A DUAL APPROACH TO DETECT POLYHEDRAL INTERSECTIONS IN ARBITRARY DIMENSIONS

by

Oliver Gunther and Eugene Wong

Memorandum No. UCB/ERL M86/88

9 December 1986

COVER PAGE

A DUAL APPROACH TO DETECT POLYHEDRAL INTERSECTIONS IN ARBITRARY DIMENSIONS

by

Oliver Gunther and Eugene Wong

Memorandum No. UCB/ERL M86/88

9 December 1986

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

TITLE PAGE

A DUAL APPROACH TO DETECT POLYHEDRAL INTERSECTIONS IN ARBITRARY DIMENSIONS

by

Oliver Gunther and Eugene Wong

Memorandum No. UCB/ERL M86/88

9 December 1986

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

A Dual Approach to Detect Polyhedral Intersections in Arbitrary Dimensions

Oliver Gunther and Eugene Wong
EECS Department, 231 Cory Hall
University of California
Berkeley CA 94720

1. Introduction

Detecting and computing intersections is a fundamental problem in computational geometry [Lee84]. Fast solutions for intersection problems are desirable in a wide range of application areas, including linear programming [Dant63] , hidden surface elimination [Newm79] , or spatial databases [Gunt]. In many of these applications, the dimension of the intersection problems may be greater than three. This is particularly obvious in linear programming; another example are database applications where geometric objects are used to represent predicates [Ston86].

It was first noted by Chazelle and Dobkin [Chaz80] that it is often easier to *detect* the intersection of two suitably preprocessed geometric objects rather than to actually *compute* it. In the detection problem, one only asks if two objects intersect or not; also, it is allowed to preprocess each of the given objects separately.

In this paper, we present algorithms to solve the intersection detection problem in arbitrary dimensions for hyperplanes and convex polyhedra. In particular, we obtain upper bounds of $O(2^d \log n)$ and $O((2d)^{d-1} \log^{d-1} n)$ for the d-dimensional hyperplane-polyhedron and polyhedron-polyhedron intersection problems, respectively. In this paper, n denotes the maximum number of vertices of any given polyhedron. These time bounds appear to be the first results for $d>3$ and match the time bounds given by Dobkin and Kirkpatrick [Dobk83,Dobk] for $d=2,3$. Furthermore, our results readily extend to unbounded polyhedra. For simplicity, this presentation is restricted to bounded polyhedra; see [Gunt] for the general case.

We obtain our results by means of a geometric duality transformation in d-dimensional Euclidean space $\mathbf{E}^d$ that is an isomorphism between points and hyperplanes [Brow79,Lee84]. Each convex polyhedron P is represented by a set of two functions in the

This research was sponsored under research contract DAAG29-85-0223 and a scholarship from the German National Scholarship Foundation.

dual space, TOP^P, BOT^P: $\mathbf{E}^{d-1} \rightarrow \mathbf{E}^1$, such that a hyperplane h intersects P if and only if the dual of h lies between TOP^P and BOT^P. Then, two polyhedra P and Q intersect if and only if for all $x \in \mathbf{E}^{d-1}$, we have $TOP^P(x) > BOT^Q(x)$ and $TOP^Q(x) > BOT^P(x)$.

For $d=2$ and $d=3$, the space and preprocessing requirements of the dual representation scheme are $O(n)$ and therefore optimal. For general d, the scheme requires $O(n^{2^{d-2}d})$ space and $O(2^d n^{2^{d-2}d} \log n)$ preprocessing. To improve these bounds is a subject of further research. In particular, we suspect that lower bounds may be achieved at the expense of slightly higher time bounds for the detection algorithms.

Section 2 introduces the dual representation scheme for convex polyhedra. Sections 3 and 4 show how the hyperplane-polyhedron and the polyhedron-polyhedron intersection detection problems can be solved efficiently using the dual scheme. Section 5 contains our conclusions.

