Farthest Point from Line Segment Queries*

Ovidiu Daescu† and Robert Serfling ‡

We discuss farthest point from line segment queries in $\mathbb{R}^3$ and related problems. The solution for answering farthest point from line segment queries relies on a solution for the **Farthest Point From Line Problem (FPFL):** Preprocess a set P of n points in $\mathbb{R}^d$, $d \geq 3$, such that for a query line L one can efficiently report the farthest point of P from L. We use $O^*(\cdot)$ to hide $O(\log^{O(1)} n)$ factors.

Theorem 1 *A set P of n points in $\mathbb{R}^d$, $d \geq 3$ a constant, can be preprocessed with $O(s \cdot \log^{O(1)} n)$ space and time such that for a query line L the farthest point of P from L can be found in $O(n \log n / s^{1/\lfloor f(d)/2 \rfloor})$ time, where $n \leq s \leq n^{f(d)/2}$.*

Lemma 1 *Let P be a set of n points in $\mathbb{R}^3$ and let $(q, \mathcal{L}_q, \Delta)$ be a triplet with q a point and $\mathcal{L}_q$ a line through q such that all points in P are within distance Δ from $\mathcal{L}_q$. Let h_q denote the plane orthogonal to $\mathcal{L}_q$ at q. If there is a $p \in P$ such that $|qp| > \Delta$, then the point of P farthest from q in the halfspace defined by h_q and containing p is a vertex of the convex hull of P.*

Proof. Let $P = \{p_1, p_2, \ldots, p_n\}$. Let $CH(P)$ be the convex hull of P, let $C = CH(P) \bigcap h_q$, and consider the part $CH(P, q)$ of $CH(P)$ in one of the two halfspaces defined by h_q. For each tetrahedron $qp_ip_kp_j$, with $p_i, p_k, p_j \in CH(P, q)$, one of p_i, p_k or p_j is the farthest point to q. For each tetrahedron of the form qp_ip_ja, with $p_i, p_j \in CH(P, q)$ and a being a vertex of the boundary ∂C of C, let p_k be the farthest point to q within that tetrahedron (see Fig. 1(a)). Let p'_k be the intersection of the line qp_k with the triangle p_ip_ja. Let $c = ap'_k \bigcap p_ip_j$. Observing that $|qa| \leq \Delta$ and $|\overline{qc}| \leq max\{|\overline{qp_i}|, |\overline{qp_j}|\}$, we have that the farthest point to q satisfying the condition in the lemma could only be one of p_i or p_j. For each tetrahedron of the form qp_iab, with $p_i \in CH(P, q)$ and a, b vertices of ∂C, let p_k be the farthest point to q within that tetrahedron (see Fig. 1(b)). Let p'_k be the intersection of the line qp_k with the triangle p_iab. Let $c = p_ip'_k \bigcap ab$. Clearly, c is within distance Δ from q. Thus, by a similar argument as above on triangle qp_ic, the farthest point from q could only be p_i. □

Our solution for computing farthest points from line segments also uses simplicial partitions. A *simplicial partition* $\Psi(P) = \{(P_1, t_1), (P_2, t_2), \ldots, (P_r, t_r)\}$ for a set P of n points in $\mathbb{R}^3$ is a collection of pairs, where the P_i's are disjoint subsets of P whose union is P, and t_i is a tetrahedron containing P_i, for $i = 1, 2, \ldots, r$. For a given simplicial partition $\Psi(P)$, the *crossing number* of a plane h is the number of tetrahedrons of $\Psi(P)$ that h intersects. The crossing number of $\Psi(P)$ is the maximum crossing number over all possible planes. We say that a simplicial partition $\Psi(P)$ is *fine* if $|P_i| \leq 2n/r$, for every $1 \leq i \leq r$. Matousek [2] proved the following result.

*This research was partially supported by NSF grant CCF-0430366.

†Department of Computer Science, University of Texas at Dallas, Richardson, TX 75080.

‡Department of Mathematical Sciences, University of Texas at Dallas, Richardson, TX 75080.

