situations, other partition schemes will be more efficient. For example, one may use high-dimensional Cartesian coordinates instead of the division of hyper-sphere. The related accuracy will also be more complicated, and is expected to be investigated in our future work.

Acknowledgments

This work has been supported by the NSFC project 11601023.

Author Contributions

Conceptualization: Jieying Hong, Wei Niu.
Data curation: Wei Niu.
Formal analysis: Zhipeng Wang, Wei Niu.
Investigation: Jieying Hong.
Methodology: Jieying Hong.
Project administration: Jieying Hong.
Validation: Zhipeng Wang.
Writing – original draft: Wei Niu.

References

1. Yao ACC. On constructing minimum spanning trees in k-dimensional spaces and related problems. SIAM J Comput. 1982; 11:721–736. https://doi.org/10.1137/0211059
2. Preparata FP, Shamos MI. Computational geometry: an introduction; 1985.
3. Bentley JL, Preparata FP, Faust GM. Approximation algorithms for convex hulls. Comm ACM. 1982; 25:64–68. https://doi.org/10.1145/358315.358392
4. Malandain G, Boissonnat JD. Computing the Diameter of a Point Set. Int J Comput Geom Appl. 2002; 12(06):489–509. https://doi.org/10.1142/S0218195902001006
5. Eječioğlu O, Kalantari B. Approximating the diameter of a set of points in the Euclidean space. Inform Process Lett. 1989; 32(4):205–211. https://doi.org/10.1016/0020-0190(89)90045-8
6. Alipour S, Kalantari B, Homapour H. Fast approximation and randomized algorithms for diameter. arXiv preprint arXiv:14102195. 2014.
7. Agarwal PK, Matoušek J, Suri S. Farthest neighbors, maximum spanning trees and related problems in higher dimensions. Comput Geom. 1992; 1(4):189–201. https://doi.org/10.1016/0925-7721(92)90001-9
8. Barequet G, Har-peled S. Efficiently approximating the minimum-volume bounding box of a point set in three dimensions. J Algorithms. 1999; 38:82–91.
9. Chan TM. Approximating the diameter, width, smallest enclosing cylinder, and minimum-width annulus. In: Proceedings of the sixteenth annual symposium on Computational geometry. ACM; 2000. p. 300–309.
10. Arya S, Chan TM. Better \( \epsilon \)-Dependencies for Offline Approximate Nearest Neighbor Search, Euclidean Minimum Spanning Trees, and \( \epsilon \)-Kernels. In: Proceedings of the thirtieth annual symposium on Computational geometry. ACM; 2014. p. 416.
11. Knuth D. The Art of Computer Programming 2: Seminumerical Algorithms. MA: Addison-Wesley; 1968.
12. Fang L, Bos WJT, Jin GD. Short-time evolution of Lagrangian velocity gradient correlations in isotropic turbulence. Phys Fluids. 2015; 27:125102. https://doi.org/10.1063/1.4936140
13. Cooper PW. The hypersphere in pattern recognition. Inf Control. 1962; 5(4):324–346. https://doi.org/10.1016/S0019-9958(62)90641-1
14. Katsuki S, Frangopol DM. Hyperspace Division Method for Structural Reliability. J Eng Mech. 1994; 120(11):2405–2427. https://doi.org/10.1061/(ASCE)0733-9399(1994)120:11(2405)

15. Sato K, Yamaji A. Uniform distribution of points on a hypersphere for improving the resolution of stress tensor inversion. J Struct Geol. 2006; 28(6):972–979. https://doi.org/10.1016/j.jsg.2006.03.007