Proximity Problems on Line Segments Spanned by Points

Ovidiu Daescu* Jun Luo* David M. Mount†

Abstract

Given a set S of n points in the plane and another point q, we give optimal $O(n \log n)$ time, $O(n)$ space algorithms for finding the closest and farthest line segments (lines) from q among those spanned by the points in S. We also give an $O(n \log n)$ time, $O(n)$ space algorithm to find the k-th closest line and show how to report the k closest lines in $O(n \log n + k)$ time and $O(n)$ space.

1 Introduction

Proximity problems are fundamental to computational geometry and find many applications in related fields, such as graph drawing, collision detection and robot path planning, to name a few. In this paper we study the following problem: *Given a set $S = \{p_1, p_2, \ldots, p_n\}$ of n points in the plane and another point q, find the closest (farthest) line segment from q among the set E of $O(n^2)$ line segments defined by the points in S.*

Our solutions for computing the closest and farthest line segments are related to efficient solutions for the simpler problem in which line segments are replaced by lines, and we also briefly address this problem. Here, we use L to denote the set of $O(n^2)$ lines spanned by S.

Without loss of generality, we assume q is the origin of the coordinate system and the points in $S \cup \{q\}$ are in general position, that is no three of them are collinear.

1.1 Related Work

The problems we study are related to the well known slope selection and distance selection problems [4–7, 9].

Computing the farthest or closest line of L from a point q is closely related to counting the number of lines in L that are intersected by a disk $\mathcal{D}$ centered at q. Using a standard point-line duality transform and parametric search, with a parallel version of the Mount and Netanyahu's [11] algorithm for counting the number of line intersections inside a bounded region, one can compute the k-th closest line from q in $O(n \log^2 n)$ time. See also [2] for an $O(n \log^2 n)$ time solution based on a slightly different parametric search algorithm.

1.2 Results

We present the following results: (1) We give $O(n \log n)$ time, $O(n)$ space algorithms for computing the closest line segment (line) from a point q among the line segments (lines) spanned by S; (2) Similarly, we give $O(n \log n)$ time, $O(n)$ space algorithms for computing the farthest line segment (line) from q among the line segments (lines) spanned by S; (3) We present an algorithm for computing the k-th closest line from q that runs in $O(n \log n)$ time and $O(n)$ space and briefly describe how to report the k closest lines from q in $O(n \log n + k)$ time and $O(n)$ space. All our algorithms are optimal in the algebraic decision tree model.

Throughout the paper a line through two points a and b is denoted as l_{ab}. When a and b correspond to some points $p_i, p_j \in S$ we use the notation l_{ij}.

2 Closest Line Segment From Point

Let $d(q, p_i)$ denote the Euclidean distance from q to p_i, and let $d(q, \overline{p_i p_j})$ denote the minimum Euclidean distance from q to the line segment $\overline{p_i p_j}$, $p_i, p_j \in S$. Let $d_{min} = \min\{d(q, \overline{p_i p_j}) \mid p_i, p_j \in S, p_i \neq p_j\}$ be the distance from q to its closest line segment. Let $p_{min} \in S$ be the closest point to q and let C_{min} be the circle centered at q and of radius $d(p_{min}, q)$. Then, $d_{min} \leq d(p_{min}, q)$ and we can ignore all the line segments of E that do not intersect C_{min}. C_{min} can be computed in $O(n)$ time.

Consider a point $p_i \in S$. Without loss of generality (WLOG), assume l_{qp_i} is vertical and p_i is above q. The line l_{qp_i} separates the points in $S \setminus \{p_i\}$ in two subsets. Let S_i^1 be the set of points to the left of l_{qp_i}, and let S_i^2 be the set of points to the right of l_{qp_i}. Consider the points in S_i^1 (for the points in S_i^2 the analysis is similar). Let $CH(S_i^1)$ denote the convex hull of S_i^1, and consider finding the closest line segment from q among those in the set E_i, defined by p_i and another point in S_i^1.

