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
gunther@postgres.berkeley.edu

Abstract

This paper presents a dual approach to detect intersections of hyperplanes and convex polyhedra in arbitrary dimensions. In d dimensions, the time complexities of the dual algorithms are $O(2^d d \log n)$ for the hyperplane-polyhedron intersection problem, and $O((2d)^{d-1}\log^{d-1} n)$ for the polyhedron-polyhedron intersection problem. In two dimensions, these time bounds are achieved with linear space and preprocessing. In three dimensions, the hyperplane-polyhedron intersection problem is also solved with linear space and preprocessing; quadratic space and preprocessing, however, is required for the polyhedron-polyhedron intersection problem. For general d, the dual algorithms require $O(n^{2^{d-2}d})$ space and $O(2^d n^{2^{d-2}d} \log n)$ preprocessing. All of these results readily extend to unbounded polyhedra.

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 [Gunt87]. 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. A *(d-dimensional, convex) polyhedron* P in d-dimensional Euclidean space E^d is defined to be the intersection of some finite number of closed halfspaces in E^d, such that the dimension of the smallest affine subspace containing P is d. If $a \in E^d - \{0\}$ and $c \in E^1$ then the $(d-1)$-dimensional set $H(a,c)=\{x \in E^d : x \cdot a = c\}$ is called a *hyperplane* in E^d.* A hyperplane $H(a,c)$ defines two closed halfspaces $H^+(a,c)=\{x \in E^d : x \cdot a \geq c\}$ and $H^-(a,c)=\{x \in E^d : x \cdot a \leq c\}$. 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 a *face* of P. The faces of dimension 1 are called *edges*; those of dimension 0 *vertices*. A supporting hyperplane is called a *boundary* hyperplane is the face $H(a,c) \cap P$ is of dimension $d-1$. The faces of P that are a subset of some supporting hyperplane $H(a,c)$, with $a_d \leq 0$, form the *upper hull* of P. The *lower hull* of P is defined similarly.

So far, the intersection detection problem has only been considered in two and three dimensions. In their original paper, Chazelle and Dobkin [Chaz80] solve the d-dimensional hyperplane-polyhedron intersection problem in time $O(\log n)$ $(d=2)$ and $O(\log^2 n)$ $(d=3)$, and the polyhedron-polyhedron intersection problem in time $O(\log n)$ $(d=2)$ and $O(\log^3 n)$ $(d=3)$. Here, n denotes the maximum number of vertices of any given polyhedron. Both problems require $O(n)$ $(d=2)$ and $O(n^2)$ $(d=3)$ space and preprocessing. A revised version of that paper has been published recently [Chaz87]. In the three-dimensional case,

† This research was sponsored under grant DAAG 29-85-0223 and an IBM Fellowship for the first author.

* $x \cdot y$ denotes the inner product of vectors x and y.

$O(n \log n)$ space and preprocessing are also sufficient [Dobk80], in which case the running times given above have to be multiplied by a $\log n$ factor.

In a later paper, Dobkin and Kirkpatrick [Dobk83] improve the running times of Chazelle and Dobkin for the three-dimensional case by a factor of $\log n$. The new upper bounds are $O(\log n)$ and $O(\log^2 n)$ for the hyperplane-polyhedron and the polyhedron-polyhedron problems, respectively. As the algorithms of Chazelle and Dobkin, their algorithms require $O(n^2)$ storage and preprocessing. Again, the results of Dobkin and Munro [Dobk80] can be used to reduce the space and preprocessing requirements in three dimensions to $O(n \log n)$, in which case the running times increase by a $\log n$ factor.

In d dimensions, we obtain upper time bounds of $O(2^d d \log n)$ to detect the intersection of a hyperplane and a polyhedron, and $O((2d)^{d-1} \log^{d-1} n)$ to detect the intersection of two polyhedra. These time bounds appear to be the first results for $d>3$ and match the time bounds given by Dobkin and Kirkpatrick [Dobk83] for $d=2$ and $d=3$. Furthermore, our results seem to be the first of their kind that extend to unbounded polyhedra as well.

