FINDING EXTREME POINTS IN THREE DIMENSIONS AND SOLVING THE POST-OFFICE PROBLEM IN THE PLANE.

H. EDELSBRUNNER and H.A. MAURER

Institute for Information Processing, Technical University of Graz, Schießstattgasse 4a, A-8010 Graz, Austria

Communicated by K. Mehlhorn
Received 13 May 1984

This paper describes an optimal solution for the following geometric search problem defined for a set \( P \) of \( n \) points in three dimensions: Given a plane \( h \) with all points of \( P \) on one side and a line \( \ell \) in \( h \), determine a point of \( P \) that is hit first when \( h \) is rotated around \( \ell \). The solution takes \( O(n) \) space and \( O(\log n) \) time for a query. By use of geometric transforms, the post-office problem for a finite set of points in two dimensions and certain two-dimensional point location problems are reduced to the former problem and thus also optimally solved.

Keywords: Computational geometry, post-office problem, point location, multi-dimensional searching, data structures, geometric algorithms

1. Introduction

Let \( G \) be a plane graph with \( m \) straight edges in \( \mathbb{E}^2 \). \( G \) induces a subdivision \( S_G \) of \( \mathbb{E}^2 \) into \( O(m) \) (open) regions, (relatively open) edges, and vertices (see Fig. 1). To locate a point \( q \) in \( S_G \) means to determine the region (or edge or vertex) of \( S_G \) that contains \( q \). In Fig. 1, \( q \) is contained in \( R_4 \).

The point location problem requires storing \( S_G \) in some data structure such that later specified points can be located efficiently. During the short history of computational geometry, a good number of methods has been proposed for a solution. However, implementable data structures that are optimal in space and time are found in [12] and [7] only.

One of the earliest and most important motivations to locate points derives from the post-office problem [13]: Let \( S \) denote a finite set of points (also called sites) in the Euclidean plane \( \mathbb{E}^2 \); store \( S \) such that the nearest site to a later specified query point can be determined efficiently. By introducing a special subdivision, called the Voronoi diagram of \( S \), this problem can be reduced to point location in \( \mathbb{E}^2 \) [15]. Fig. 1 actually shows the Voronoi diagram for seven sites indicated by small empty circles.

A new and optimal solution of the post-office problem is described in Section 3.2 of this paper.

* Research reported in this paper was partially supported by the Austrian Fonds zur Förderung der wissenschaftlichen Forschung.
Unlike previous solutions it does not reduce the post-office problem to point location. In some sense, it rather reduces locating a point in a Voronoi diagram to a sequence of distance calculations and comparisons. Section 3.3 extends the method to locate points in more general subdivisions than Voronoi diagrams.

The solution of the post-office problem in Section 3.2 follows by geometric transformation from an optimal solution to the following slightly more general problem in the three-dimensional Euclidean space $\mathbb{E}^3$:

Let $S$ denote a finite set of points in $\mathbb{E}^3$. $S$ is to be stored such that extremal queries defined as follows can be answered efficiently: Let $h$ be a plane such that $S$ is contained in one of the two open halfspaces defined, and let $\ell$ be a directed line (possibly at infinity) in $h$; determine a point of $S$ that is hit first when $h$ is rotated around $\ell$ in counterclockwise direction. If $\ell$ is at infinity, then the rotation degenerates to a translation.

We call this the extremal search problem.

It is easy to see that attention might be restricted to extreme points of $S$, that is, to the vertices of the convex hull of $S$. This suggests to store, somehow, the convex hull of $S$. This suggestion is followed in our paper.

The organization of the paper is as follows: Section 2 gives a time- and space-optimal solution for the extremal search problem as follows: After some general remarks we describe the data structure (Section 2.1), show how to use it (Section 2.2), and how to construct it (Section 2.3). Section 3 describes geometric transforms that yield applications of the results of Section 2 to other problems: Section 3.1 deals with the dual setting, Section 3.2 considers the post-office problem, and Section 3.3 discusses the generalization of the methods to a new and optimal solution for certain point location problems in $\mathbb{E}^3$. Finally, Section 4 reviews the main results and formulates open problems suggested by the examinations of this paper.