2. The Dual Representation Scheme

Let h denote some non-vertical $(d-1)$-dimensional hyperplane in $\mathbf{E}^d$. That is, in a d-dimensional Cartesian coordinate system, h intersects the d-th coordinate axis in a unique and finite point and can be represented by an equation

$$x_d = a_1 x_1 + \, . \, . + a_{d-1} x_{d-1} + a_d .$$

F_h denotes the function whose graph is h, i.e.

$$F_h : \mathbf{E}^{d-1} \rightarrow \mathbf{E}^1$$

$$F_h(x_1 \, . \, . \, x_{d-1}) = a_1 x_1 + \, . \, . + a_{d-1} x_{d-1} + a_d .$$

A point $p = (p_1 \, . \, . \, p_d)$ lies *above* (*on*, *below*) h if $p_d > (=, <) \; F_h(p_1 \, . \, . \, p_{d-1})$.

We define a duality transformation D in $\mathbf{E}^d$ that maps hyperplanes into points and vice versa. The dual $D(h)$ of hyperplane h is the point $(a_1 \, . \, . \, a_d)$ in $\mathbf{E}^d$. Conversely, the dual $D(p)$ of a point p is the hyperplane defined by the equation

$$x_d = -p_1 x_1 - p_2 x_2 - \, . \, . - p_{d-1} x_{d-1} + p_d .$$

Lemma 2.1: A point p lies above (on, below) a hyperplane h if and only if the dual $D(h)$ lies below (on, above) $D(p)$.

Proof: Let h be given by the equation $F_h(x_1 \, . \, . \, x_{d-1}) = a_1 x_1 + \, . \, . + a_{d-1} x_{d-1} + a_d$ and let $p = (p_1 \, . \, . \, p_d)$ be a point above (on, below) h, i.e.

$$p_d > (=, <) \; F_h(p_1 \, . \, . \, p_{d-1}) \qquad (*).$$

Inserting $D(h) = (a_1 \, . \, . \, a_d)$ into $F_{D(p)}$ yields

$$F_{D(p)}(a_1 \,.\,.\, a_{d-1}) = -p_1a_1- \,.\,.\, -p_{d-1}a_{d-1}+p_d > (=,<)\, a_d \qquad \text{(due to (*))}$$

Hence, $D(p)$ lies below (on, above) $D(h)$. □

A hyperplane h intersects the polyhedron P if and only if there are two vertices v and w of P such that h lies between v and w (i.e. v lies on or above h and w lies on or below h, or vice versa). According to lemma 2.1, this is the case if and only if the dual $D(h)$ lies between the duals $D(v)$ and $D(w)$.

This observation leads to a new representation scheme for finite convex polyhedra. Consider the functions TOP^P, BOT^P: $\mathbf{E}^{d-1} \rightarrow \mathbf{E}^1$ that are defined for a convex polyhedron P as follows. Here, V_P denotes the set of vertices of P.

$$TOP^P(x_1 \,.\,.\, x_{d-1}) = \max_{v \in V_P} F_{D(v)}(x_1 \,.\,.\, x_{d-1})$$

$$BOT^P(x_1 \,.\,.\, x_{d-1}) = \min_{v \in V_P} F_{D(v)}(x_1 \,.\,.\, x_{d-1})$$

Obviously, both functions are piecewise linear, continuous, and TOP^P is convex, whereas BOT^P is concave [Rock70]. With this notation, a non-vertical hyperplane h intersects P if and only if $D(h)$ lies between TOP^P and BOT^P. More formally, the hyperplane h, given by the equation $x_d = a_1x_1+ \,.\,.\, +a_{d-1}x_{d-1}+a_d$, intersects P if and only if $BOT^P(a_1 \,.\,.\, a_{d-1}) \leq a_d \leq TOP^P(a_1 \,.\,.\, a_{d-1})$.

The two functions can be viewed as a mapping that map any slope $(a_1 \,.\,.\, a_{d-1})$ of a non-vertical hyperplane into the maximum (TOP^P) or minimum (BOT^P) intercept a_d such that the hyperplane given by $x_d = a_1x_1+ \,.\,.\, +a_{d-1}x_{d-1}+a_d$ intersects the polyhedron. We have

Theorem 2.2: Each convex polyhedron P corresponds to exactly one pair of functions (TOP^P, BOT^P), and conversely.

Proof: The functions TOP^P and BOT^P are uniquely defined for any convex regular polyhedron P, i.e. there is only one pair of functions (TOP^P, BOT^P) for any P.