We obtain our results by means of a geometric duality transformation in d-dimensional Euclidean space 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 dual space, $TOP^P, BOT^P: E^{d-1} \rightarrow E^1$, 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 $x \in E^{d-1}$, we have $TOP^P(x) \geq BOT^Q(x)$ and $TOP^Q(x) \geq BOT^P(x)$.

For $d=2$ and for the hyperplane-polyhedron intersection problem in $d=3$, the space and preprocessing requirements of the dual representation scheme are $O(n)$ and therefore optimal. For the three-dimensional hyperplane-polyhedron intersection problem, this represents an improvement over the results of Dobkin and Kirkpatrick [Dobk83] by a factor of n. The three-dimensional polyhedron-polyhedron problem takes quadratic space and preprocessing, as does the algorithm of Dobkin and Kirkpatrick.

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 presents several extensions of our approach, and section 6 contains our conclusions.

2. The Dual Representation Scheme

Let h denote some non-vertical $(d-1)$-dimensional hyperplane in 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

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

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

$$F_h : E^{d-1} \rightarrow E^1$$
$$F_h(x_1 \ldots x_{d-1}) = a_1 x_1 + \ldots + a_{d-1} x_{d-1} + a_d .$$

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

Brown [Brow79] defines a duality transformation D in E^d that maps hyperplanes into points and vice versa. The dual $D(h)$ of hyperplane h is the point $(a_1 \ldots a_d)$ in E^d. Conversely, the dual $D(p)$ of a point p is the hyperplane defined by the equation

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

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

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

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

$F_{D(p)}(a_1 \ldots a_{d-1}) = -p_1 a_1 - \ldots - p_{d-1} a_{d-1} + p_d > (=,<)\ a_d$ (due to (*))

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

A hyperplane h intersects a bounded 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 bounded 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.

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

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

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 $x_d = a_1 x_1 + \ldots + a_{d-1} x_{d-1} + a_d$, intersects P if and only if $BOT^P(a_1 \ldots a_{d-1}) \le a_d \le TOP^P(a_1 \ldots a_{d-1})$. A two-dimensional example of a polyhedron P and the corresponding functions TOP^P and BOT^P is given in figure 2.1.

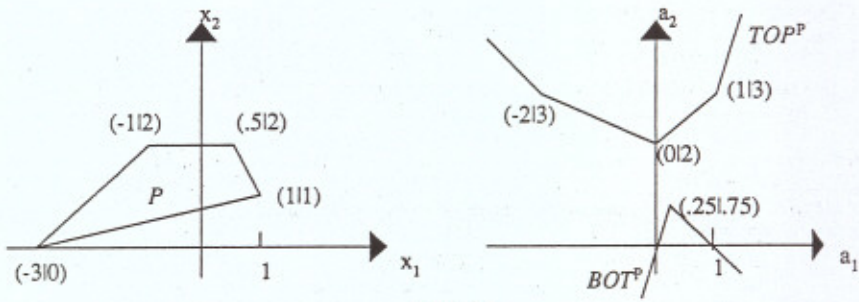


Figure 2.1

It is easily possible to extend this representation scheme to unbounded polyhedra. For simplicity, however, the main part of this paper is restricted to bounded polyhedra; the case of unbounded polyhedra is discussed in more detail in section 5.1.

The two functions TOP^P and BOT^P can be viewed as a mapping that map any slope $(a_1 \ldots a_{d-1})$ of a non-vertical hyperplane into the maximum (TOP^P) or minimum (BOT^P) intercept a_d such that the hyperplane given by $x_d = a_1 x_1 + \ldots + a_{d-1} x_{d-1} + a_d$ 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 $TOP^P(x_1 \ldots x_{d-1}) = TOP^Q(x_1 \ldots x_{d-1})$ and $BOT^P(x_1 \ldots x_{d-1}) = BOT^Q(x_1 \ldots x_{d-1})$ for all $(x_1 \ldots x_{d-1}) \in \mathbf{E}^{d-1}$.

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., 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. □

3. Hyperplane-Polyhedron Intersection Detection

For simplicity of presentation, we assume that the given hyperplane is non-vertical. This 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 section 5.2 for details.

