On Computing the Diameter of a Point Set in High Dimensional Euclidean Space*

Daniele V. Finocchiaro[1] and Marco Pellegrini[2]

[1] Scuola Normale Superiore in Pisa
Piazza dei Cavalieri 7, I-56126 Pisa, Italy
fino@cibs.sns.it

[2] Institute for Computational Mathematics of CNR
via S. Maria 46, I-56126 Pisa, Italy
pellegrini@imc.pi.cnr.it

Abstract. We consider the problem of computing the diameter of a set of n points in d-dimensional Euclidean space under Euclidean distance function. We describe an algorithm that in time $O(dn \log n + n^2)$ finds with high probability an arbitrarily close approximation of the diameter. For large values of d the complexity bound of our algorithm is a substantial improvement over the complexity bounds of previously known exact algorithms. Computing and approximating the diameter are fundamental primitives in high dimensional computational geometry and find practical application, for example, in clustering operations for image databases.

1 Introduction

We consider the following problem: given a set of n points in d-dimensional Euclidean space, compute the maximum pair-wise Euclidean distance between two points in the set. This problem is known as the *diameter problem* or the *furthest pair* problem. There are several efficient algorithms for the case when $d = 2$ [32] and $d = 3$ [3, 33, 8, 7, 28] which, however, do not extend to higher dimensional spaces. The diameter problem is one of the basic problems in high dimensional computational geometry [18, 19, 20].[1] In this paper we consider a setting in which the number of points n and the dimension of the space d are equally important in the complexity analysis.

The exact solution to the diameter problem in arbitrary dimension can be found using the trivial algorithm that generates all possible $\Theta(n^2)$ inter-point distances and determines the maximum value. This algorithm runs in time $O(dn^2)$. As noted in [32] substantial improvements of the asymptotic complexity in terms

* A preliminary version of this paper has appeared in the Proceedings of the 7th Annual European Symposium on Algorithms (ESA '99) pp. 366-377.

[1] Gritzmann and Klee have proposed the term *Computational Convexity* to denote the study of combinatorial and algorithmic aspects of polytopes in high dimensional spaces.

of n and d must overcome the fact that for $d \geq 4$ the number of diametral pairs of points can be $\Omega(n^2)$ [14],[2] while computing a single inter-point distance takes time $\Omega(d)$. Yao in [39] gives an algorithm to compute the diameter in time $O(n^{2-a(d)} \log^{1-a(d)} n)$ where $a(d) = 2^{-(d+1)}$ for $d \geq 3$ and fixed. The technique in [39] can be extended to an $o(n^2)$ algorithm in non-fixed dimension for $d \leq {}^1\!/\!{}_2 \log\log n$.

A result of Yao [38] (cited in [39]) shows that all inter-point distances can be computed in time $O(M(n,d) + nd + n^2)$, where $M(n,d)$ is the time to multiply an $n \times d$ matrix by a $d \times n$ matrix. Using the asymptotically fastest known square matrix multiplication algorithm [9] we have $M(n,d) \leq O(n^2 d^{s-2})$ when $n > d$, and $M(n,d) \leq O(dn^{s-1})$ when $n < d$, where $s \approx 2.376$. The furthest pair is then found trivially with $O(n^2)$ extra time.

In order to obtain faster algorithms for large values of d, we relax the requirement by considering algorithms that approximate the diameter up to multiplicative factors arbitrarily close to one. A powerful tool in this approximate setting is the Johnson–Lindenstrauss (JL) Lemma (see [25, 16]), showing that a projection of the data point set on a random linear subspace of dimension $O(\log n)$ preserves with high probability all inter–point distances within a multiplicative factor arbitrarily close to 1. Thus, with an overhead cost of $O(nd \log n)$ operations we can reduce the dimension of the ambient space to $d' \leq O(\log n)$, when $d \geq \Omega(\log n)$. Afterwards, using the trivial exact algorithm, we attain a bound $O(min\{nd \log n + n^2 \log n,\ dn^2\})$. If, instead, Yao's second algorithm is used it is possible to achieve a bound $O(min\{nd \log n + n^2 \log^{s-2} n,\ n^2 d^{s-2}\})$.

The main result in this paper (Theorem 3) is the following: the diameter of a point-set of size n in dimension d is approximated in time $O(dn \log n + n^2)$ within a factor $1 + \varepsilon$, for any real value $\varepsilon > 0$, with probability $1 - \delta$. The constants hidden in the big-Oh notation depend on user controlled parameters ε and δ, but not on d.

Our result improves asymptotically, by a factor $\min\{d,\ \log n\}$, over the trivial exact algorithm with the LJ projection in the range $\Omega(1) \leq d \leq n$. Our algorithm improves asymptotically, by a factor $(\min\{d,\ \log n\})^{s-2}$, over Yao's second algorithm with the LJ projection for d in the range: ${}^1\!/\!{}_2 \log\log n \leq d \leq n$. For $d > n$ the overhead due to the application of the LJ lemma dominates any other cost in all the above mentioned algorithms.

Another approximation algorithm in literature, [12], attains a fixed approximation factor $c = \sqrt{5 - 2\sqrt{3}}$, which is not arbitrarily close to one.

In a recent paper Borodin, Ostrowsky and Rabani [5], amongst other important results, solve the approximate furthest pair problem in time roughly $O(nd \log n + n^{2-\theta(\varepsilon^2)})$ with high probability. Their result uses the LJ projection and improves asymptotically over our bound for $d < n/\log n$, however, for reasonable values of ε, say $\varepsilon = 0.1$, the asymptotic improvement is of the order of $n^{1/100}$, and such gain should be weighted against a more complex coding effort.

[2] Instead, in dimensions 2 and 3, the number of diametral pairs of points is $O(n)$ [13, 34].