2. Searching for extreme points in $\mathbb{E}^3$:

Let $S$ denote a set of $n$ points in $\mathbb{E}^3$. In approaching the extremal search problem for $S$, we make use of a data structure sketched in [5]. For the sake of completeness, we repeat and detail its description. This data structure is based on convex hulls of various sets of points. We start with the introduction of definitions.

Let $P = \text{conv} S$, that is, $P$ is the convex hull of $S$. A point $p$ in $S$ is a vertex of $P$ if $P \neq \text{conv}(S - \{p\})$. If $P$ has $m \leq n$ vertices, then Euler's theorem [11] implies that $P$ has at most $3m - 6$ edges and at most $2m - 4$ faces, for $m \geq 3$. We let $\text{ext} S$ denote the set of extreme points of $S$, that is, of vertices of $P$. A vertex $v$ and an edge $e$ of $P$ are incident if $v$ is an endpoint of $e$, and two vertices $v \neq w$ are adjacent if they are incident upon a common edge. The number of vertices adjacent to $v$ is called the degree $\deg(v)$ of $v$. Finally, a subset $I$ of $\text{ext} S$ is independent (on $P$) if the vertices in $I$ are pairwise non-adjacent.

The data structure of [5] relies on the existence of an independent set $I$ of $P$ with

1. $|I| \geq |\text{ext} S|/c_1$, and
2. $\deg(v) \leq c_2$ for $v$ in $I$,

for positive constants $c_1$ and $c_2$. The existence of such a set $I$ follows from Euler's theorem (see Kirkpatrick [12] who shows it for $c_1 = 24$ and $c_2 = 12$). However, it is not clear what the optimal trade-off between $c_1$ and $c_2$ is. For $c_1 = 24$ and $c_2 = 12$, Kirkpatrick [12] also demonstrates a straightforward $O(n)$ time algorithm to find $I$.

2.1. The data structure

This section presents in detail the data structure called a hierarchic description $\text{Hier}(P)$ of $P$. $\text{Hier}(P)$ is a sequence $P_0 \subseteq \cdots \subseteq P_n$ of convex polytopes that approximate $P$ increasingly well. Let $V_i$ be the set of vertices of $P_i$. We require that

---

1 The impression of counterclockwise rotation is realized when we look in the direction of $\ell$.
2 The convex hull $\text{conv} S$ of $S$ can be defined as the intersection of all closed halfspaces that contain $S$ [9].
Hier(P), some 'vertical' structure that connects consecutive polytopes is needed. Let e be a directed edge in $P_i$ ($0 \leq i \leq k - 1$). If there is an edge $f$ in $P_{i+1}$ with $\text{org } f = \text{org } e$ and $\text{dest } f = \text{dest } e$, then $f$ is the (direct) successor $\text{dsuc } e$ of $e$. Otherwise, there is a unique vertex $v$ in $I_{i+1}$ such that the relative interior of the triangle spanned by $e$ and $v$ does not intersect $P_i$.

We call edge $g$ in $P_{i+1}$ with $\text{org } g = v$ and $\text{dest } g = \text{org } e$ the (indirect) successor $\text{isuc } e$ of $e$. In Fig. 2, all edges except for (I, II) and (II, I) in $P_i$ have direct successors.

Fig. 3 shows the implementation of edge (I, II) of $P_i$ in Fig. 2. Only one pointer has to be reserved for successors since an edge cannot have a direct and an indirect successor at the same time.

We summarize the results of this section.

**Theorem 2.1.** Let $P$ be a convex polytope with $n$ vertices in $E^3$. A hierarchical description of $P$ has height $O(\log n)$ and requires $O(n)$ space.