Lemma 1 *If a line segment $\overline{p_i p_k}$, $p_k \in S_i^1$, intersects C_{min} and all points in $S_i^1 \setminus \{p_k\}$ are above l_{ik} then $\overline{p_i p_k}$ is the closest line segment from q among those in E_i.*

Proof: Since $\overline{p_i p_k}$ intersects C_{min}, $d(q, \overline{p_i p_k}) < d_{min}$. Consider a line segment $\overline{p_i p_j}$, ($p_j \in S_i^1, p_j \neq p_i, p_k$). If $\overline{p_i p_j} \cap C_{min} = \emptyset$, then $d(q, \overline{p_i p_j}) > d_{min} > d(q, \overline{p_i p_k})$. If $\overline{p_i p_j}$ intersects C_{min} let x be the orthogonal projection point of q to $\overline{p_i p_j}$, that is $|\overline{qx}| = d(q, \overline{p_i p_j})$. Then, $\overline{qx}$ must intersect $\overline{p_i p_k}$ at some point y, and $d(q, \overline{p_i p_j}) = d(q, x) > d(q, y) \geq d(q, \overline{p_i p_k})$. Since p_k cannot be inside C_{min} (there are no points of S inside C_{min}) it follows $\overline{p_i p_k}$ is the closest line segment of E_i from q. □

Lemma 2 *(1) If $CH(S_i^1)$ does not intersect C_{min} then either p_i is not an endpoint of the closest line segment*

*Dept. of Computer Science, University of Texas at Dallas, Richardson, TX 75080, USA, {daescu,ljroger}@utdallas.edu.
Daescu's research is supported by NSF grant CCF-0430366.

†Dept. of Computer Science, University of Maryland, College Park, MD 20742, USA, mount@cs.umd.edu. This research was partially supported by NSF grant CCF-0098151.

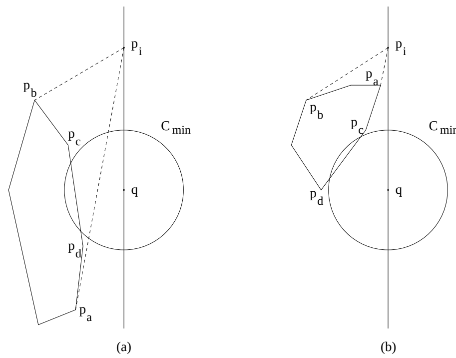

Figure 1: (a) $\overline{p_i p_a}$ intersects C_{min} and (b) $\overline{p_i p_a}$ does not intersect C_{min}.

of E from q or the closest line segment of E_i from q is tangent to $CH(S_i^1)$. (2) If $CH(S_i^1)$ intersects with C_{min}, let e_i be the closest edge to q among those edges of $CH(S_i^1)$ that have nonempty intersection with C_{min}. Then the only possible candidates for the closest line segment from q are e_i and the tangent line segments from p_i to $CH(S_i^1)$.

Proof. For case (1), let the tangent line segments from p_i to $CH(S_i^1)$ have the other endpoints at p_a and p_b, respectively, and assume that p_b is above the line l_{ia}. If the line segment $\overline{p_i p_a}$ does not intersect with C_{min} then, since $d_{min} \leq d(p_{min}, q)$, the closest line segment of E from q is not in E_i. If the line segment $\overline{p_i p_a}$ intersects C_{min} then, according to Lemma 1, $\overline{p_i p_a}$ is the closest line segment of E_i to q, since the endpoints of the other line segments in E_i are above the line l_{ia}.

For part (2), let $\overline{p_c p_d}$ be the edge e_i (see Fig. 1). Let the tangent line segments from p_i to $CH(S_i^1)$ have the other endpoints at p_a and p_b, respectively, and assume that p_b is above the line l_{ia}. If $\overline{p_i p_a}$ intersects C_{min} then all the other line segments in E_i have the endpoint that is different from p_i above the line l_{ia} and according to Lemma 1 $\overline{p_i p_a}$ is the closest line segment of E_i from q (see Fig. 1(a)). If $\overline{p_i p_a}$ does not intersect with C_{min} then all the points of $(S_i^1 \cup \{p_i\}) \setminus \{p_c, p_d\}$ are above the line l_{cd}. It results $\overline{p_c p_d}$ is closer to q than any line segment in E_i and thus we can ignore E_i. □

Then, it follows that the closest line segment $\overline{p_i p_j}$ of E from q is tangent to $CH(S_i^1)$ or $CH(S_i^2)$.