Conversely, suppose there were two convex polyhedra P and Q such that $P \neq Q$, but $TOP^P(x_1 \,.\,.\, x_{d-1}) = TOP^Q(x_1 \,.\,.\, x_{d-1})$ and $BOT^P(x_1 \,.\,.\, x_{d-1}) = BOT^Q(x_1 \,.\,.\, x_{d-1})$ for all $(x_1 \,.\,.\, x_{d-1}) \in \mathbf{E}^{d-1}$.

Case 1: $P \cap Q = \phi$. Then there exists a non-vertical separating hyperplane h such that all points of P lie above h and all points of Q lie below h, or vice versa. There also exists a hyperplane h' parallel to h that intersects P. h' does not intersect Q. I.e., according to lemma 2.2, the dual $D(h')$ lies between TOP^P and BOT^P, but not between TOP^Q and BOT^Q. This is a contradiction to our assumption.

Case 2: $P \cap Q \neq \phi$. Because of $P \neq Q$ it is $P-Q \neq \phi$ or $Q-P \neq \phi$. W.l.o.g., let $P-Q \neq \phi$. Let p be some interior point of $P-Q$. There exists a non-vertical separating hyperplane h such that all points of Q lie above h and point p lies below h, or vice versa. There also exists a hyperplane h' parallel to h that goes through p. Because of $p \in P$, h' intersects P, but it does not intersect Q. Contradiction to our assumption as above. □

An example of a polyhedron P and the corresponding functions TOP^P and BOT^P is given in figure 2.1.

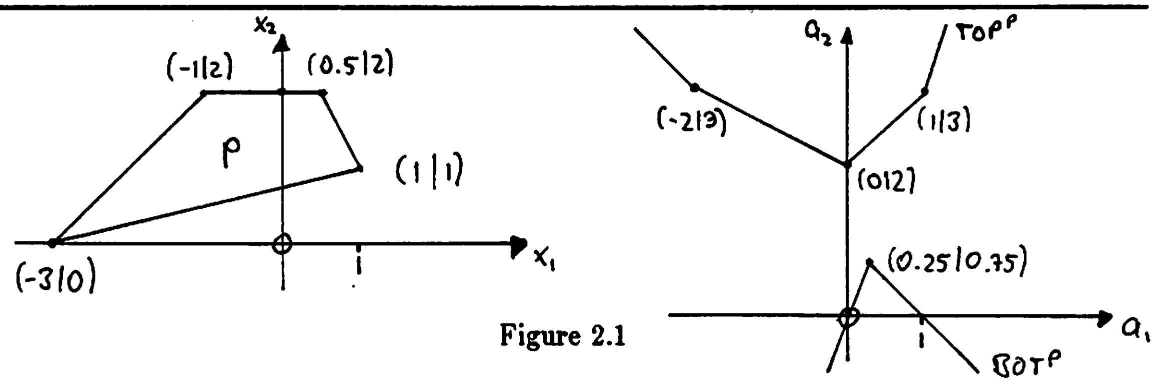

Figure 2.1

3. Hyperplane-Polyhedron Intersection Detection

For simplicity of presentation, we exclude the case of a vertical hyperplane. This exclusion can always be achieved by a suitable rotation of the coordinate system. It is also possible to extend our detection algorithm to detect intersections with a vertical hyperplane; see [Gunt] for details.

A non-vertical hyperplane h, given by $x_d = a_1 x_1 + \ldots + a_{d-1} x_{d-1} + a_d$ intersects a polyhedron P if and only if $BOT^P(a_1 \ldots a_{d-1}) \leq a_d \leq TOP^P(a_1 \ldots a_{d-1})$. Therefore, the intersection detection problem can be solved by obtaining the functional values $TOP^P(a_1 \ldots a_{d-1})$ and $BOT^P(a_1 \ldots a_{d-1})$. Without loss of generality, we only show how to obtain $TOP^P(a_1 \ldots a_{d-1})$.

The graph of TOP^P is a polyhedral surface in $\mathbf{E}^d$, consisting of $O(n)$ convex $(d-1)$-dimensional patches with $m = O(n^2)$ $(d-2)$-dimensional boundary segments. With the properties of the duality transformation the following lemma can be proven.