---

3 A triangle is the convex hull of three noncollinear points.
4 Two such vertices contradict the independence of $I_{i+1}$ and the nonexistence of $v$ contradicts the nonexistence of $f$. 
2.2. Searching in a hierarchical description

This section demonstrates the use of hierarchic descriptions for answering extremal queries. Let S be a finite set of points in $E^3$. Recall that an extremal query specifies a plane h disjoint from con S and a directed line $\ell$ in h, and asks for a point p in S that is extreme w.r.t. h and $\ell$, that is, p is hit first when h is turned around $\ell$. For the time being, we assume that no two points of S are coplanar with $\ell$. It follows that the answer is a unique vertex of P = con S. The degenerate case will be addressed at the end of this section.

We start with clarifying the role of direct and indirect successors of directed edges in Hier(P) consisting of polytopes $P_0, \ldots, P_k$. To this end, let $v_i$ in $P_i$ (for $0 \leq i \leq k$) be the extreme vertex of $P_i$ w.r.t. h and $\ell$. We call edges $e_i$ and $f_i$ extreme (w.r.t. h and $\ell$) if org $e_i = \text{org } f_i = v_i$, and if $P_i$ is projected orthogonally onto a plane normal to $\ell$, then the projections of $e_i$ and $f_i$ appear on the boundary of the resulting polygon. Fig. 4 depicts the polytopes of Fig. 2 together with a query plane h and a directed line $\ell$ in h. The view is taken in the direction of $\ell$ so that h appears as a line and $\ell$ as a point. Vertices and edges extreme w.r.t. h and $\ell$ are drawn heavily.

**Lemma 2.2.** $v_{i+1} = v_i$ or $v_{i+1}$ is the origin of at least one of

- $\text{isuc } e_i$,
- $\text{sym onext } \text{dsuc } e_i$,
- $\text{sym dnext } \text{sym } \text{dsuc } e_i$,
- $\text{isuc } f_i$,
- $\text{sym onext } \text{dsuc } f_i$ and $\text{sym dnext } \text{sym } \text{dsuc } f_i$.

**Proof.** The assertion is clear if $v_{i+1} = v_i$. So assume $v_{i+1} \neq v_i$. Then $v_{i+1}$ is in $P_{i+1}$ and, by convexity of $P_{i+1}$, it is adjacent to $v_i$ and at least one of $\text{dest } e_i$ and $\text{dest } f_i$. \(^5\) Say $v_{i+1}$ is adjacent to $\text{dest } e_i$. If $e_i$ has an indirect successor, then $v_{i+1} = \text{org isuc } e_i$. Otherwise, $v_{i+1}$ is the destination of the next-edges of $\text{dsuc } e_i$, or $\text{sym dsuc } e_i$. □

On an intuitive level of understanding, the search for $v_k$ (which is extreme in S w.r.t. h and $\ell$) starts at $P_0$ and proceeds through the hierarchy to $P_k$ (see Fig. 4, where the search proceeds from right to left). For each $P_{i+1}$, $v_{i+1}$ is computed from $v_i$, $e_i$, and $f_i$ (0 $\leq i < k$) (see Lemma 2.2). If $v_{i+1} \neq v_i$, then $e_{i+1}$ and $f_{i+1}$ are determined in $O(\text{deg}(v_{i+1}))$ time from an arbitrary edge $g$ with org $g = v_{i+1}$ as follows:

For any three consecutive edges $g_0$, $g_1$, $g_2$ emanating from $v_{i+1}$ test whether the plane that contains $g_1$ and is parallel to $\ell$ separates $\text{dest } g_0$ from $\text{dest } g_2$. \(^6\) By convexity of $P_{i+1}$ this is false exactly twice, namely for $g_1 = e_{i+1}$ and for $g_1 = f_{i+1}$.

If, otherwise, $v_{i+1} = v_i$, then $e_{i+1}$ and $f_{i+1}$ can be chosen out of

$$\{ \text{sym isuc } e_i, \text{dsuc } e_i, \text{nodext } \text{dsuc } e_i \}$$

\(^5\) Note that all vertices of $P_k$ are also vertices of $P_{i+1}$.