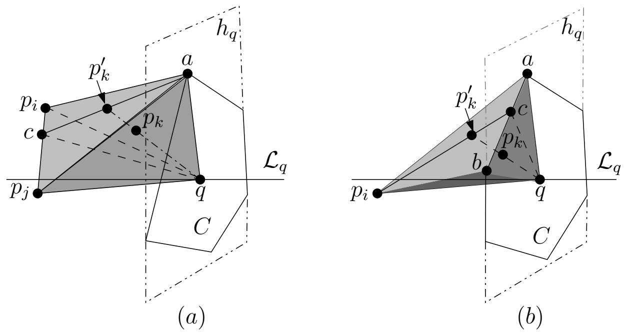

Figure 1: The point p_k farthest from q is a vertex of $CH(P, q)$ if $|qp_k| > \Delta$.

Theorem 2 *[2] For any given set P of n d-dimensional points, $d \geq 2$, and a parameter r, $1 \leq r \leq n/2$, a fine simplicial partition of size r and crossing number $O(r^{1-1/d})$ exists. If r is a constant such a simplicial partition can be constructed in $O(n)$ time with $O(n)$ space.*

A simplicial partition can be used to construct an efficient data structure, called a *partition tree*: the root of the partition tree has r children, each associated with a simplex t_i in a simplicial partition $\Psi(P)$, as well as any secondary information related to the enclosed point set, and being the root of a recursively defined partition tree on the point set that belongs to this node.

Theorem 3 *Given a set P of n points in $\mathbb{R}^3$, in $o(n^{4/3})$ time one can construct a data structure of size $o(n^{4/3})$ such that the farthest point of P from a query line segment s can be reported in $O(n^{2/3+\epsilon})$ time, for arbitrary $\epsilon > 0$.*

Proof. Let L_s be the line supporting s. The farthest point from s is either (i) the farthest point from L_s or (ii) the farthest point $p \in P$ from one of the endpoints of s, with the property that the plane orthogonal to s at that endpoint has s and p on different sides. The point for (i) can be obtained using **FPFL**. This gives us a value Δ such that all points in P are within distance Δ from L_s. The point for (ii) can then be found by performing two queries, each on a data structure associated with $CH(P, q_i)$, where q_i, $i = 1, 2$, is an endpoint of s, and $CH(P, q_i)$ and q_{3-i} are on different sides of the defining plane for $CH(P, q_i)$. Consider the query for $CH(P, q_1)$. The problem is to report the farthest point $p \in P$ to q_1 such that $p \in CH(P, q_1)$ and $|pq_1| > \Delta$, if such a point exists. From Lemma 1, it follows that one should only consider the vertices of $CH(P, q_1)$. We first construct a partition tree based on a fine simplicial partition $\Psi(P)$. Let P_v be the subset of P stored at a node v of the tree, and let t_v be the tetrahedron of $\Psi(P)$ that contains P_v. For each node v, we compute the convex hull CH_v of P_v. We also compute and store a data structure for the subset of P_v corresponding to the vertices of CH_v, that can answer farthest neighbor queries in $\mathbb{R}^3$. In $\mathbb{R}^d$, d a constant, with $O^*(m)$ time preprocessing and $O(m)$ space one can answer farthest neighbor queries in $O(n/m^{1/\lceil d/2 \rceil})$, for any m such that $n \leq m \leq n^{\lceil d/2 \rceil}$ (see [1], page 594). Thus, in $\mathbb{R}^3$, the query time is $O(n/m^{1/2})$. A careful analysis of this data structure shows it requires $O^*(n)$ time preprocessing and $O(n \log n)$ storage. The query time in this data structure is $O(n^{2/3+\epsilon})$, for any constant $\epsilon > 0$. Balancing the space and preprocessing time for **FPFL** to obtain $O(n^{2/3+\epsilon})$ query time, we get $O^*(n^{4/3-4\epsilon})$ preprocessing time and space. $\square$

References

[1] J.E. Goodman and J. O'Rourke. *Handbook of Discrete and Comput. Geometry*, CRC Press, 1997.

[2] J. Matoušek. Efficient partition trees. *Discrete and Computat. Geom.*, 8, 1992.