1.1 Applications

One of the most interesting applications of the computation of the diameter is in *clustering*: we are given a set of objects, and we want to group them in such a way that objects in the same group are "similar" [22]. One strategy is to map the objects to points in Euclidean space, so that similarity is mapped into closeness, and to seek for clusters of points. One of the most natural measure of quality for a cluster is its diameter. One application is data clustering for images database [21]. To map an image into a point in an high dimensional space, we associate a dimension to each term of a wavelet expansion of the image considered as a two-dimensional piece-wise constant function [15, 24]. Thus similarity clustering of images is translated in the problem of determining clusters of such points. Other applications of clustering algorithms are in statistics, pattern recognition [31], biology, web search engines [6], distributed networks. Note that in such applications the number of dimensions is very high, thus the complexity due to the dimension must be taken into consideration in the design of efficient algorithms. On the other hand, clustering algorithms are usually based on heuristic arguments and approximations, so using an approximation of the diameter is as good as using its exact value.

A different application of the diameter computation is in a greedy heuristic for the maximum-weight matching problem on Euclidean graphs [2, 35]: the greedy algorithm iteratively matches a farthest pair, and deletes those two points from the set, until there are no more points. The weight of the matching produced is at least half of the weight of the maximum matching.

1.2 The Algorithm

Our algorithm has been inspired by a recent technique for nearest neighbor search described by Kleinberg [26]. Although the formulation of the closest pair problem seems not too different from that of the diameter (searching for the smallest inter-point distance instead of the maximum), the mathematical properties of the two problems are quite different so that in general not any efficient algorithm for the closest pair problem yields an efficient one for the farthest pair problem.

Intuitively, the method of Kleinberg is based on the idea that if a vector $x \in \mathbb{R}^d$ is longer than a vector $y \in \mathbb{R}^d$ then this relation is preserved with probability greater than $1/2$ in a projection of x and y over a random line. Thus using several projections of the same set of vectors and a majority voting scheme we can retrieve, with probability close to one, the "actual" relative length relation between x and y. The theory of range spaces of bounded VC-dimension is invoked to determine how many random lines are needed to satisfy the constraints imposed by the error parameter ε and the confidence parameter δ.

1.3 Organization of the Paper

In Section 2 we review some basic properties of projections of random vectors and of range spaces with bounded VC-dimension. In Section 3 we give the pre-processing and query algorithms for furthest point queries. Finally in Section

4 we apply the data structures of Section 3 to the problem of determining the diameter, thus establishing the main result.

2 Basic Results

We denote with S^{d-1} the set of directions in $\mathbb{R}^d$; it can be identified with the set of unit vectors, or with the set of points on the unit sphere. For any two vectors $x, y \in \mathbb{R}^d$, we define the set of directions over which the length of the projection of x is longer or equal than that of y:

$$\mathcal{Z}_{x;y} = \{\, v \in S^{d-1} : |v \cdot x| \geq |v \cdot y| \,\} \ .$$

We denote with $z_{x;y}$ the relative measure of $\mathcal{Z}_{x;y}$ with respect to the entire set of directions; it represents the probability that the projection of x on a random direction is longer than the projection of y:

$$z_{x;y} = \frac{|\mathcal{Z}_{x;y}|}{|S^{d-1}|} = \Pr\left[|v \cdot x| \geq |v \cdot y|\right] \ .$$

Note that for $x \neq y$ the probability that $|v \cdot x| = |v \cdot y|$ is null. Figure 1 shows, by means of simple geometric intersections, the angles *orthogonal* to directions in $\mathcal{Z}_{x;y}$ (marked as "good").

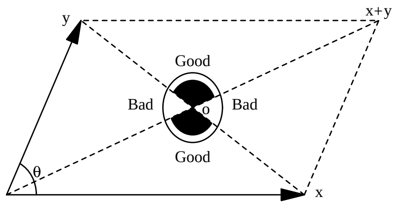

Fig. 1. Good and bad projection angles

The idea of this work is that if x is 'significantly' longer than y, then the set $\mathcal{Z}_{x;y}$ is large, and it is very probable that a random direction belongs to it. The following lemma captures this idea (for a complete proof see [26]; here we detail some minor corrections):

Lemma 1. *If* $(1-\gamma)\|x\| \geq \|y\|$*, with* $0 \leq \gamma \leq 1$*, then* $z_{x;y} \geq 1/2 + \gamma/\pi$.

Proof Sketch.. Consider the plane $\mathcal{A}$ containing the vectors x, y, and the projection $\pi_{\mathcal{A}}(v)$ of v on this plane. Observe that $v \cdot x = \pi_{\mathcal{A}}(v) \cdot x$ and $v \cdot y = \pi_{\mathcal{A}}(v) \cdot y$. So the relation $v \in \mathcal{Z}_{x;y}$ depends only on $\pi_{\mathcal{A}}(v)$, or equivalently on the angle ϕ between $\pi_{\mathcal{A}}(v)$ and the vector x (or y).

Note that ϕ is uniformly distributed in $[0, 2\pi)$. Let θ be the angle formed by x and y: we find that $v \in \mathcal{Z}_{x;y}$ whenever $\cos^2(\theta - \phi) \leq \cos^2\phi \cdot \|x\|^2/\|y\|^2$, and the quantity $z_{x;y}$ is minimum when x and y are orthogonal. In this case

$$z_{x;y} = \frac{2}{\pi} \tan^{-1} \frac{\|x\|}{\|y\|} \geq \frac{2}{\pi} \tan^{-1} \frac{1}{1-\gamma} \enspace .$$

In order to bound to the latter quantity, we consider that $\tan^{-1}(1-\gamma)^{-1}$ is a convex function, and its derivative in $\gamma = 0$ is $1/2$. So we obtain

$$\tan^{-1} \frac{1}{1-\gamma} \geq \tan^{-1} 1 + \frac{1}{2}\gamma = \frac{\pi}{4} + \frac{1}{2}\gamma \enspace ,$$

from which the lemma follows. □

Intuitively, we want to fix a set V of directions, and compare the lengths of vectors by comparing the lengths of their respective projections over elements of V, and making a majority vote. For two vectors x, y we will write that

$$x \rhd_V y \qquad \Longleftrightarrow \qquad |V \cap \mathcal{Z}_{x;y}| \geq 1/2|V| \enspace . \tag{1}$$

This means that the projection of x is longer than the projection of y over at least half of the vectors in V. Note that $x \rhd_V y$ and $y \rhd_V x$ can simultaneously hold with respect to the same set V.