Our algorithm to find the closest line segment of E from q is as follows. First, find the closest point $p_{min} \in S$ from q, let $d_{min} = d(q, p_{min})$ and set the closest line segment to any line segment of E that has p_{min} as one endpoint. Next, sort the points of S around q according to the slopes of the lines l_{qp_i}, $i = 1, 2, \ldots, n$. Let S correspond to this sorted order. Then, **for** $i = 1$ to n, construct (a data structure for) $CH(S_i^1)$ and $CH(S_i^2)$, find the four tangent line segments from p_i to $CH(S_i^1)$ and $CH(S_i^2)$, compute the distance from q to these line segments, and update d_{min} and the closest line segment from q if needed. From the properties above we know that the closest line segment of E from q is one of the tangent line segments from some $p_i \in S$ to $CH(S_i^1)$ or $CH(S_i^2)$. The algorithm checks all these cases and thus it outputs the closest line segment of E from q. In the **for** loop, once $CH(S_1^1)$ and $CH(S_1^2)$ are available, each update for $CH(S_i^1)$ and $CH(S_i^2)$ can be done in $O(\log n)$ time, using the dynamic convex hull data structure in [3]. The tangent line segments from p_i to $CH(S_i^1)$ and $CH(S_i^2)$ can also be computed in $O(\log n)$ time [3]. Then, the **for** loop takes $O(n \log n)$ time. Adding up, the time complexity is $O(n \log n)$ and the space is $O(n)$. The optimality follows from the lower bound proof for lines in [2].

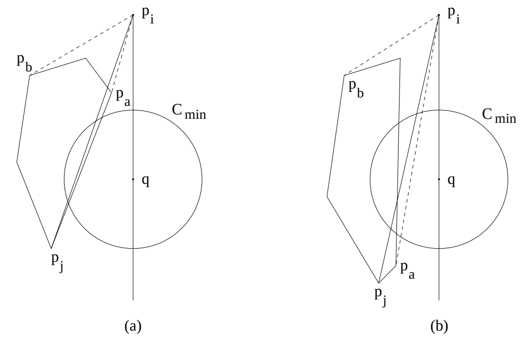

Figure 2: Two cases for the closest line segment $\overline{p_i p_j}$.

Theorem 3 *Given a set S of n points in the plane and another point q, the closest line segment from q among the line segments defined by the points in S can be found in $O(n \log n)$ time and $O(n)$ space, which is optimal.*

Our algorithm for finding the closest line from q spanned by S is simpler and we only sketch the main idea below. Two lines in L define two double-wedges at their intersection point. A double-wedge is *acute* if the angle between its defining lines is no greater than $\pi/2$. Let p_i and p_j be the points defining the closest line to q. It is easy to see that the acute double-wedge defined by l_{ij} and $l_{p_i q}$ does not contain any point of S. The acute double-wedge at p_i separates the points of $S \setminus \{p_i\}$ into two subsets, such that the line l_{ij} is tangent to the convex hull of one of the two subsets. Then, we can use an algorithm similar to that for the closest line segment and compute the closest line from q optimally, in $O(n \log n)$ time and $O(n)$ space.

3 Farthest Line Segment From Point

Let $p_i \in S$ and WLOG assume p_i is vertically above q. The line $l_{p_i q}$ separates $S \setminus \{p_i\}$ into two subsets. Let S_i^1 be the set of points to the left of $l_{p_i q}$, and let S_i^2 be the set of points to the right of $l_{p_i q}$. Consider the points in S_i^1 (for the points in S_i^2 the analysis is similar). Let $CH(S_i^1)$ denote the convex hull of S_i^1, and consider finding the farthest line segment from q among those in the set E_i, defined by p_i and another point in S_i^1. Let

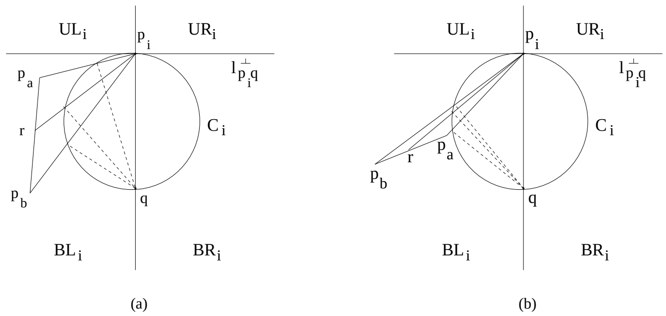

Figure 3: $\overline{p_a p_b}$ has empty intersection with C_i.

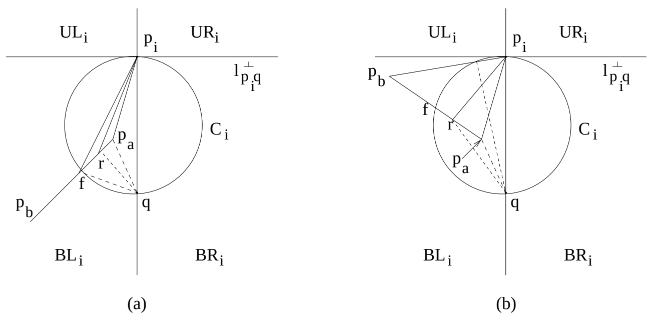

Figure 4: $\overline{p_a p_b}$ intersects C_i at one point.

