A LINEAR ALGORITHM FOR DETERMINING THE SEPARATION OF CONVEX POLYHEDRA

by

David P. Dobkin & David G. Kirkpatrick

TECHNICAL REPORT 83-5

A LINEAR ALGORITHM FOR DETERMINING THE SEPARATION OF CONVEX POLYHEDRA

David P. Dobkin
Electrical Engineering and Computer Science Department
Princeton University
Princeton, N.J. 08544

David G. Kirkpatrick
Department of Computer Science
University of British Columbia
Vancouver, B.C. V6T 1W5

Abstract

The separation of two convex polyhedra is defined to be the minimum distance from a point (not necessarily an extreme point) of one to a point of the other. We present a linear algorithm for constructing a pair of points that realize the separation of two convex polyhedra in three dimensions. Our algorithm is based on a simple hierarchical description of polyhedra that is of interest in its own right.

Our result provides a linear algorithm for detecting the intersection of convex polyhedra. Separation and intersection detection algorithms have applications in clustering, the intersection of half-spaces, linear programming, and robotics.

1. Introduction

Let P and Q denote two convex polyhedra in R^k (real k-dimensional space). The separation of P and Q, denoted $\sigma(P,Q)$, is given by

$$\sigma(P,Q) = \min\{|p-q| \ |p \in P \text{ and } q \in Q\}^{/1/}$$

A pair of points (p,q) where $p \in P$ and $q \in Q$ is said to realize the separation of P and Q if $|p-q| = \sigma(P,Q)$. It is clear that the separation of convex polyhedra is realized only by facial points (i.e. points on the surface) of P and Q, but it is not always realized by the extreme points (or vertices) of P and Q. Even in three dimensions there exist convex polyhedra whose separation is uniquely realized by a pair of non-extreme points.

The problem of determining the separation of two convex polyhedra in R^k (hereafter the k-d separation problem) is a clear generalization of the problem of detecting the intersection of two convex polyhedra (the k-d intersection detection problem). The latter asks only for a witness - a point in common if the intersection is non-empty, or a separating hyperplane otherwise. This, of course, is a very special case of the problem of explicitly constructing the intersection (the k-d intersection construction problem). Intuitively, a pair of points that realizes the separation of two polyhedra encodes what could be considered the "best" separating hyperplane (i.e. the thickest separating "hyperslab".

The 2-d intersection construction problem is one of a number of geometric problems discussed by Shamos [15]. He presents an O(n) algorithm for constructing the intersection of arbitrary convex n-gons in the plane. Shamos notes that this can be used to answer the question of (linear) separability, but he does not address the question of finding the separation (or "best" linear separators).

Muller and Preparata [11] address the 3-d intersection construction problem. Their O(n log n) algorithm has two phases both of which are Ω(n log n) in the worst case. The first phase is what we have called 3-d intersection detection. The second phase works with a witness to the intersection and proceeds to construct the intersection by geometric dualization. Muller and Preparata note that their algorithm can be applied to the three dimensional linear separability problem. Their construction, however, does not lend itself to the determination of the (non-zero) separation of two polyhedra.

We have recently become aware of the work of M. Dyer [5,6] on the 3-d intersection detection problem. A new O(n log n) algorithm for intersection detection is presented in [5]; the bound is reduced to O(n) by a subsequent algorithm presented in [6]. Dyer, like Muller and Preparata, shows how to minimize the "vertical distance" between two given polyhedra. Unfortunately, this provides only the "vertical" separation and hence only an upper bound on the true separation. In this paper, we present an O(n) algorithm for the 3-d separation problem. An O(n) algorithm for the 3-d intersection detection problem/2/ is an immediate corollary.