For a fixed set of directions V, consider two arbitrary vectors x and y such that x is significantly longer than y. Lemma 1 says that the set $\mathcal{Z}_{x;y}$ is large, so we can hope that in V there are enough vectors of $\mathcal{Z}_{x;y}$ so that $x \rhd_V y$. However, we want this to hold for *any* vectors x, y, with respect to the *same* (fixed) set V. For this reason we cannot use directly Lemma 1, but we will use some VC-dimension techniques. For a detailed treatment of this topic see [36, 37, 1], or the paper which introduced the use of VC-dimension in computational geometry [23]. Here we will present only the definitions needed.

A *range space* is a pair $(\mathcal{P}, \mathcal{R})$, where $\mathcal{P} = (\Omega, \mathcal{F}, \mu)$ is a probability space, and $\mathcal{R} \subseteq \mathcal{F}$ is a collection of measurable (w.r.t. the measure μ) subsets of Ω. A finite set $A \subseteq \Omega$ is said to be *shattered* by $\mathcal{R}$ if every subset of A can be expressed in the form $A \cap R$ for some $R \in \mathcal{R}$. The *VC-dimension* $\mathit{VC\text{-}dim}(\mathcal{R})$ of the range space $(\mathcal{P}, \mathcal{R})$ is the maximum size of a set that can be shattered by $\mathcal{R}$ (hence no set of size $> \mathit{VC\text{-}dim}(\mathcal{R})$ can be shattered).

There is a natural identification between collections of sets and families of binary-valued function: to each subset $R \in \mathcal{R}$ corresponds its indicator function $f_R(x) : \Omega \to \{0, 1\}$ such that $f_R(x) = 1 \iff x \in R$. This identification is useful to express combinations of range spaces in the following manner.

For an integer $k \geq 2$, let $u : \{0,1\}^k \to \{0,1\}$ and $f_1, \ldots, f_k : \mathcal{R} \to \{0,1\}$ be binary-valued functions (u represents the combination operator and the f_i's are indicator functions). Define $u(f_1, \ldots, f_k) : \mathcal{R} \to \{0,1\}$ as the binary-valued function $x \mapsto u(f_1(x), \ldots, f_k(x))$. Finally, if $\mathcal{A}_1, \ldots, \mathcal{A}_k$ are families of binary-valued functions, we define $\mathcal{U}(\mathcal{A}_1, \ldots, \mathcal{A}_k)$ to be the family of binary-valued functions $\{\, u(f_1, \ldots, f_k) : f_i \in \mathcal{A}_i \quad \forall i \,\}$. In this way we can obtain $\mathcal{A} \oplus \mathcal{B} =$

$\{ A \cup B : A \in \mathcal{A}, B \in \mathcal{B} \}$ by choosing $u(a,b) = \max\{a,b\}$, and $\mathcal{A} \odot \mathcal{B} = \{ A \cap B : A \in \mathcal{A}, B \in \mathcal{B} \}$ by choosing $u(a,b) = a \cdot b$.

We will use the following theorem by Vidyasagar [37, Th. 4.3], which improves on a result by Dudley [10, 11]:

Theorem 1. *If VC-dim($\mathcal{A}_i$) is finite for each i, then $\mathcal{U} = \mathcal{U}(\mathcal{A}_1, \ldots, \mathcal{A}_k)$ also has finite VC-dimension, and VC-dim($\mathcal{U}$) $< \alpha_k \cdot d$, where $d = \max_i$ VC-dim($\mathcal{A}_i$), and α_k is the smallest integer such that $k < \alpha_k / \log_2(e\alpha_k)$.*

Some values of α_k are the following:

k	2	3	4	5	6	7	8	9	10
α_k	10	17	25	33	41	50	59	68	78

We consider the probability space $\mathcal{P}$ made by the set of directions S^{d-1}, with the uniform distribution, where $\mu(X) = |X|/|S^{d-1}|$; we take as $\mathcal{R}$ the collection of all the sets $\mathcal{Z}_{x;y}$. The following lemma bounds the VC-dimension of this range space:

Lemma 2. *The VC-dimension of the range space $(\mathcal{P}, \mathcal{R})$ defined above is strictly less than $25(d+1)$.*

Proof. For a vector $u \in \mathbb{R}^d$, let H_u denote the closed hemisphere $\{ v \in S^{d-1} : u \cdot v \geq 0 \}$, and let $\mathcal{H}$ be the collection of all such hemispheres. Then any set $\mathcal{Z}_{x;y}$ can be expressed as the Boolean combination $(H_{x-y} \cap H_{x+y}) \cup (H_{-x-y} \cap H_{-x+y})$. Thus we have $\mathcal{R} \subseteq \mathcal{U}$, where $\mathcal{U} = (\mathcal{H} \odot \mathcal{H}) \oplus (\mathcal{H} \odot \mathcal{H})$. A theorem by Radon [4] implies that VC-dim($\mathcal{H}$) $\leq d+1$. We then apply Theorem 1, with $u(a,b,c,d) = \max\{a \cdot b, c \cdot d\}$, and the fact that $\alpha_4 = 25$ to obtain that VC-dim($\mathcal{R}$) $\leq$ VC-dim($\mathcal{U}$) $< 25(d+1)$. □

A subset A of Ω is called a γ-approximation (or γ-sample) for a range space $(\mathcal{P}, \mathcal{R})$ is for every $R \in \mathcal{R}$ the following relation holds:

$$\left| \frac{|R \cap A|}{|A|} - \mu(R) \right| \leq \gamma \ . \tag{2}$$

This means that A can be used to obtain a good estimate of the measure of any set $R \in \mathcal{R}$. The main result, which follows from Lemma 2 and from a fundamental theorem by Vapnik and Chervonenkis, is the following.