Lemma 3.1: Each k-dimensional face f of the upper hull of the polyhedron P corresponds to exactly one $(d-k-1)$-dimensional face $D(f)$ of TOP^P's graph, and vice versa. Furthermore, if two faces f_1 and f_2 of P's upper hull are adjacent, then so are the faces $D(f_1)$ and $D(f_2)$ of TOP^P's graph.

Proof: see [Gunt]. □

The projection of TOP^P's graph on the $(d-1)$-dimensional hyperplane $J{:}a_d=0$ partitions J into no more than n convex $(d-1)$-dimensional partitions with no more than m $(d-2)$-dimensional boundary segments. Any given partition $E \subseteq J$ corresponds to a vertex $v(E)$ of P's upper hull, such that for any point $(p_1 \ldots p_{d-1}) \in E$, it is $TOP^P(p_1 \ldots p_{d-1}) = F_{D(v(E))}(p_1 \ldots p_{d-1})$. Hence, $TOP^P(a_1 \ldots a_{d-1})$ can be obtained by a $(d-1)$-dimensional point location in J to find the partition E that contains the point $(a_1 \ldots a_{d-1})$, followed by a computation of $F_{D(v(E))}(a_1 \ldots a_{d-1})$.

The computation of $F_{D(v(E))}(a_1 \ldots a_{d-1})$ takes time $O(d)$. Dobkin and Lipton [Dobk76] solve a $(d-1)$-dimensional point location problem with m $(d-2)$-dimensional boundary segments recursively as follows. In a preprocessing step, they compute the $O(m^2)$ $(d-3)$-dimensional intersection segments formed by the m original boundary segments, and project them on some $(d-2)$-dimensional hyperplane K. This way, the point location problem can be solved by a point location problem in K, followed by a binary search of the m original segments. According to Dobkin and Lipton [Dobk76] , this computation can be carried out in no more than $(3 \cdot 2^{d-3}+d-3)(\lfloor \log m \rfloor+1) \le 2^{d+1}\log n = O(2^d \log n)$ steps.* We obtain a total time complexity of $O(2^d \log n)$.

The space requirements of this algorithm are as follows. The equations of the $O(n)$ patches require space $O(dn)$. The space requirements to store a space partitioning of $\mathbf{E}^2$ with m boundary segments, $SP(2,m)$, is $O(m^2)$ [Dobk76]. For a space partitioning of $\mathbf{E}^{d-1}$ with m boundary segments, one has to store a space partitioning of the $(d-2)$-dimensional projection hyperplane K with m^2 boundary segments and a sequence of no more than m boundary segments for each of the partitions. The number of partitions is no more than $m^{2(d-2)}$ [Edel83]. Therefore,

$$
\begin{aligned}
& SP(d-1,m) \\
& \le SP(d-2,m^2) + m^{2(d-2)}m \\
& \le SP(d-3,m^4) + m^{4(d-3)}m^2 + m^{2(d-2)}m \\
& \le \ldots \\
& \le SP(2,m^{2^{d-3}}) + O(m^{2^{d-3}d}) = O(m^{2^{d-3}d}) = O(n^{2^{d-3}d}) .
\end{aligned}
$$

We obtain a total space complexity of $O(n^{2^{d-3}}d)$.

---

* Here, Dobkin and Lipton assume that it is an elementary operation to determine on which side of a given hyperplane a point is located. Without this assumption, all time bounds are to be multiplied by d.

The preprocessing requirements of this algorithm are as follows. According to lemma 3.1, each $(d-2)$-dimensional boundary segment of the space partitioning can be obtained from the original polyhedron P in time $O(d)$ by dualization. Here, we assume that P is given by a list of its faces and the corresponding adjacency relations. As there are $m = O(n^2)$ $(d-2)$-dimensional boundary segments, it takes time $O(dn^2)$ to obtain all of them.