$l^{\perp}_{p_i q}$ be the line orthogonal to $l_{p_i q}$ at p_i. The lines $l_{p_i q}$ and $l^{\perp}_{p_i q}$ divide the plane into four regions: upper left, upper right, bottom left, and bottom right denoted as UL_i, UR_i, BL_i, and BR_i, respectively. Let C_i be the circle having the line segment $\overline{p_i q}$ as a diameter. Clearly, the farthest line segment from q having p_i as an endpoint has nonempty intersection with the closed disk bounded by C_i. Note also that if there is a point $p_a \in S_i^1$ with p_a in UL_i then $\overline{p_i p_a}$ is a farthest line segment of E_i from q. Then, it suffices to consider the point $p_a \in UL_i$ with $\overline{p_i p_a}$ tangent to $CH(S_i^1)$. This implies that if there is a point of S_i^1 in UL_i then we only need to consider the tangent line segments from q to $CH(S_i^1)$. Then, we focus on the case when $CH(S_i^1) \subset BL_i$.

Lemma 4 *For a segment $\overline{p_a p_b}$, $p_a, p_b \in BL_i$, and any point $r \in \overline{p_a p_b}$, $d(q, \overline{p_i r}) \leq max\{d(q, \overline{p_i p_a}), d(q, \overline{p_i p_b})\}$.*

Proof. There are four cases. (1) $\overline{p_a p_b} \cap C_i = \emptyset$. Without loss of generality, assume the y-coordinate of p_a is no smaller than the y-coordinate of p_b. There are two sub-cases: (a) p_a is above l_{ib} (Fig. 3 (a)). In this case, when r moves from p_b to p_a on $\overline{p_a p_b}$, the intersection of $\overline{p_i r}$ with the orthogonal line from q to $\overline{p_i r}$ moves towards p_i on the semicircle of C_i that is included in BL_i. Then $d(q, \overline{p_i r})$ monotonically increases when r moves from p_b to p_a and thus $d(q, \overline{p_i p_a}) \leq d(q, \overline{p_i r}) \leq d(q, \overline{p_i p_b})$. (b) The point p_a is below the line l_{ib} (Fig. 3 (b)). By an argument similar to the one above, $d(q, \overline{p_i r})$ monotonically decreases when r moves from p_b to p_a. Then, $d(q, \overline{p_i p_b}) \leq d(q, \overline{p_i r}) \leq d(q, \overline{p_i p_a})$. (2) The line segment $\overline{p_a p_b}$ is inside C_i. Then, for any point $r \in \overline{p_a p_b}$, $d(q, \overline{p_i r}) = d(q, r)$. Since $\overline{qr}$ is within the triangle defined by q, p_a and p_b we have that $d(q, \overline{p_i r}) = d(q, r) \leq \max\{d(q, p_a), d(q, p_b)\} = \max\{d(q, \overline{p_i p_a}), d(q, \overline{p_i p_b})\}$. (3) The line segment $\overline{p_a p_b}$ intersects C_i at a point f and one of p_a, p_b is outside C_i and the other one is inside C_i. Without loss of generality, assume p_a is inside C_i and p_b is outside C_i. There are three sub-cases: (a) The y-coordinate of p_a is no smaller than the y-coordinate of p_b and p_a is above the line l_{ib}. Then, for any point r on $\overline{p_a p_b}$, $\overline{p_i r}$ has nonempty intersection with $\overline{q p_a}$. Since $d(q, \overline{p_i p_a}) = d(q, p_a)$ it results that $d(q, \overline{p_i r}) \leq d(q, \overline{p_i p_a})$. (b) The y-coordinate of p_a is no smaller than the y-coordinate of p_b and p_a is below the line l_{ib} (Fig. 4 (a)). Then, $d(q, \overline{p_i f}) \leq d(q, \overline{p_i r}) \leq d(q, \overline{p_i p_b})$ when $r \in \overline{p_b f}$, and $d(q, \overline{p_i r}) \leq \max\{d(q, \overline{p_i p_a}), d(q, \overline{p_i p_f})\}$ when $r \in \overline{f p_a}$, which implies that $d(q, \overline{p_i r}) \leq \max\{d(q, \overline{p_i p_a}), d(q, \overline{p_i p_b})\}$ for any $r \in \overline{p_a p_b}$. (c) The y-coordinate of p_b is larger than the y-coordinate of p_a (Fig. 4 (b)). Note that p_a must be below the line l_{ib}. Using similar arguments as above for $r \in \overline{p_b f}$ and $r \in \overline{f p_a}$ we have $d(q, \overline{p_i r}) \leq \max\{d(q, \overline{p_i p_a}), d(q, \overline{p_i p_b})\}$ for any $r \in \overline{p_a p_b}$. (4) The line segment $\overline{p_a p_b}$ intersects C_i in two points f and g, and thus p_a and p_b are outside C_i (Fig. 5). Without loss of generality, assume the y-coordinate of p_a is no smaller than the y-coordinate of p_b. In this case, $d(q, \overline{p_i p_b}) \leq d(q, \overline{p_i r}) \leq d(q, \overline{p_i g})$ when $r \in \overline{p_b g}$, $d(q, \overline{p_i g}) \leq d(q, \overline{p_i r}) \leq d(q, \overline{p_i f})$ when $r \in \overline{g f}$, and $d(q, \overline{p_i f}) \leq d(q, \overline{p_i r}) \leq d(q, \overline{p_i p_a})$ when $r \in \overline{f p_a}$, which implies that $d(q, \overline{p_i r}) \leq d(q, \overline{p_i p_a})$ for any $r \in \overline{p_a p_b}$. □