Lemma 3. *With probability at least $1-\delta$ a set V of cardinality*

$$f(\gamma, \delta) = \frac{16}{\gamma^2} \left(25(d+1) \log \frac{400(d+1)}{\gamma^2} + \log \frac{4}{\delta} \right) = \Theta(d \log d)$$

of vectors chosen uniformly at random from S^{d-1} is a γ-approximation for the range space defined above.

The lemma above permits to make comparisons in the following way: choose a set V of $f(\varepsilon/\pi, \delta)$ random directions. With probability $1-\delta$ it is a (ε/π)-approximation. For any x, y with $(1-\varepsilon)\|x\| \geq \|y\|$, we have that $\mu(\mathcal{Z}_{x;y}) \geq 1/2 + \varepsilon/\pi$, by Lemma 1. By definition of γ-approximation we have that $|\mathcal{Z}_{x;y} \cap V| \geq 1/2|V|$, so $x \triangleright_V y$ by definition (1). This is the idea which we are going to use in the following section. However, in order to save computations, we will make comparisons using random subsets of the fixed set V.

3 An Algorithm for Farthest-point Queries

We will present in this section a (ε, δ)-approximation scheme for computing the farthest site of a query point q. This means that with probability $1-\delta$ the algorithm gives an answer which is within a factor $1-\varepsilon$ from the optimal one. The parameters ε and δ are chosen by the user before the algorithm starts.

Let $P = \{p_1, \ldots, p_n\}$ the set of given sites, and $q \in \mathbb{R}^d$ the query point. For simplicity we will assume that n is a power of 2. Let p^* be the farthest-site from q, i.e., the point such that

$$d(p^*, q) = \max_{p_i \in P} d(p_i, q) \quad ,$$

where $d(p, q) = \|p - q\|$ is the standard Euclidean metric in $\mathbb{R}^d$. Let Z_ε be the set of sites that are far from q 'almost' like p^*:

$$Z_\varepsilon = \{\, p_i \in P : d(p_i, q) \geq (1-\varepsilon) d(p^*, q) \,\} \quad .$$

The purpose of the algorithm is to give, with probability at least $1-\delta$, an element of Z_ε.

3.1 Building the Data Structure

The preprocessing stage is the following. Let $\varepsilon_0 = \log(1+\varepsilon)/\log n$. We choose randomly L vectors from S^{d-1}, where $L = f(\varepsilon_0/12, \delta) = \Theta(d \log d \log^2 n \log\log n)$. These vectors can be obtained for example by the method described by Knuth [27, p. 130]. Let $V = \{v_1, \ldots, v_L\}$ the set of L directions generated above. The data structure is a matrix M, of dimension $L \times n$, where $M[i, j] = v_i \cdot p_j$.

3.2 Processing a Query

We first define the following relation between sites of P: for a set of directions $\Gamma \subseteq V$, we say that $p_i \succeq_\Gamma p_j$ if the projection of p_i is farther than the projection of p_j from the projection of q, for at least half of the directions in Γ. Formally:

$$p_i \succeq_\Gamma p_j \quad \Longleftrightarrow \quad (p_i - q) \triangleright_\Gamma (p_j - q) \quad .$$

We will use this relation to build a 'tournament' between sites, by making comparisons with respect to a fixed set Γ: if $p_i \succeq_\Gamma p_j$ and $p_j \not\succeq_\Gamma p_i$, the winner of the comparison is p_i; if $p_i \not\succeq_\Gamma p_j$ and $p_j \succeq_\Gamma p_i$, the winner is p_j; finally, if both $p_i \succeq_\Gamma p_j$ and $p_j \succeq_\Gamma p_i$, the winner is chosen arbitrarily. Note that the above description is complete, i.e., it cannot occur that $p_i \not\succeq_\Gamma p_j$ and $p_j \not\succeq_\Gamma p_i$.

Phase A. We first extract a random subset $\Gamma \subseteq V$ of cardinality $c_1 \log^3 n$ (the value of c_1 will be given in Lemma 7). We make extractions with replacement, so we permit Γ to be a multi-set. Let $b = \log|\Gamma| = \Theta(\log\log n)$. We assume for simplicity that b has an integer value.

We compute the values $v \cdot q$ for all $v \in \Gamma$. Then we build a complete binary tree T of depth $\log n$. We associate randomly the sites in P to the leaves of T. To every internal node x at height $1 \leq h \leq b$, we associate a random subset $\Gamma_x \subseteq \Gamma$ of size $c_2' + c_2 h$ (the appropriate values for c_2 and c_2' will be given in Lemma 8). To higher nodes, with height $h > b$, we associate the entire set Γ.

Now we make a tournament between the sites, proceeding from the leaves towards the root of T: to each internal node x we associate the winner of the comparison between the sites associated to its children, with respect to the set Γ_x. Let $\tilde{p}_A$ be the winner of this tournament, i.e., the site which at the end of Phase A will be associated to the root of T.

Phase B. Independently from all the above, we randomly choose a subset $P_0 \subseteq P$ of size $c_3 \log^3 n$, where c_3 will be defined in Lemma 6. We compute the distances between q and the sites in P_0. Let $\tilde{p}_B$ be the site of P_0 farthest from q.

The algorithm finishes returning the site among $\tilde{p}_A$ and $\tilde{p}_B$ that is the farthest from q.