It is natural to consider the effect of preprocessing on intersection detection and related problems. This issue was first raised by Shamos [15,16] and was studied in detail by Chazelle and Dobkin who present sublinear (in fact, polylogarithmic) 2-d and 3-d intersection detection algorithms [2]. Dobkin and Kirkpatrick [3] employ a projective technique to extend and unify the results of Chazelle and Dobkin [2]. Schwartz [14] and more recently Edelsbrunner [7] and Chen and Wang [1] have studied the preprocessed version of the separation problem in 2-d. In a companion paper [4], we show how the hierarchical representation introduced in this paper can be employed in this related context, extending and unifying the results of [3] and the present paper. (In particular, we present an $O(\log n)$ solution to the 2-d separation problem and an $O(\log^2 n)$ solution to the 3-d separation problem both of which use only O(n) preprocessing.)

In the next two sections we discuss the representation of convex polyhedra in two and three dimensions. In particular, Section 3 describes a new hierarchical representation of polyhedra, and sets out some of its important properties. Our separation algorithm is developed in Section 4. Section 5 presents a brief summary and discussion.

2. Definitions and Representations

2.1 Basic definitions

We now set out definitions of polyhedra and some of their important properties. See, for example, [8] for a more detailed treatment.

A (convex) polyhedron in R^k is defined to be the intersection of some finite number of half-spaces in R^k. Bounded polyhedra are called polytopes. (A polytope can be defined equivalently as the convex hull of a finite point set in R^k). The dimension of a polyhedron P, denoted dim P, is the dimension of the smallest flat (affine subspace) containing the polyhedron. A polyhedron (respectively polytope), P is called a d-polyhedron (respectively d-polytope) if dim P=d.

If $a \in R^k - \{0\}$ and $c \in R$ then the set $H(a,c) = \{x \in R^d \mid \langle x,a \rangle = c\}$/3/ is called a hyperplane in R^k. A hyperplane $H(a,c)$ defines two closed half-spaces $H^+(a,c) = \{x \in R^d \mid \langle x,a \rangle \geq c\}$ and $H^-(a,c) = \{x \in R^d \mid \langle x,a \rangle \leq c\}$. We say that a hyperplane $H(a,c)$ supports a polyhedron P if $H(a,c) \cap P \neq \phi$ and $P \subseteq H^+(a,c)$. If $H(a,c)$ is any hyperplane supporting P then $P \cap H(a,c)$ is said to be a face of P. The faces of dimension (dim P)-1 are called facets; those of dimension 1 (respectively 0) are called edges (respectively vertices) of P.

The _1-skeleton_ (hereafter simply _skeleton_) of a polytope P is the graph whose vertices (respectively edges) are the vertices (respectively edges) of P under the obvious incidence relation. Hereafter, we will often not distinguish between P and its skeleton referring, for example, to the degree of a vertex v in P (denoted _deg(v,P)_) rather than in the skeleton of P.

We find it convenient to refer to polytopes with fewer than some fixed constant number of vertices as _elementary_ polytopes. For the purposes of this paper the fixed constant can be taken as 11 (see lemma 3.1).

The _extreme points_ of a polytope P constitute the smallest set of points whose convex hull is P. A d-polytope which is the convex hull of precisely d+1 points is called a _d-simplex_ (a 2-simplex is a _triangle_ and a 3-simplex is a _tetrahedron_). If B is a (d-1)-polytope and c is a point (not on the (d-1)-flat of B) then the convex hull of $B \cup \{c\}$ is called a _d-pyramid_ with basis B and apex c.

2.2 Initial Representations of 2- and 3-polytopes

Convex polytopes admit numerous different representations not all of which are necessarily equivalent in the sense that they can be transformed from one to the other in time linear in the size of the representation. (For example, the extreme points uniquely characterize a polytope but super-linear time may be required to construct the full facial graph/4/ of a polytope from its extreme points. In fact, in dimensions higher than three the facial graph may demand a non-linear description [8]). Thus, in discussing algorithms, especially linear algorithms, for the manipulation of polytopes, we are obliged to present and defend a choice of a standard (initial) representation.