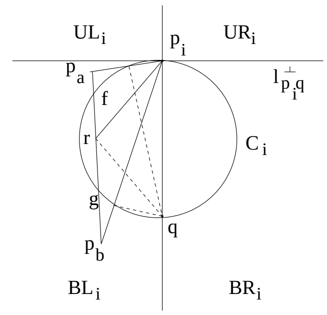

Figure 5: $\overline{p_a p_b}$ intersects C_i at two points.

Lemma 5 *If all points of S_i^1 are in BL_i then for any point $p_a \in S_i^1$ that is inside $CH(S_i^1)$ there is some vertex p_b of $CH(S_i^1)$ such that $d(q, \overline{p_i p_a}) \leq d(q, \overline{p_i p_b})$.*

From Lemma 5 it follows that we only need to consider the vertices of $CH(S_i^1)$. Let p_a and p_b be the two vertices of $CH(S_i^1)$ such that the lines l_{ia} and l_{ib} are tangent to $CH(S_i^1)$. WLOG, assume p_a is above l_{ib}. The line l_{ab} separates $CH(S_i^1)$ into an upper convex hull $UCH(S_i^1)$, which is inside the triangle defined by p_i, p_a, and p_b, and a lower convex hull $LCH(S_i^1)$, which is outside that triangle. Lemma 5 implies that if $CH(S_i^1) \subset BL_i$ then for any vertex p_c of $LCH(S_i^1)$ there is some vertex p_d of $UCH(S_i^1)$ such that $d(q, \overline{p_i p_c}) \le d(q, \overline{p_i p_d})$. Then, we can ignore the vertices of $LCH(S_i^1)$.

Lemma 6 *Let $CH(S_i^1) \subset BL_i$, and let p_j be a vertex of $UCH(S_i^1)$ such that $\overline{p_i p_j}$ is the farthest line segment of E_i from q. Then, either the line l_{ij} is tangent to $CH(S_j^2)$ or E_i can be ignored since it does not contain the farthest line segment of E from q.*

Our algorithm for finding the farthest line segment is similar to the algorithm for finding the closest line segment and we leave it out due to space constraints.

Theorem 7 *Given a set S of n points in the plane and another point q, the farthest line segment from q among the $O(n^2)$ line segments defined by S can be found in $O(n \log n)$ time and $O(n)$ space, which is optimal.*

For the farthest line problem we note that the disk centered at q and tangent to the farthest line of L from q intersects all other lines in L. For a point $p_i \in S$ let $l^{\perp}_{p_i q}$ be the line through p_i and orthogonal to the line $l_{p_i q}$, defined by p_i and q. The line $l^{\perp}_{p_i q}$ partitions $S \setminus \{p_i\}$ into two subsets S_i^1 and S_i^2. Observing that the point $p_j \in S \setminus \{p_i\}$ defining the farthest line from q through p_i is one of the four tangency points corresponding to the tangents from p_i to $CH(S_i^1)$ and $CH(S_i^2)$, we have:

Theorem 8 *Given a set S of n points in the plane and another point $q \notin S$, the farthest line from q defined by two points in S can be found in $O(n \log n)$ time and $O(n)$ space, which is optimal.*

4 Computing the k-th Closest Line from a Point

In this section we briefly discuss how to find the k-th closest line from q in $O(n \log n)$ time and $O(n)$ space. This is optimal in the algebraic decision tree model by the lower bounds on the closest and farthest line problems. Our algorithm is randomized and the running time bound holds with high probability.