3.3 Correctness

We now prove that the algorithm correctly computes an element of Z_ε. First of all, observe that, as a consequence of Lemma 3, the set V created during the preprocessing is an $(\varepsilon_0/12)$-approximation of the range space $(\mathcal{P}, \mathcal{R})$, with probability $1-\delta$. In the following, we will assume that this event occurred.

Lemma 4. *Let V be an $(\varepsilon_0/12)$-approximation, and v a random element of V. Let $q \in \mathbb{R}^d$, $p_i, p_j \in P$, and suppose that $(1-\gamma)d(p_i, q) \geq d(p_j, q)$, with $\varepsilon_0 \leq \gamma \leq 1/2$. The probability that $|v \cdot (p_i - q)| \geq |v \cdot (p_j - q)|$ is at least $1/2 + \gamma/5$.*

Proof. Let $x = p_i - q$ and $y = p_j - q$. The measure $z_{x;y}$ of $\mathcal{Z}_{x;y}$ is at least $1/2 + \gamma/\pi$, by Lemma 1. As V is an $(\varepsilon_0/12)$-approximation, the probability that an element of V belongs to $\mathcal{Z}_{x;y}$ is at most $\varepsilon_0/12$ less than that value. So this probability is at least $z_{x;y} - \varepsilon_0/12 \geq 1/2 + \gamma/5$, and the lemma follows. $\square$

Lemma 5. *Suppose $(1-\gamma)d(p_i, q) \geq d(p_j, q)$, and let Γ' be a random subset of Γ, of cardinality k. The probability that $p_j \succeq_{\Gamma'} p_i$ is at most $e^{-\gamma^2 k/36}$.*

Proof. Let $x = p_i - q$ and $y = p_j - q$. The relation $p_j \succeq_{\Gamma'} p_i$ is equivalent to $y \rhd_{\Gamma'} x$, that means $|\Gamma' \cap \mathcal{Z}_{y;x}| \geq 1/2 k$. As $S^{d-1} \setminus \mathcal{Z}_{y;x} \subseteq \mathcal{Z}_{x;y}$, we need only to bound the probability that $|\Gamma' \cap \mathcal{Z}_{x;y}| \leq 1/2 k$. Define k Bernoulli random variables $X_1, \ldots, X_k$, where X_r is 1 if the r-th vector in Γ' belongs to $\mathcal{Z}_{x;y}$, and 0 otherwise. By application of Lemma 4 we obtain that the probability of success is $\Pr[X_r = 1] \geq 1/2 + \gamma/5$. Defining the random variable $X = \sum_r X_r = |\Gamma' \cap \mathcal{Z}_{x;y}|$, the event we are interested in is $X \leq 1/2 k$. The lemma follows by some calculations and by application of the Chernoff bound (see e.g. [29]).

Here are the details. The expected value of X is $E[X] \geq k(1/2 + \gamma/5)$. Using the fact that $1/2 < (1-\gamma/3)(1/2 + \gamma/5)$ we have:

$$\Pr\left[X \leq \frac{1}{2}k\right] \leq \Pr\left[X < \left(1 - \frac{\gamma}{3}\right)\left(\frac{1}{2} + \frac{\gamma}{5}\right)k\right] = \Pr\left[X < \left(1 - \frac{\gamma}{3}\right)E[X]\right].$$

The Chernoff bound (see e.g. [29, Th. 4.2]) affirms that $\Pr[X < (1-\delta)E[X]] < e^{-\delta^2 E[X]/2}$. Applying this result we obtain

$$\Pr\left[X < \left(1 - \frac{\gamma}{3}\right)E[X]\right] < \exp\left(-\left(\frac{\gamma}{3}\right)^2 k\left(\frac{1}{2} + \frac{\gamma}{5}\right)\frac{1}{2}\right) \leq e^{-\gamma^2 k/36},$$

where we also used that $e^{-p} \leq e^{-1/2}$ for $p \geq 1/2$. $\square$

The motivation for Phase B is that if in the set Z_ε there are too many sites, one of them could eliminate p^* during the early stage of the tournament, and then be eliminated by some site not in Z_ε. On the other hand, if Z_ε is sufficiently small, there is a high probability that p^* will reach at least level b of the tournament tree: in this case, we can show that the winner $\tilde{p}_A$ is an element of Z_ε. Intuitively, at each comparison the winner can be slightly closer to q than the loser: if p^* is eliminated too early in the competition, all these small errors can take us too close to q; if instead p^* reaches at least level b, the final error is small with high probability.

We begin by proving that if Z_ε is large, then Phase B succeeds in finding an approximate farthest-point of q with high probability:

Lemma 6. *If $|Z_\varepsilon| \geq \gamma_1 n/\log^3 n$, then $\tilde{p}_B \in Z_\varepsilon$, with probability at least $1-\delta$.*

Proof. When we randomly select the elements of P_0 from P, the probability of taking an element not in Z_ε is less than $1 - \gamma_1/\log^3 n$. The probability that no element of P_0 belongs to Z_ε is less than that quantity raised to power $|P_0| = c_3 \log^3 n$. We can choose $c_3 \equiv c_3(\gamma_1, \delta, n)$ such that $(1 - \gamma_1/\log^3 n)^{c_3 \log^3 n} \leq \delta$. If we choose such a c_3, with probability at least $1-\delta$ there is at least one element of Z_ε in P_0, so $\tilde{p}_B$ certainly belongs to Z_ε. $\square$

Let us see now what happens if Z_ε is small. Define the constant $\gamma_2 \equiv \gamma_2(\varepsilon)$ such that $e^{-\gamma_2} \geq 1 - \varepsilon$. We denote by $\mathcal{E}_1$ the following event:

$$\mathcal{E}_1 \equiv \left\{ \exists\, p_i, p_j \in P, \quad i \neq j : \quad p_j \succeq_\Gamma p_i, \quad \left(1 - \frac{\gamma_2}{\log n}\right) d(p_i, q) \geq d(p_j, q) \right\},$$

that is, the event that there exist two sites, whose distances from q differ significantly, for which the comparison based on projections on the vectors of Γ gives the wrong answer. We can give a bound to the probability that this event occurs:

Lemma 7. *The probability of event $\mathcal{E}_1$ is less or equal to $\delta/3$.*

Proof. There are $n(n-1) \leq n^2$ ordered pairs of sites to consider: for each pair we apply Lemma 5, with $\gamma = \gamma_2/\log n$ and $k = |\Gamma| = c_1 \log^3 n$, to bound the probability of error. The probability of $\mathcal{E}_1$ is bounded by the sum of these probabilities. The lemma follows by defining suitably the value of $c_1 \equiv c_1(\gamma_2, \delta, n)$ in such a way that $n^2 \cdot e^{-c_1\gamma_2^2 \log n/36} \leq \delta/3$. □

Let us denote by $\mathcal{E}_2$ the event "p^* does not reach level b in the tournament tree", that is, it is not assigned, during the tournament in Phase A, to any node at level b.

Lemma 8. *If $|Z_\varepsilon| < \gamma_1 n/\log^3 n$, the probability of $\mathcal{E}_2$ is less or equal to $2\delta/3$.*

Proof. Consider the leaf of T to which p^* was assigned, the node x at level b along the path from this leaf to the root (x is the node to which p^* should be assigned, if $\mathcal{E}_2$ does not occur), and the sub-tree T^x rooted at x.

We first of all exclude the possibility that in T^x there be sites of Z_ε other than p^*. The number of leaves in T^x is $2^b = |\Gamma| = c_1 \log^3 n$, so the probability that $|T^x \cap Z_\varepsilon| = 1$ is

$$\binom{n-|Z_\varepsilon|}{2^b-1}\binom{n}{2^b-1}^{-1} .$$

By defining suitably $\gamma_1 \equiv \gamma_1(c_1, \delta, n)$ we can make this probability greater than $1-\delta/3$. So we can assume that p^* is the only site of Z_ε in the sub-tree T^x. We now show that in this case it reaches level b with probability $1-\delta/3$.

If p^* reached level $h-1$, it is compared, at level h, with an element not in Z_ε, made using $|\Gamma_x| = c_2' + c_2 h$ vectors of Γ. By application of Lemma 5, with $\gamma = \varepsilon$, the probability that p^* loses the comparison is bounded by

$$e^{-\varepsilon^2(c_2'+c_2 h)/36} = e^{-\varepsilon^2 c_2'/36}(e^{-\varepsilon^2 c_2/36})^h .$$

We can define the constants $c_2' \equiv c_2'(\varepsilon, \delta)$ and $c_2 = c_2(\varepsilon)$ such that the left factor is less than $\delta/3$, and the right factor is less than 2^{-h}.

Summing these quantities for $h = 1, \ldots, b$ we finally obtain that the probability that p^* is eliminated is bounded by $\delta/3$. Together with the fact that the probability that more than one site of Z_ε belongs to T^x is less than $\delta/3$, this implies the lemma. □

We can now prove that if Z_ε is small, Phase A succeeds in finding a good answer:

Lemma 9. *If $|Z_\varepsilon| < \gamma_1 n/\log^3 n$, then $\tilde{p}_A \in Z_\varepsilon$, with probability at least $1-\delta$.*

Proof. With probability $1-\delta$ neither $\mathcal{E}_1$ nor $\mathcal{E}_2$ occur. In this case we show by induction on $h \geq b$ that there exists a site $p_{(h)}$, assigned to a node at level h, such that

$$d(p_{(h)}, q) \geq \left(1 - \frac{\gamma_2}{\log n}\right)^{h-b} d(p^*, q) \ , \tag{3}$$

and this implies that $\tilde{p}_A \in Z_\varepsilon$.

The fact that $\mathcal{E}_2$ does not occur guarantees that p^* reached level b, so we can start the induction, for $h = b$, with $p_{(h)} = p^*$. Suppose now that (3) holds at level h, and let $p'_{(h)}$ be the adversary of $p_{(h)}$. If $d(p'_{(h)}, q) \geq (1 - \gamma_2/\log n)d(p_{(h)}, q)$, then both $p_{(h)}$ and $p'_{(h)}$ satisfy (3) for level $h+1$, so relation (3) holds in any case. If instead $d(p'_{(h)}, q) < (1 - \gamma_2/\log n)d(p_{(h)}, q)$, as we are assuming that $\mathcal{E}_1$ does not occur, certainly $p_{(h)}$ wins the comparison, so $p_{(h+1)} = p_{(h)}$ and relation (3) is valid for level $h+1$.

From the inductive argument we deduce that (3) holds for $h = \log n$, where $p_{(\log n)} = \tilde{p}_A$ is the site assigned to the root of T. Being $b \geq 1$, from the fact that $(1 - x/m)^{m-1} \geq e^{-x}$, and from the definition of γ_2, we obtain

$$d(\tilde{p}_A, q) \geq \left(1 - \frac{\gamma_2}{\log n}\right)^{\log n - 1} d(p^*, q) \geq e^{-\gamma_2} d(p^*, q) \geq (1 - \varepsilon) d(p^*, q) \enspace ,$$

that is what we wanted to prove. □

The correctness of the algorithm follows from both Lemma 6 and Lemma 9.

3.4 Complexity

The preprocessing consists in the computation of a distance, which requires time $O(d)$, for each element of the matrix M. The time needed is thus $O(dLn) = O(d^2 n \log d \log^2 n)$. The space required to store M is $O(Ln) = O(dn \log d \log^2 n)$.