The standard representation for convex polygons in 2-space seems to be so obvious as to hardly deserve mention. An ordered ring of vertices (or edges) is clearly linear in the number of vertices and is the usual endproduct of conventional convex hulls algorithms. The efficient dynamic maintenance of convex hulls [12] seems to require additional structure, tailored to this particular application.

In three dimensions, one exploits the fact that the surface of a 3-polytope is topologically equivalent to a bounded planar subdivision. Though many different choices of representation are possible, the most natural choices, including those produced by the most efficient convex hull algorithms [13], are all linearly equivalent. Muller and Preparata [11] make precisely this point in proposing their doubly connected edge list, DCEL, representation for planar subdivisions (equivalently, 3-polyhedra). A key assumption in the DCEL and related representations is that the adjacency or incidence information is ordered - that is each vertex has associated with it an ordered (say clockwise) list of incident edges and each face has associated with it an ordered list of bounding edges. We will adopt the DCEL representation as the standard initial representation for 3-polytopes. However, since the conversion to an alternate representation is central to our algorithm, we should note that the only assumption that we make of our initial representation R is that it be possible to convert R into a representation of a triangulation (i.e. triangular refinement) of the associated planar subdivision, in linear time. Obviously, having an ordered list of boundary edges for each facet makes this possible, since each facet (a convex polygon) can be easily triangulated in linear time.

The representation of (potentially unbounded) 3-polyhedra presents only minor technical difficulties. The problem lies in the non-uniformity imposed by the presence of both bounded and unbounded faces. It is possible to enforce uniformity by either taking a suitable projection (e.g. to a sphere) or by introducing pseudo-vertices at infinity. These techniques serve only to cloud the essential ideas in both our representations and algorithm. Hence, we will proceed under the general assumption that the polyhedra that we encounter are bounded, reserving only the occasional remark for the most general case.

3. Hierarchical Representations - Definition, Construction and Properties

Our algorithm is based on a hierarchical representation of polytopes. Informally, our hierarchical representation of a polytope P may be viewed as a finite sequence of progressively finer polytopal approximations to P, each approximation containing its predecessor. Obviously, if such approximations of two polytopes intersect then the polytopes must themselves intersect. While the converse is not true, we are able to exploit information on the non-intersection of two polytopal approximations to test more rapidly for the intersection of their successors in the approximation hierarchies.

In the remainder of this section, we describe our hierarchical representation more formally. We discuss the efficient construction of hierarchical representations in two and three dimensions and present properties of the representations that will be exploited in the separation algorithms that follow.

3.1 Hierarchical representations of d-polytopes

Let P be a d-polytope with vertex set V(P). A sequence of polytopes, $H(P) = P_1, \ldots, P_k$, is said to be a *hierarchical representation* of P if

i) $P_1 = P$ and P_k is a d-simplex;

ii) $P_{i+1} \subset P_i$, for $1 \leq i < k$;

iii) $V(P_{i+1}) \subset V(P_i)$; and

iv) the vertices of $V(P_i)-V(P_{i+1})$ form an independent set (i.e. are non-adjacent) in P_i.

We call k the <u>height</u> of H(P). The <u>size</u> of H(P), denoted $|H(P)|$, is given by

$$|H(P)| = \sum_{i=1}^{k} |P_i|$$

where $|P_i|$, the size of the polytope P_i, is defined in general to be the number of faces of P_i, of dimensions 0 through d-1. Size is meant to be a measure of the storage requirements of both polytopes and our polytopal hierarchies. In dimensions two and three it is equivalent, up to constant factors, to define the size of P_i to be the number of vertices of P_i.

The <u>degree</u> of H(P) is given by

$$\max_i \max_{v \in V(P_i)-V(P_{i+1})} \deg(v, P_i)$$

This reflects the maximum local change that takes place in moving from some element of H(P) to its successor.