\(^6\) The sorted sequence of edges with origin $v_{i+1}$ can be computed in $O(\text{deg}(v_{i+1}))$ time using $\text{nodext}$- and $\text{dnext}$-pointers.
Theorem 2.3. Let $S$ be a set of $n$ points in $E^3$. Hie(s $\text{con} S$) allows us to answer an extremal query in $O(n \log n)$ time.

2.3. Constructing a hierarchic description

Let $S$ be a set of $n$ points in $E^3$, and let $P$ be a convex polytope with $n$ vertices in $E^3$. To set up a hierarchic description, the following components are used:

1. Preparata and Hong [14] describe an $O(n \log n)$ time and $O(n)$ space algorithm to construct the convex hull of $S$. It is an easy exercise to convert the representation for convex hulls chosen in [14] in $O(n)$ time to the one described in Section 2.1.

2. $O(n)$ time suffices to find an independent set $I$ of $P$ with $|I| > n/c_1$ and $\deg(v) \leq c_2$, for $v$ in $I$ (see the very beginning of Section 2).

3. Given $P$ (with vertex set $V$) and the vertices in $I$ (as computed in (2)) marked, $\text{con}(V - I)$ can be computed in $O(n)$ time by deleting a vertex of $I$ from $P$ at a time. With nominal extra cost, the $\text{dsuc}$- and $\text{isuc}$-pointers from $\text{con}(V - I)$ to $P$ can be established simultaneously. This implies the following result.

Theorem 2.4. Let $S$ be a set of $n$ points in $E^3$. $O(n \log n)$ time and $O(n)$ space suffice to construct Hie(s $S$).

3. Applications

This section demonstrates solutions of three problems from computational geometry using the structure and algorithms introduced in Section 2. For the three problems, two geometric transforms are used to kind of reduce them to the extremal search problem in $E^3$. We discuss the transforms first and come back to the problems later.

Both geometric transforms are based on the paraboloid $U: z = x^2 + y^2$ and on various of its properties. The dual transform $D$ maps a point $p = (p_x, p_y, p_z)$ of $E^3$ into the plane $D(p) = \{z = 2p_x x + 2p_y y - p_z, \text{ and versa, that is, } D(D(p)) = p\}$. The direction of the normal of $D(p)$ is determined by the x- and y-coordinates of $p$ so the points of a vertical line map into parallel planes, if $p$ is below $U$, that is, $p_z < p_x^2 + p_y^2$, then $D(p) \cap U$ is the set of points $u$ on $U$ such that the tangent plane through $u$ contains $p$. We refer to Fig. 5 for an illustration: The view is taken along the y-axis and the point shown has y-coordinate equal to zero. To simplify the forthcoming discussion, some notation has to be introduced: A plane $h$ is vertical if the z-axis is parallel to $h$. Note that $D(p)$ is nonvertical for all points $p$ in $E^3$. A point $p$ is
above, on, below a nonvertical plane \( h: z = h_x x + h_y y + h_z \) if \( p \) is greater than, equal to, less than \( h_x p_x + h_y p_y + h_z \), respectively. Straightforward calculations yield the following.

**Observation 3.1.** Let \( p \) and \( h \) be a point and a nonvertical plane in \( \mathbb{E}^3 \), respectively. If \( p \) is above, on, below \( h \), then \( D(h) \) is above, on, below \( D(p) \), respectively.

The other transform \( E \) embeds a two-dimensional structure in \( \mathbb{E}^3 \). A (closed) disc

\[
\mathcal{D}(c): (x - c_x)^2 + (y - c_y)^2 \leq c_z^2
\]

in \( \mathbb{E}^2 \) (identified in the natural way with the xy-plane in \( \mathbb{E}^3 \)) maps into the plane

\[
D(c) = \{ (x, y, z) \mid z = 2c_x x + 2c_y y + (c_z^2 - c_x^2 - c_y^2) \}
\]

and vice versa. It is worth mentioning that the center \( (c_x, c_y) \) of \( c \) and \( D(E(c)) \) lie on the same vertical line. Also, the vertical projection of \( E(c) \cap U \) onto the xy-plane is the boundary of \( c \). Let \( E(c)^+ \) and \( E(c)^- \) be the open halfspaces above and below \( E(c) \), respectively. Then \( E(c)^- \cap U \) projects vertically into the interior of \( c \) and \( E(c)^+ \cap U \) into the complement of \( c \). We define \( \overline{p} = D(E(p)) \) for a point \( p \) in the xy-plane, that is, \( \overline{p} \) is the vertical projection of \( p \) onto \( U \). Then we have the following.