To answer a query, we first compute $v \cdot q$ for $v \in \Gamma$. The time required for this is $O(|\Gamma| \cdot d) = O(d \log^3 n)$. Each comparison of type $p_i \succeq_{\Gamma_x} p_j$ requires time $O(|\Gamma_x|)$, because all values $v \cdot p_i$ are already stored in M. The number of nodes at level h is $n2^{-h}$, so the total number of operations at levels $1 \leq h \leq b$ is

$$\sum_{h=0}^{b} (c'_2 + c_2 h) n 2^{-h} = c'_2 n \sum 2^{-h} + c_2 n \sum h 2^{-h} \leq 2(c'_2 + c_2) n = O(n) \enspace .$$

For levels $h > b$ each comparison is made using the entire set Γ, where $|\Gamma| = 2^b$. We have

$$\sum_{h=b+1}^{\log n} |\Gamma| \cdot n 2^{-h} = \sum_{h=b+1}^{\log n} |\Gamma| \cdot n 2^{b-h} 2^{-b} = n \sum_{h=b+1}^{\log n} 2^{b-h} \leq n \enspace .$$

We thus obtain that the total cost of Phase A is $O(n + d \log^3 n)$. Phase B requires time $O(d \log^3 n)$. We conclude that the time required to answer a query is $O(n + d \log^3 n)$. We summarize the properties of the farthest-point algorithm in the following theorem:

Theorem 2. *With probability $1 - \delta$ the set V created in the preprocessing is an $(\varepsilon_0/12)$-approximation. In this case, the probability that the algorithm returns an element of Z_ε is, for each query, at least $1 - \delta$. The complexity of the algorithm is $O(d^2 \log d \cdot n \log^2 n)$ for the preprocessing and $O(n + d \log^3 n)$ for answering a query.*

4 Computing the Diameter of a Point Set

We now show how to use the algorithm for farthest-point queries to find the diameter d_P of a set of points $P = \{p_1, \ldots, p_n\}$ in $\mathbb{R}^d$. First of all, we use a dimension-reduction technique, by projecting all the points on a random subspace of dimension $k = O(\varepsilon^{-2} \log n)$. Let P' the resulting set of points in $\mathbf{R}^k$ and $d_{P'}$ its diameter. The Johnson–Lindenstrauss Lemma (see [25, 16]) affirms that with high probability all inter-point distances are preserved, so that $(1 + \varepsilon/2)d_P \geq d_{P'} \geq (1 - \varepsilon/2)d_P$.

Next, we compute the set V and the matrix M as in the preprocessing stage of the farthest-point algorithm, using $\varepsilon/2$ as approximation parameter. For each $p'_i \in P'$, we perform a farthest-point query where $q = p'_i$ and the sites are the remaining points in P'; let $F_\varepsilon(p'_i)$ the point returned by the algorithm described in Section 3. We compute all the distances $d(p'_i, F_\varepsilon(p'_i))$. Let $\tilde{d}_{P'}$ the maximum distance computed in this way. Our main result is the following:

Theorem 3. *Let d_P be the diameter of the point set P, and $\tilde{d}_{P'}$ the value computed by the algorithm above. Then, with probability $1 - \delta$, it holds that $(1 + \varepsilon/2)d_P \geq \tilde{d}_{P'} \geq (1 - \varepsilon)d_P$. The complexity of the algorithm is $O(nd \log n + n^2)$.*

Proof. Let (p^*, q^*) be a pair of points in P' such that $d(p^*, q^*) = d_{P'}$. When we make the farthest-point query with $q = p^*$, the point $F_\varepsilon(p^*)$ is, with probability $1 - \delta$, such that $d(p^*, F_\varepsilon(p^*)) \geq (1 - \varepsilon/2)d_{P'}$. The value $\tilde{d}_{P'}$ computed in the last step is $\tilde{d}_{P'} \geq (1 - \varepsilon/2)d_{P'} \geq (1 - \varepsilon/2)^2 d_P \geq (1 - \varepsilon)d_P$. Moreover, the distances computed by this algorithm are certainly less than $d_{P'}$.

The time to perform the projection at the first step is $O(nd \log n)$. Then we perform the preprocessing and n queries of the algorithm in Section 3, in dimension $O(\log n)$. As a consequence of Theorem 2, we obtain that the overall complexity of the algorithm is $O(nd \log n + n^2)$. □

Note that by taking the points whose projected distance is $\tilde{d}_{P'}$, and computing their distance in the original d-dimensional space, we can obtain a value $\tilde{d}'_{P'}$ such that $d_P \geq \tilde{d}'_{P'} \geq (1 - \varepsilon)d_P$.

5 Conclusions

We have shown how the application of randomized projection techniques results in an algorithm which computes an approximation of the diameter in time $O(n^2)$, independent of the dimension, for dimensions up to $n/\log n$. The algorithm is simple (although its analysis is not) and has a potential for an impact on practical application.

Acknowledgements. We thank Piotr Indyk and Jon Kleinberg for pointing out the paper [5], and Yuval Rabani for sending us a preliminary version of it. We thank Jirka Matoušek for his comments and suggestions and Andrew Yao for clarifications on his algorithms in [38]. We thank an anonymous referee who kindly pointed out a mistake in a previous version of this paper.

References

[1] Noga Alon and Joel H. Spencer. *The Probabilistic Method.* Wiley, 1992.

[2] David M. Avis. A survey of heuristics for the weighted matching problem. *Networks*, 13:475–493, 1983.

[3] Sergei N. Bespamyatnikh. An efficient algorithm for the three-dimensional diameter problem. In *Proc. 9th Symp. on Discrete Algorithms*, pages 137–146, 1998.

[4] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 4(36):929–965, 1989.

[5] Allan Borodin, Rafail Ostrovsky, and Yuval Rabani. Subquadratic approximation algorithms for clustering problems in high dimensional spaces. In *Proc. 31st ACM Symp. on the Theory of Computing*, pages 435–444, 1999.