<u>3.2 Construction of 2-d and 3-d hierarchies</u>

The definition of hierarchical representation suggests an approach to its construction, namely at each phase identify and remove an independent set of vertices, and let the next approximation be the convex hull of the remaining vertices.

As we shall see, in two and three dimensions we are guaranteed of the existence of hierarchical representations of low degree. Furthermore, provided each polytope in the hierarchy is represented with its surface fully triangulated (which is automatic if the vertices are in general position), then a low degree hierarchical representation can be constructed very efficiently.

Let b be any fixed positive integer. The following algorithm - when it terminates - it guaranteed to produce a hierarchical representation of the polytope P with degree at most b.

<u>Algorithm</u> <u>A</u> (constructs a hierarchical representation of P)

    <u>input</u> P

    P_1 <- P; i <- 1

    <u>while</u> P_i is not a simplex <u>do</u> <u>begin</u>

        S <- any maximal independent set among the vertices of degree at most b in P_i

        P_{i+1} <- hull($P_i \setminus S$)

        i <- i+1 <u>end</u>

Algorithm A clearly terminates provided every d-polytope is guaranteed to have at least one vertex of degree at most b. The reason for specializing our discussion to two and three dimensions at this point should become clear with the following.

Lemma 3.1. There exist constants b_0 and $c<1$ such that for all $b \geq b_0$ and all polytopes P in 2 or 3 dimensions, Algorithm A produces a hierarchical representation of P, $P_1,\ldots,P_k$, satisfying $|P_{i+1}| < c|P_i|$, $1 \leq i < k$.

Proof. Since the skeleton of P is planar, it is an immediate consequence of Euler's formula (cf. [9]) that any maximal independent subset S of the vertices of degree ≤ 11 in P, has size $|S| \geq |P|/24$. The result follows by choosing $b_0 = 11$ and $c = 23/24$./5/ □

Corollary 3.2. For all sufficiently large b, Algorithm A produces a hierarchical representation of an arbitrary 2- or 3-polytope P with degree d, height $O(\log(|P|))$, and size $O(|P|)$.

Proof. The sequence of vertex removals forms a decreasing geometric series. □

Theorem 3.3. For every 2- or 3-polytope P there exists a hierarchical representation of constant degree, $O(\log(|P|))$ height, and $O(|P|)$ size, that can be constructed from a standard representation of P in $O(|P|)$ time.

Proof. By Corollary 3.2, it suffices to show that each P_{i+1} in the hierarchy constructed by Algorithm A can be formed from its predecessor P_i in $O(|P_i|)$ steps. Obviously, a maximal independent set S of vertices of bounded degree in P_i can be constructed in $O(|P_i|)$ steps. The convex hull of $P_i \setminus S$ can be computed from P_i in $O(|S|)$ steps since each vertex in S has degree bounded by some constant (and hence, since the P_i was assumed to be fully triangulated, its removal leaves a neighbourhood of constant size). $\square$

3.3 Localization of change

The attribute of hierarchical representations that subsequent polytopal approximations differ only on an independent set of vertices was certainly exploited in the design of an efficient algorithm for the construction of hierarchical representations. This attribute also gives rise to the following important property of hierarchical representations, that is exploited in our separation algorithm.

Lemma 3.4. Let $P_1,\ldots,P_k$ be a hierarchical representation of some d-polytope P and let H be any hyperplane in R^d such that $P_{i+1} \subset H^+$, for some $i \geq 1$. Then either

i) $P_i \subset H^+$; or

ii) there is a unique vertex $v \in V(P_i)$ such that $v \in H^-$.

Proof. Suppose that $P_{i+1} \subset H^+$ and $v \in V(P_i)$ satisfies $v \in H^-$. Then, by the definition of hierarchical representation all of the neighbours of v in P_i are also vertices of P_{i+1} and hence belong to H^+. Hence, by the convexity of P_i, v is the unique vertex of P_i in H^-. $\square$

Corollary 3.5. Let P be any 2- or 3-polytope and let $P_1,\ldots,P_k$

be a hierarchical representation of P of degree at most d. Let H be any hyperplane such that $P_{i+1} \subset H^+$, for some $i \geq 1$. Then either,

i) $P_i \subset H^+$; or

ii) $P_i \cap H^-$ is a pyramid whose apex has degree at most d.