**Observation 3.2.** Let \( p \) and \( c \) be a point and a disc in the xy-plane, respectively. \( p \) lies in the interior of \( c \) on the boundary of \( c \) outside of \( c \) if and only if \( \overline{p} \) is in \( E(c)^- \), on \( E(c) \), in \( E(c)^+ \), respectively.

Fig. 5 illustrates the transforms \( D \) and \( E \) one dimension lower. So a plane appears as a line, a circle is represented by a segment, and \( U \) is shown as a parabola.

### 3.1. Penetration search in \( \mathbb{E}^3 \)

Let \( HS \) be a finite set of (open) halfspaces in \( \mathbb{E}^3 \) with nonempty intersection \( I \) and let \( H \) be the set of planes bounding halfspaces in \( HS \). Furthermore, let \( q \) be a point in \( I \) and let \( v \) be a nonzero vector in \( \mathbb{E}^3 \). A plane \( h \in H \) is termed **extreme** (w.r.t. \( q \) and \( v \)) if it is hit first when \( q \) moves in the direction of \( v \). The penetration search problem requires storing \( HS \) such that an extreme plane of a point in \( I \) and a vector can be found efficiently.

Without loss of generality, we assume that each plane in \( H \) is nonvertical and bounds the corresponding halfspace from below. Otherwise, vertical planes are treated separately, and halfspaces bounded from above symmetrically. By Observation 3.1, \( D \) maps a point \( q \) in \( I \) into a plane \( D(p) \) below \( D(H) = \{ D(h) \mid h \in H \} \). Clearly, if \( q \) moves into the direction of \( v \), it describes a line in \( \mathbb{E}^3 \) which implies that plane \( D(q) \) rotates around line \( \ell = D(q) \cap D(q + v) \) by the following argument: Assume that a plane \( D(q + \lambda v) \) does not contain \( \ell \). Then it is parallel to \( \ell \) or intersects it in a point \( r \). By Observation 3.1, there is a unique plane that contains \( q, q + v, \) and \( q + \lambda v \)—a contradiction. The direction of the rotation is also determined by \( v \). By Theorems 2.1, 2.3, and 2.4, we have the following result.

**Theorem 3.3.** There is a data structure that solves the penetration search problem for \( n \) halfspaces in \( \mathbb{E}^3 \) with \( O(\log n) \) query time, \( O(n) \) space, and \( O(n \log n) \) time for construction.


### 3.2. The post-office problem

Let \( S \) be a set of \( n \) points (also called sites) in \( \mathbb{E}^3 \). Recall that the post-office problem requires storing \( S \) such that a nearest site for a given query
point $q$ can be determined efficiently.

To solve the problem, we embed the scenario in $E^2$. $E^2$ is identified with the xy-plane, $\mathbb{S} = \{s \mid s \text{ in } S\}$, and $q$ maps into the plane $E(q)$. By Observation 3.2, $\mathbb{S}$ is above $E(q)$ unless $q$ is in $S$ in which case one point of $\mathbb{S}$ is in $E(q)$ and the others are above. A nearest neighbour $s$ of $q$ can be characterized as follows:

Imagine a disc $c$ with center $q$ and initial radius $c_3 = 0$. When $c_3$ grows continuously, then $s$ is encountered first by $c$ (see Fig. 6).

In the three-dimensional embedding, the growth of $c$ goes along with an upward rise of $E(c)$. By Observation 3.2, $s$ is a first point encountered by $E(c)$. Thus, the hierarchical description of $con \mathbb{S}$ solves the post-office problem with optimal $O(\log n)$ query time, optimal $O(n)$ space, and $O(n \log n)$ time for construction.