Given the m boundary segments of the space partitioning of $\mathbf{E}^2$, it takes $PRP(2,m) = O(m^2 \log m)$ preprocessing to compute all intersections, project them on some line, and to sort the boundary segments for each of the $O(m)$ intersections [Dobk76]. For a space partitioning of $\mathbf{E}^{d-1}$ with m boundary segments, one has to compute m^2 intersections, and to project them on some $(d-2)$-dimensional hyperplane K. For each of the $O(m^{2(d-2)})$ partitions, one has to sort the $O(m)$ boundary segments. Einally, one has to do the necessary preprocessing for the space partitioning of K. Therefore,

$$
\begin{aligned}
& PRP(d-1,m) \\
\leq\ & PRP(d-2,m^2) + m^{2(d-2)} m \log m \\
\leq\ & PRP(d-3,m^4) + m^{4(d-3)} m^2 \log m^2 + m^{2(d-2)} m \log m \\
\leq\ & \ldots \\
\leq\ & PRP(2,m^{2^{d-3}}) + O(m^{2^{d-2}d} \log m^{2^{d-3}}) = O(2^d m^{2^{d-2}d} \log m) = O(2^d n^{2^{d-2}d} \log n)\ .
\end{aligned}
$$

We obtain a total preprocessing time of $O(2^d n^{2^{d-2}d} \log n)$.

Theorem 3.2 summarizes our results for the hyperplane-polyhedron intersection detection problem. The bounds for $d=2$ and $d=3$ are more favorable than for general d, because for that case the numbers of vertices, edges, and faces are pairwise proportional [Prep85], and Dobkin and Lipton's point location method has been superseded by more efficient algorithms [Edel86].

Theorem 3.2: Given a non-vertical $(d-1)$-dimensional hyperplane h and a d-dimensional convex polyhedron P, h and P can be tested for intersection in time $T(n,d)$ with $S(n,d)$ space and $PP(n,d)$ preprocessing:

$P \cap h = \phi$?	$T(n,d)$	$S(n,d)$	$PP(n,d)$
$d=2$	$O(\log n)$	$O(n)$	$O(n)$
$d=3$	$O(\log n)$	$O(n)$	$O(n)$
$d>3$	$O(2^d \log n)$	$O(n^{2^{d-2}d})$	$O(2^d n^{2^{d-2}d} \log n)$

Proof: follows from the preceding discussion. □

4. Polyhedron-Polyhedron Intersection Detection

For simplicity of presentation, we exclude the case of two non-intersecting polyhedra, whose only separating hyperplane is vertical (Fig. 4.1).

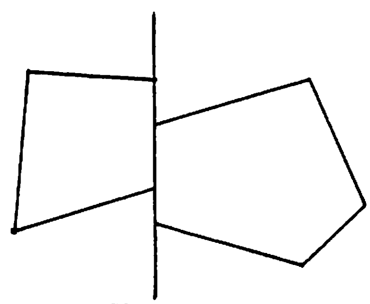

Figure 4.1

This can always be achieved by a suitable rotation of the coordinate system. It is also possible to extend the detection algorithm to solve this case correctly; see [Gunt] for details.

Under this assumption, two convex polyhedra P and Q do not intersect if and only if there is a separating non-vertical hyperplane between them. Any such hyperplane h does not intersect the interior of either P or Q, but there are hyperplanes h' and h'' parallel to h, such that h' is above h and h'' is below h, and either h' intersects the interior of P and h'' intersects the interior of Q, or vice versa. More formally, a hyperplane h, given by the equation $x_d = a_1 x_1 + \ldots + a_{d-1} x_{d-1} + a_d$, separates the polyhedra P and Q if and only if

$$TOP^P(a_1 \ldots a_{d-1}) \leq a_d \leq BOT^Q(a_1 \ldots a_{d-1}), \text{ or}$$

$$TOP^Q(a_1 \ldots a_{d-1}) \leq a_d \leq BOT^P(a_1 \ldots a_{d-1}).$$

Therefore, two polyhedra P and Q do not intersect if and only if*

* We write $(f \pm g)(x)$ for $f(x) \pm g(x)$.

(i) $\min_{(x_1..x_{d-1})\in E^{d-1}} (TOP^P - BOT^Q)(x_1 \,..\, x_{d-1}) \le 0$, or

(ii) $\min_{(x_1..x_{d-1})\in E^{d-1}} (TOP^Q - BOT^P)(x_1 \,..\, x_{d-1}) \le 0$.

To test these conditions, we present a multidimensional search technique that finds the minimum of a convex piecewise linear function in arbitrary dimensions. The technique is recursive; it solves a d-dimensional problem by solving $O(d\log n)$ $(d-1)$-dimensional problems, and so on.