Remark. In this entire section we have made no essential use of the boundedness of polytopes. It is a straightforward though detailed exercise to modify the definition of hierarchical representation (allowing P_k to be unbounded and generalizing the notion of vertex) and Algorithm A, so that all of the subsequent results carry over to the case of arbitrary convex polyhedra.

4. Determining the Separation of 3-polyhedra

Let P and Q denote arbitrary 3-polytopes. We start by noting the most naive algorithm for determining the separation of P and Q.

Lemma 4.1. If P and Q are arbitrary 3-polytopes then $\sigma(P,Q)$ can be determined in $O(|P|.|Q|)$ time.

Proof. Determine the separation of each face, edge and vertex of P with each face, edge, and vertex of Q and minimize. □

Corollary 4.2. If P is an elementary 3-polytope then $\sigma(P,Q)$ can be determined in $O(|Q|)$ time.

Our algorithm for determining the separation of arbitrary polytopes makes use of the hierarchical representation described in Section 3 to reduce the complete separation problem to a sequence of progressively larger elementary separation problems. The central lemma below shows how we can step through the hierarchical representation exploiting the separation knowledge of each preceeding step.

Let $P_1,\ldots,P_r$ (respectively, $Q_1,\ldots,Q_s$) be a hierarchical representation of degree $\leq d$ of the 3-polytope P (respectively Q).

Lemma 4.3. Suppose the point pair (p_{i+1},q_{i+1}) realizes $\sigma(P_{i+1},Q_{i+1})$. Then a pair (p_i,q_i) realizing $\sigma(P_i,Q_i)$ can be determined in $O(|P_i|+|Q_i|)$ steps.

Proof. If $p_{i+1} = q_{i+1}$ then it suffices to choose $p_i = p_{i+1}$ and $q_i = q_{i+1}$. Alternatively, let H_p be the hyperplane normal to the line segment $\overline{p_{i+1},q_{i+1}}$ supporting P_{i+1} and let H_q be the hyperplane normal to the line segment $\overline{p_{i+1},q_{i+1}}$ supporting Q_{j+1}. Then

$$P_i = (P_i \cap H_p^+) \cup (P_i \cap H_p^-),$$
$$Q_i = (Q_i \cap H_q^+) \cup (Q_i \cap H_q^-)$$

and $$\sigma(P_i,Q_i) = \min \begin{cases} \sigma(P_i \cap H_p^+,\ Q_i \cap H_q^+) \\ \sigma(P_i \cap H_p^-,\ Q_i) \\ \sigma(P_i,\ Q_i \cap H_q^-) \end{cases}$$

But since H_p and H_q are parallel, $\sigma(P_i \cap H_p^+,\ Q_i \cap H_q^+)$ is realized by the pair (p_{i+1},q_{i+1}) and hence

$$\sigma(P_i,Q_i) = \min \begin{cases} \sigma(P_{i+1},\ Q_{i+1}) \\ \sigma(P_i \cap H_p^-,\ Q_i) \\ \sigma(P_i,\ Q_i \cap H_q^-) \end{cases}$$

However, by Corollary 3.5, $P_i \cap H_p^-$ and $Q_i \cap H_q^-$ are both elementary polytopes and hence $\sigma(P_i \cap H_p^-,\ Q_i)$ (respectively, $\sigma(P_i,\ Q_i \cap H_q^-)$) can be determined in $O(|Q_i|)$ (respectively, $O(|P_i|)$) steps, by corollary 4.2. Thus $\sigma(P_i,Q_i)$, and its realization, can be determined in $O(|P_i|+|Q_i|)$ steps in total. $\square$

Lemma 4.3 suggests the following algorithm for determining the separation of two polytopes P and Q.