The novelty of this solution is not that it is the first optimal solution but that it is not a reduction to point location. Examining the hierarchical description of $\mathbb{S}$ carefully reveals that the following simplification can be performed:

The search in $\text{Hier}(\mathbb{S})$ is guided by special ‘distance calculations’ between points and a plane rotating around a line. For the post-office problem, the line is at infinity, thus the ‘distance’ is proportional to the vertical distance between point $s$ and plane $E(q)$. This, in turn, is the square of the Euclidean distance between site $s$ and query point $q$.

Consequently, the search as required by $\text{Hier}(con \mathbb{S})$ can be interpreted as being guided by a sequence of distance calculations between sites and query point. In fact, the vertical projections of the lower parts of the polytopes in $\text{Hier}(con \mathbb{S})$ are Delaunay triangulations of subsets of $S$. Their duals are Voronoi diagrams (see [8]).

3.3. Point location

Using the dual transform $D$, the solution for the post-office problem in Section 3.2 can be interpreted as follows:

$D(\mathbb{S})$ is a set of planes all tangent to $U$. The intersection of all halfplanes bounded from below by a plane in $D(\mathbb{S})$ is a convex polyhedron $P$. The vertical projection of $P$'s boundary onto the xy-plane is the Voronoi diagram $\text{VOD}(S)$ of $S$. Raising the plane $E(q)$ now translates to lowering the point $q$, and a first point hit by $E(q)$ corresponds to a first plane encountered by $q$.

So the post-office problem is also a special case of the penetration search problem, namely for the vector $v = (0, 0, -1)$. The same picture applies to every point location problem provided the subdivision $S_0$ given can be transformed into a convex polyhedron $P$ such that $S_0$ is the vertical projection of $P$'s boundary.

Unfortunately, not every subdivision of $E^2$ with straight edges can be obtained by projection of the boundary of some convex polyhedron $P$ in $E^3$. For instance, the subdivision in Fig. 7(a) cannot since the extensions of the three unbounded edges do not meet in a point. By similar reasons, the triangulation in Fig. 7(b), which we took from [2], fails to derive from convex polyhedra: The planes supporting $\Delta_1$, $\Delta_2$, $\Delta_3$ (if there were any) intersect in the shaded angles. Since these angles have no common point, the intersections of the planes have neither which implies that the planes do not exist. We refer to [3,17,18,16] for literature on cell complexes and projections of convex polyhedra.

Nevertheless, many naturally arising subdivi-

---

The Voronoi diagram of $S$ associates each site $s$ with the region $V(s) = \{p \in E^2 \mid d(p, s) < d(p, t), t \in S - \{s\}\}$. 

Fig. 6. Growing disc identifies nearest neighbour.
sions of $E^2$ can be obtained by projecting the boundaries of convex polyhedra. For instance, higher-order Voronoi diagrams belong to this class [1].

4. Discussion

We start with a review of what we consider the significant contributions of this paper. It offers a unified view of four problems in computational geometry: New and optimal solutions for the post-office problem and certain point location problems in $E^2$ are derived via geometric transforms from an optimal solution for the extremal search problem in $E^3$: Find the first point hit by a rotating or sweeping plane. Also, the dual problem of finding the penetration of a moving point and a convex polytope in $E^3$ is thus solved. The solution for the post-office problem is the first one which is optimal and does not reduce it to point location. In some sense even, the method given for point location is a reduction to a sequence of sort of distance calculations. In this context it is interesting to note that the hierarchic description, that is, the central data structure, is essentially a DAG—as are the other optimal solutions for point location given in literature.

As an open problem we point out the investigation of the trade-off between constants $c_1$ and $c_2$ inherent in the definition of a hierarchic description of a polytope. Results on this trade-off might answer the question of the practicality of our results.

Furthermore, no complexity results are available for deciding whether a given subdivision of $E^2$ can be obtained by projecting the boundary of a three-dimensional convex polyhedron.

References