Condition (i) holds if and only if the graphs of TOP^P and BOT^Q do not intersect. In the two-dimensional case, this can be tested by a variation of Dobkin and Kirkpatrick's algorithm [Dobk83] to detect the intersection of two polygons. The graphs of TOP^P and BOT^Q are monotone convex polygonal chains with edges $t_1 \ldots t_k$ and $b_1 \,..\, b_l$ $(k+l \le 2n)$; see also figure 2.1. The relative position and the slopes of the edges $t_{\lfloor k/2 \rfloor}$ and $b_{\lfloor l/2 \rfloor}$ give enough information to eliminate half of the edges of one (or both) chains from further consideration without missing any intersection. The algorithm proceeds recursively, eliminating at least one quarter of the remaining edges at each recursion level. Therefore, any intersection is detected in time $O(\log n)$ without any preprocessing or extra storage. A similar analysis yields the same bound to test condition (ii).

In order to solve the d-dimensional problem, we solve $O(d\log n)$ $(d-1)$-dimensional problems. Without loss of generality, we only show how to test condition (i). It is well known [Dant63] that the global minimum of $TOP^P - BOT^Q$ occurs at some vertex of the graph of $TOP^P - BOT^Q$, i.e. at some vertex $M=(M_1 \,..\, M_d)$ of TOP^P's graph TG or BOT^P's graph BG. Let $(v_1 \,..\, v_{|TG|})$ denote the sequence of vertices in V_{TG}, sorted by increasing x_1-coordinate. We consider the vertex $v_{\lfloor |TG|/2 \rfloor}$ and its x_1-coordinate $\overline{a_1}$, and compute the local minimum of $TOP^P - BOT^Q$ along the hyperplane $x_1 = \overline{a_1}$. This is a $(d-1)$-dimensional minimization problem and can be solved recursively; let $m=(m_1=\overline{a_1}, m_2, \ldots, m_d)$ denote some point where the local minimum is assumed. Due to the convexity of $TOP^P - BOT^Q$, we can determine the position of M relative to m from the local slope of $TOP^P - BOT^Q$. We have

Lemma 4.1: It is $M_1 > (<) m_1$ if and only if there is an $\epsilon_0 > 0$, such that for all ϵ with $0 < \epsilon < \epsilon_0$ it is

$$\begin{aligned} & TOP^P - BOT^Q(m_1-\epsilon, m_2 \,..\, m_d) \\ >(<)\ & TOP^P - BOT^Q(m_1 \,..\, m_d) \\ >(<)\ & TOP^P - BOT^Q(m_1+\epsilon, m_2 \,..\, m_d). \end{aligned}$$

Otherwise, m is a global minimum of $TOP^P - BOT^Q$.

Proof: Due to the convexity of the function $TOP^P - BOT^Q$, there is always an $\epsilon_0 > 0$, such that for all ϵ with $0 < \epsilon < \epsilon_0$ exactly one of the following conditions holds:

(i) $TOP^P-BOT^Q(m_1-\epsilon,m_2\,..\,m_d) > TOP^P-BOT^Q(m_1\,..\,m_d) > TOP^P-BOT^Q(m_1+\epsilon,m_2\,..\,m_d)$,

(ii) $TOP^P-BOT^Q(m_1-\epsilon,m_2\,..\,m_d) < TOP^P-BOT^Q(m_1\,..\,m_d) < TOP^P-BOT^Q(m_1+\epsilon,m_2\,..\,m_d)$,

(iii) $TOP^P-BOT^Q(m_1-\epsilon,m_2\,..\,m_d) \geq TOP^P-BOT^Q(m_1\,..\,m_d)$

$\wedge$ $TOP^P-BOT^Q(m_1+\epsilon,m_2\,..\,m_d) \geq TOP^P-BOT^Q(m_1\,..\,m_d)$

We now show indirectly that $M_1>m_1$ implies condition (i). Suppose that $M_1>m_1$, but (i) does not hold, i.e. $TOP^P-BOT^Q(m_1+\epsilon,m_2\,..\,m_d) \geq TOP^P-BOT^Q(m_1\,..\,m_d)$ (*). Let $r = (r_1,r_2=m_2,r_3,\,.\,.\,,r_d)$ denote the minimum of TOP^P-BOT^Q along the hyperplane $x_2=m_2$. Due to the convexity of TOP^P-BOT^Q and condition (*), it is $r_1\leq m_1$ and $r_d\leq m_d$. Therefore, the line segment (M,r) intersects the hyperplane $x_1=m_1$ in some point $s = (s_1=m_1,s_2,\,.\,.\,,s_d)$. Because of $M_d<r_d$, it is $s_d<r_d$, and because of $r_d\leq m_d$, it is $s_d<m_d$. This is a contradiction, because s lies on the hyperplane $x_1=m_1$, and m is the minimum along this hyperplane. A two-dimensional example is given in figure 4.2.