Algorithm B (determine the separation $\sigma(P,Q)$ of polytopes P and Q)

```
construct hierarchical representations P_1,...,P_r and
      Q_1,...,Q_s, using Algorithm A
i <- min(r,s)
(p,q) <- closest_pair(P_i,Q_i)
while p≠q and i≥1 do begin
      (p,q) <- closest_pair(P_i,Q_i)
      i <- i-1 end
σ(P,Q) <- |p-q|
```

Theorem 4.4. Algorithm B correctly determines $\sigma(P,Q)$, and its realization, in $O(|P|+|Q|)$ steps.

Proof. By Theorem 3.3, the hierarchical representations of P and Q can be constructed in $O(|P|+|Q|)$ steps. Finding the initial closest pair takes $O(|P_m|\cdot|Q_m|)$ steps, where $m = \min(r,s)$. However since both P_r and P_s are elementary, this is just $O(|P_m|+|Q_m|)$. In light of Lemma 4.3, it is clear that the loop uses $O(\sum_{l=1}^{m} (|P_i|+|Q_i|))$ steps in total. Thus the entire algorithm uses $O(|P|+|Q|) + O(\sum_{i=1}^{r} |P_i| + \sum_{j=1}^{s} |Q_j|)$ steps, which, by Theorem 3.3, is $O(|P|+|Q|)$. $\square$

Remark. In keeping with our earlier remarks, we note that Algorithm B and Theorem 4.4 carry over directly to unbounded polyhedra, using the modified representations discussed in Section 3.

5. Discussion and Summary

To this point we have maintained the assumption that "distance" means "Euclidean distance". However, our results depend very little on this specific choice of metric.

Let d be any distance metric on R^k. We can define the d-separation of two polyhedra P and Q as

$$\sigma_d(P,Q) = \min\{d(p,q)\,|\,p\varepsilon P \text{ and } q\varepsilon Q\}.$$

Let $|x|_d$ denote $d(x,0)$. A distance metric d is said to be invariant under translation if $d(x,y) = d(x+z,y+z)$, for all $x,y,z\varepsilon R^k$. For any such metric $d(x,y) = |x-y|_d$. We say that a metric is scale respecting if $|\lambda x|<|x|$, for all $\lambda<1$, $\lambda\varepsilon R$. For metrics d that are scale respecting the d-separation of two polyhedra is realized by facial points of the polyhedra.

Inspection of Algorithm B and Theorem 4.4 reveals that we need only an analog of Lemma 4.3 to generalize our result to other metrics. Lemma 4.3, in turn, depends only on the property that any pair of points realizing the separation of two polytopes must admit parallel planes of support. In the Euclidean case the planes are normal to the vector joining the given pair, but this property is not exploited.

Lemma 5.1.

If the distance metric d on R^2 is invariant under translation and is scale respecting then any pair of points realizing the d-separation of two polytopes P and Q admit parallel planes of support.

Proof.

Suppose that the pair p,q realizes the d-separation of P and Q (i.e. $\sigma_d(P,Q) = |p-q|_d$). Consider the polytope Q' formed by translating Q by the vector p-q. The polytope P intersects the polytope Q' only in facial points (including P), since if some point $x \epsilon P$ is interior to Q' then y = x+q-p is interior to Q and $|x-y|_d = |p-q|_d$, contradicting the scale respecting property of d. Thus, there exists a plane X separating P and Q' passing through p. This plane and its translation X-(p-q) support P and Q at points p and q respectively. □

By the discussion preceeding Lemma 5.1, we are now able to conclude the following generalization of Theorem 4.4.

Theorem 5.2.

Algorithm B correctly determines $\sigma_d(P,Q)$, and its realization, in $O(|P|+|Q|)$ steps, for any pair of polytopes P and Q and any translation invariant and scale respecting metric d.