[6] Andrei Broder, Steven Glassman, Mark Manasse, and Geoffrey Zweig. Syntactic clustering of the Web. In *Proc. 6th International World Wide Web Conference*, pages 391–404, 1997.

[7] Bernard Chazelle, Herbert Edelsbrunner, Leonidas J. Guibas, and Micha Sharir. Diameter, width, closest line pair, and parametric searching. *Discrete and Computational Geometry*, 10:183–196, 1993.

[8] Kenneth L. Clarkson and Peter W. Shor. Applications of random sampling in computational geometry, II. *Discrete and Computational Geometry*, 4:387–421, 1989.

[9] Donald Coppersmith and Shmuel Winograd. Matrix multiplication via arithmetic progressions. In *Proc. 19th ACM Symp. on Theory of Computing*, pages 1–6, 1987.

[10] Richard M. Dudley. Central limit theorems for empirical measures. *Annals of Probability*, 6:899–929, 1978.

[11] Richard M. Dudley. A course on empirical processes. In *Lecture Notes in Mathematics*, number 1097, pages 1–142. Springer-Verlag, 1984.

[12] Ömer Eğecioğlu and Bahman Kalantari. Approximating the diameter of a set of points in the Euclidean space. *Information Processing Letters*, 32:205–211, 1989.

[13] Paul Erdös. On sets of distances of n points. *American Mathematical Monthly*, 53:248–250, 1946.

[14] Paul Erdös. On sets of distances of points in Euclidean spaces. *Magyar Tud. Akad. Mat. Kutató Int. Közl.*, 5:165–169, 1960.

[15] Myron Flickner, Harpreet Sawhney, Wayne Niblack, Jonathan Ashley, Qian Huang, Byron Dom, Monika Gorkani, Jim Hafner, Denis Lee, Dragutin Petkovic, David Steele, and Peter Yanker. Query by image and video content: The QBIC system. *Computer*, 28(9):23–32, 1995.

[16] Peter Frankl and Hiroshi Maehara. The Johnson-Lindenstrauss lemma and the sphericity of some graphs. *Journal of Combinatorial Theory*, Series B 44:355–362, 1988.

[17] Jacob E. Goodman and Joseph O'Rourke, editors. *Handbook of Discrete and Computational Geometry.* CRC Press, 1997.

[18] Peter Gritzmann and Victor Klee. Inner and outer j-radii of convex bodies in finite dimensional normed spaces. *Discrete and Computational Geometry*, 7:255–280, 1992.

[19] Peter Gritzmann and Victor Klee. Computational complexity of inner and outer j-radii of polytopes in finite-dimensional normed spaces. *Mathematical programming*, 59:163–213, 1993.

[20] Peter Gritzmann and Victor Klee. On the complexity of some basic problems in computational convexity: I. Containment problems. *Discrete Mathematics*, 136:129–174, 1994.

[21] Venkat N. Gudivada and Vijay V. Raghavan. Guest editors' introduction: Content-based image retrieval systems. *Computer*, 28(9):18–22, 1995.

[22] John A. Hartigan. *Clustering Algorithms*. Wiley, New York, 1975.

[23] David Haussler and Emo Welzl. ε-Nets and Simplex Range Queries. *Discrete & Computational Geometry*, 2:127–151, 1987.

[24] Charles E. Jacobs, Adam Finkelstein, and David H. Salesin. Fast multiresolution image querying. In R. Cook, editor, *SIGGRAPH 95 Conference Proceedings*, pages 277–286. Addison Wesley, August 1995.

[25] William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. In *Conference in Modern Analysis and Probability*, volume 26 of *Contemporary Mathematics*, pages 189–206. American Mathematical Society, 1984.

[26] Jon M. Kleinberg. Two algorithms for nearest-neighbor search in high dimensions. In *Proc. 29th ACM Symp. on the Theory of Computing*, pages 599–608, 1997.

[27] Donald E. Knuth. *Seminumerical Algorithms*, volume 2 of *The Art of Computer Programming*. Addison-Wesley Longman, 3rd edition, 1997.

[28] Jiří Matoušek and Otfried Schwarzkopf. A deterministic algorithm for the three-dimensional diameter problem. In *Proc. 25th ACM Symp. on the Theory of Computing*, pages 478–484, 1993.

[29] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[30] Ketan Mulmuley. *Computational Geometry. An Introduction Through Randomized Algorithms*. Prentice Hall, 1994.

[31] Joseph O'Rourke and G. Toussaint. Pattern recognition. In [17], pages 797–813. 1997.

[32] Franco P. Preparata and Michael I. Shamos. *Computational Geometry: an Introduction*. Springer-Verlag, New York, 1985.

[33] Edgar A. Ramos. Construction of 1-d lower envelopes and applications. In *Proc. of the 13th Symp. on Computational Geometry*, pages 57–66, 1997.

[34] Stefan Straszewicz. Sur un problème géometrique de P. Erdös. *Bull. Acad. Polon. Sci. Cl. III*, 5:39–40, 1957.

[35] Kenneth J. Supowit. New techniques for some dynamic closest-point and farthest-point problems. In *Proc. 1st ACM-SIAM Symp. on Discrete Algorithms*, pages 84–90, 1990.

[36] Vladimir N. Vapnik and Alexei Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16:264–280, 1971.

[37] Mathukumalli Vidyasagar. *A Theory of Learning and Generalization: with Applications to Neural Networks and Control Systems*. Springer-Verlag, London, 1997.

[38] Andrew C. Yao. Personal communication. Also in "On computing the distance matrix of n points in k dimensions", TR IBM San Jose Research Center, about 1982.

[39] Andrew C. Yao. On constructing minimum spanning trees in k-dimensional spaces and related problems. *SIAM Journal on Computing*, 11(4):721–736, 1982.