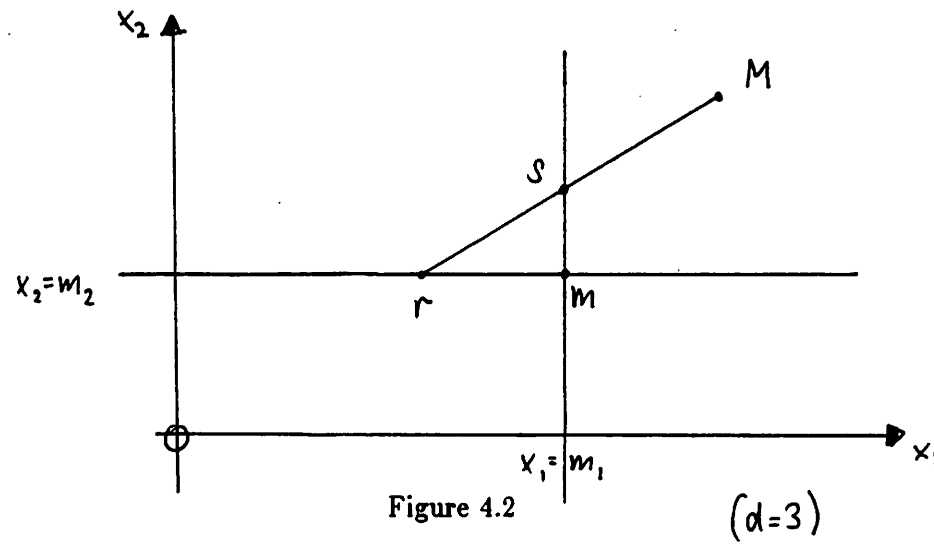

Figure 4.2

Hence, $M_1>m_1$ implies condition (i). Similarly, it can be shown that $M_1<m_1$ implies condition (ii), and that $M_1=m_1$ implies condition (iii). Due to the mutual exclusiveness of conditions (i), (ii) and (iii), we obtain that (i) implies $M_1>m_1$ and so on. This proves the lemma. □

Therefore, looking up the functional values $TOP^P-BOT^Q(m_1\pm\epsilon,m_2\,..\,m_d)$ for some suitable $\epsilon>0$ gives us enough information to eliminate half of the vertices in V_{TG} (and some vertices in V_{BG}) from the search without missing the global minimum. If the search among the vertices in TG does not yield a global minimum, one continues with a similar search among the remaining vertices of BG. Hence the global minimum is obtained in no more than $\log(\,|\,TG\,|\,+\,|\,BG\,|\,)$ iterations.

The analysis of this algorithm obviously depends on the cardinalities of TG and BG. A simple combinatorial analysis shows that at any recursion level it is $|TG|+|BG| \leq n^d$, i.e. the algorithm requires no more than $d\log n$ iterations. Each iteration involves a $(d-1)$-dimensional minimization and the four function lookups necessary to obtain $TOP^P - BOT^Q(m_1 \pm \epsilon, m_2 \ldots m_d)$. As shown in section 3, each lookup can be carried out in no more than $2^{d+1}\log n$ steps. We obtain a total time complexity

$$
\begin{aligned}
& T(d,n) \\
\leq\ & d\log n \cdot (4 \cdot 2^{d+1}\log n + T(n,d-1)) \\
\leq\ & 2^{d+3} d\log^2 n + d\log n T(n,d-1) \\
\leq\ & 2^{d+3} d\log^2 n + d\log n\, 2^{d+2}(d-1)\log^2 n + d(d-1)\log^2 n T(n,d-2) \\
\leq\ & \ldots \\
\leq\ & \sum_{i=1}^{d-2} 2^{d+4-i} d^i \log^{i+1} n = O((2d)^{d-1}\log^{d-1} n)\,.
\end{aligned}
$$

Of course, in practice one might be able to solve the intersection detection problem much faster by checking at various stages if $(TOP^P - BOT^Q)(x_1 \ldots x_{d-1}) \leq 0$, or $(TOP^Q - BOT^P)(x_1 \ldots x_{d-1}) \leq 0$.