An immediate corollary of Theorem 4.4 is the existence of a linear algorithm for testing the intersection of polyhedra. It remains an open problem to determine if the actual construction of the intersection can be carried out within this same time bound.

It should be noted that our algorithm provides not only a separating hyperplane (if one exists) for two polyhedra (that may arise, for example, as the convex hulls of two point sets), but in fact it specifies the "thickest" possible such separating slab. This provides an added measure of disjointness that is of use in pattern recognition and clustering algorithms. It should also be useful in determining collision-free paths in robotics applications.

Our algorithm is based on a new space efficient representation for polyhedral objects. It should be viewed as yet another application of hierarchical representations in the context of computational geometry. The algorithm is very similar in spirit to the subdivision search algorithm of Kirkpatrick [10] and the preprocessed polyhedral intersection detection algorithms of Dobkin and Kirkpatrick [3]. We anticipate that this approach to the design of geometric algorithms will have other applications as well.

Footnotes

/1/ $|p-q|$ denotes the Euclidean distance between p and q. The sensitivity to the distance metric is discussed in Section 5.

/2/ This result is of interest in its own right. The algorithm, which was discovered independently of that of Dyer [ ], is fundamentally different to Dyers algorithm in its approach.

/3/ $\langle x,y \rangle$ denotes the inner product of vectors x and y.

/4/ That is, the generalized skeleton that records the incidence structure of faces of all dimensions.

/5/ We make no attempt to optimize b_0 and c here.

REFERENCES

1. Chin, F. and Wang, C.A., "Optimal algorithms for the intersection and the minimum distance problems between planar polygons", Technical Report TR 82-8, Department of Computer Science, University of Alberta, Edmonton, Alberta, Canada, August 1982.

2. Chazelle, B. and Dobkin, D.P., "Detection is easier than computation", Proc. 12th Annual ACM Symposium on Theory of Computing, 1980, pp. 146-153.

3. Dobkin, D.P. and Kirkpatrick, D.G., "Fast detection of polyhedral intersection", Theoretical Computer Science, to appear.

4. Dobkin, D.P. and Kirkpatrick, D.G., "Fast algorithms for preprocessed polyhedral intersection detection", in preparation.

5. Dyer, M.E., "A simplified O(n log n) algorithm for the intersection of 3-polyhedra", unpublished manuscript.

6. Dyer, M.E., "An O(n) algorithm for locating a point in the intersection of polyhedra", unpublished manuscript.

7. Edelsbrunner, H., "On computing the extreme distances between two convex polygons", Report 96, Institut fur Informationsverarbeitung, Technical Univ. Graz, Graz, Austria, August 1982.

8. Grunbaum, B., Convex Polytopes, Wiley Interscience, New York, 1967.

9. Harary, F., Graph Theory, Addison-Wesley, Reading, MA, 1969.

10. Kirkpatrick, D.G., "Optimal search in planar subdivisions", SIAM J. Comput. 12, 1 (Jan. 1983), pp. 28-35.

11. Muller, D. and Preparata, F., "Finding the intersection of two convex polyhedra", Theoretical Computer Science 7 (1978), pp. 217-236.

12. Overmars, M.H. and van Leeuwen, J., "Maintenance of configurations in the plane", JCSS 23, 2 (1981), pp. 166-204.

13. Preparata, F.P. and Hong, S.J., Convex hulls of finite sets in two and three dimensions, Commun. ACM 20 (1977), pp. 87-93.

14. Schwartz, J.T., "Finding the minimum distance between two

convex polygons", Information Processing Letters, 13 (1981), pp. 168-170.

15. Shamos, M.I., "Geometric complexity", Proc. 7th Annual ACM Symposium Theory of Computing, 1975, pp. 224-233.

16. Shamos, M.I. and Hoey, D., "Geometric intersection problems", Proc. 17th Annual IEEE FOCS Conference, 1976, pp. 208-215.