The algorithm employs parametric search for the distance d_k from q to the k-th closest line of L. The parametric search maintains a half-open interval $I = [d_{\min}, d_{\max})$ that contains d_k. Let L_I be the set of lines of L whose distance from q lies in this interval. We also maintain the number $n_{\min}$ of lines of L whose distance from q is less than $d_{\min}$. Thus, the problem reduces to finding the $(k - n_{\min})$-th closest line of L_I. Initially $I = [0, \infty)$ and $n_{\min} = 0$. The algorithm contracts this interval through a sequence of stages. We can show that the expected number of stages is a constant.

Consider the problem in its dual setting, by mapping the n points of S into a planar arrangement of n lines. The set of lines of L that lie within a given distance from q are in 1–1 correspondence with the set of arrangement vertices that lie within a region bounded by two branches of a hyperbola [2]. Mount and Netanyahu [11] showed that by an analysis of the order of these $O(n)$ intersection points, it is possible in $O(n \log n)$ time to count, sample, and enumerate the arrangement vertices lying in this region. Then, using a probabilistic argument from [10] we obtain the claimed bounds.

We can use this result to obtain an algorithm that reports the k closest lines from q. Let $\mathcal{D}$ be the disk centered at q and tangent to the k-th closest line. Then, only the $k-1$ closest lines from q have nonempty intersection with the interior of $\mathcal{D}$. In the dual plane, using Balaban's algorithm [1] we can report the intersections of the segments between the corresponding branches of the hyperbola in $O(n \log n + k)$ time and $O(n)$ space.

Theorem 9 *The k closest lines from a point q, among the lines spanned by a set S of n points in the plane, can be found in $O(n \log n + k)$ time and $O(n)$ space.*

5 Conclusion

In this paper we presented optimal $O(n \log n)$ time, $O(n)$ space algorithms for computing the closest and farthest line segments (lines) from a point q, among those spanned by a set S of n points in the plane. We also presented optimal solutions for finding the k-th closest line to q and for reporting the k closest lines to q. Our techniques can also be applied to other problems, such as finding the minimum and maximum area triangles defined by q with $S \setminus q$.

References

[1] I.J. Balaban, An optimal algorithm for finding segments intersections, *Proc. 11th ACM Sympos. Comput. Geom.* (1995) 211–219.

[2] S. Bespamyatnikh and M. Segal, Selecting distances in arrangements of hyperplanes spanned by points, *Journal of Discrete Algorithms* **2** (2004) 333–345.

[3] G.S. Brodal and R. Jacob, Dynamic planar convex hull, *Proc. 43rd Annual Symp. on Foundations of Computer Science* **25** (2002) 617–626.

[4] H. Brönnimann and B. Chazelle, Optimal slope selection via cuttings, *Comput. Geom. Theory Appl.* **10(1)** (1998) 23–29.

[5] T. Chan, On enumerating and selecting distances, *Internat. J. Comput. Geom. Appl.* **11** (2001) 291–304.

[6] R. Cole and J. Salowe and W. Steiger and E. Szemeredi, An optimal-time algorithm for slope selection, *SIAM J. Comput.* **18(4)** (1989) 792–810.

[7] M.B. Dillencourt and D.M. Mount and N.S. Netanyahu, A randomized algorithm for slope selection, *Internat. J. Comput. Geom. Appl.* **2(1)** (1992) 1–27.

[8] P. Erdös and J. Spencer, *Probabilistic Methods in Combinatorics*, (Academic Press, New York, 1974).

[9] M.J. Katz and M. Sharir, Optimal slope selection via expanders, *Inf. Proc. Let.* **47** (1993) 115–122.

[10] J. Matoušek, Randomized optimal algorithm for slope selection, *Inform. Proc. Lett.* **39** (1991) 183–187.

[11] D. Mount and N. Netanyahu, Efficient randomized algorithms for robust estimation of circular arcs and aligned ellipses, *Comput. Geom. Theory Appl.* **19** (2001) 1–33.