An analysis similar to the one in section 3 shows that the space and preprocessing requirements are the same as for the hyperplane-polyhedron intersection detection problem. We obtain

Theorem 4.2: Given two d-dimensional convex polyhedra P and Q, P and Q can be tested for intersection in time $T(n,d)$ with $S(n,d)$ space and $PP(n,d)$ preprocessing:

$P \cap Q = \phi$?	$T(n,d)$	$S(n,d)$	$PP(n,d)$
$d=2$	$O(\log n)$	$O(n)$	$O(n)$
$d=3$	$O(\log^2 n)$	$O(n)$	$O(n)$
$d>3$	$O((2d)^{d-1}\log^{d-1} n)$	$O(n^{2^{d-2}d})$	$O(2^d n^{2^{d-1}d} \log n)$

Proof: follows from the preceding discussion. □

5. Conclusions

We showed that in arbitrary, but fixed dimensions, the hyperplane-polyhedron and the polyhedron-polyhedron intersection detection problems can be solved in logarithmic and polylogarithmic time, respectively. For dimensions larger than three, these results appear to be new. There are two reasons why, as of now, these results are of primarily theoretical interest. First, the coefficient which is exponential in d becomes prohibitively high for higher dimensions. Second, the storage and preprocessing requirements are not suitable for practical purposes. It is subject to further research to improve these results in order to achieve practical algorithms for intersection detection in higher dimensions.

References

[Brow79] Brown, K. Q., *Geometric transformations for fast geometric algorithms,* Ph.D. dissertation, Carnegie-Mellon University, Pittsburgh, Pa., Dec. 1979.

[Chaz80] Chazelle, B. and D. P. Dobkin, Detection is easier than computation, in *Proc. 12th Annual ACM Symposium on Theory of Computing*, 1980, pages 146-153.

[Dant63] Dantzig, G. B., *Linear Programming and Its Extensions,* Princeton University Press, Princeton, NJ, 1963.

[Dobk] Dobkin, D. P. and D. G. Kirkpatrick, *Fast algorithms for preprocessed polyhedral intersection detection,* in preparation.

[Dobk76] Dobkin, D. P. and R. J. Lipton, Multidimensional searching problems, *SIAM J. Comput.* **5**, 2 (June 1976), pages 181-186.

[Dobk83] Dobkin, D. P. and D. G. Kirkpatrick, Fast detection of polyhedral intersection, *Theoret. Comput. Sci.* **27** (1983), pages 241-253.

[Edel83] Edelsbrunner, H., J. O'Rourke, and R. Seidel, Constructing arrangements of lines and hyperplanes with applications, in *Proc. IEEE Annu. Symp. Found. Comput. Sci.*, Nov. 1983, pages 83-91.

[Edel86] Edelsbrunner, H., L. J. Guibas, and J. Stolfi, Optimal point location in a monotone subdivision, in *SIAM J. Comput.* **15**, 1986.

[Gunt] Gunther, O., *Data bases and data structures for geometric data,* Ph.D. dissertation, University of California, Berkeley, Ca., in preparation.

[Lee84] Lee, D. T. and F. P. Preparata, Computational geometry - a survey, *IEEE Trans. on Computers* **C-33**, 12 (Dec. 1984), pages 1072-1101.

[Newm79] Newman, W. M. and R. F. Sproull, *Principles of Interactive Computer*

Graphics, McGraw-Hill, New York, NY, 1979.

[Prep85] Preparata, F. P. and M. I. Shamos, *Computational geometry,* Springer-Verlag, New York, NY, 1985.

[Rock70] Rockafellar, R. T., *Convex Analysis,* Princeton, NJ, 1970.

[Ston86] Stonebraker, M., T. Sellis, and E. Hanson, An analysis of rule indexing implementations in data base systems, in *Proc. of the 1st International Conference on Expert Data Base Systems*, April 1986.