A non-vertical hyperplane h, given by $x_d = a_1x_1 + \ldots + a_{d-1}x_{d-1} + a_d$ intersects a bounded polyhedron P if and only if $BOT^P(a_1 \ldots a_{d-1}) \leq a_d \leq TOP^P(a_1 \ldots a_{d-1})$. Moreover, an intersecting hyperplane h supports P if and only if $a_d = BOT^P(a_1 \ldots a_{d-1})$ or $a_d = TOP^P(a_1 \ldots a_{d-1})$. Therefore, the intersection detection problem can be solved by obtaining the functional values $TOP^P(a_1 \ldots a_{d-1})$ and $BOT^P(a_1 \ldots a_{d-1})$. It follows from the definition of TOP^P and BOT^P that these values can be found in time $O(d \cdot n)$ by computing $F_{D(v)}(a_1 \ldots a_{d-1})$ for each vertex $v \in V_P$. With some preprocessing, however, one can obtain polylogarithmic time bounds as follows.

It follows from [Brow79] that there is the following isomorphism between the upper hull of the polyhedron P and the graph of TOP^P. Each k-dimensional face f of the upper hull of 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. The same isomorphism exists between P's lower hull and the graph of BOT^P. Hence, the graphs of TOP^P and BOT^P are polyhedral surfaces in E^d, consisting of no more than n convex $(d-1)$-dimensional faces and no more than $m = O(n^2)$ $(d-2)$-dimensional faces.

Without loss of generality, we only show how to obtain $TOP^P(a_1 \ldots a_{d-1})$. The projection of TOP^P's graph on the $(d-1)$-dimensional hyperplane $J: a_d = 0$ subdivides J into no more than n convex $(d-1)$-dimensional polyhedral 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 $F_{D(v(E))}(a_1 \ldots a_{d-1})$.

For $d=2$ and $d=3$, the computation of $F_{D(v(E))}(a_1 \ldots a_{d-1})$ takes only constant time. The point location can be performed in time $O(\log n)$, using the algorithm of Edelsbrunner, Guibas, and Stolfi [Edel86a]. The total time complexity to detect the intersection of a hyperplane and a polyhedron is therefore $O(\log n)$. The space and preprocessing requirements are only $O(n)$, due to the fact that, in our case, the given partitions are convex and therefore monotone.

For general d, it takes time $O(d)$ to compute the functional value $F_{D(v(E))}(a_1 \ldots a_{d-1})$. 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. Therefore, the time complexity of the point location is

$$
\begin{aligned}
& TPL(d-1, m) \\
& \leq TPL(d-2, m^2) + (d-1)(\lfloor \log m \rfloor + 1) \\
& \leq \ldots \\
& \leq TPL(2, m^{2^{d-3}}) + \sum_{i=1}^{d-3} (d-i)\left(\left\lfloor 2^{i-1} \log m \right\rfloor + 1\right)
\end{aligned}
$$

$= O(2^d d \log m) = O(2^d d \log n)$

We obtain a total time complexity of $O(2^d d \log n)$.*

For general d, the space requirements of the dual algorithm are as follows. The equations of the $O(n)$ faces require space $O(dn)$. The space requirements to store a convex subdivision of E^2 with m boundary segments, $SP(2,m)$, is $O(m)$ [Edel86a]. For a subdivision of E^{d-1} with m boundary segments, one has to store a subdivision 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 $m^{2(d-2)}$ [Edel86b]. Therefore,

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

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

The preprocessing requirements of this algorithm are as follows. Each $(d-2)$-dimensional boundary segment of the subdivision is obtained from the original polyhedron P in time $O(d)$ by dualization and projection. 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.

The preprocessing requirements to solve a point location problem in a convex subdivision of E^2 with m boundary segments, $PRP(2,m)$, are $O(m)$ [Edel86a]. For a subdivision of 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. Finally, one has to do the necessary preprocessing for the subdivision of K. Therefore,

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

We obtain a total preprocessing time of $O(2^d n^{2^{d-2}d} \log n)$. Theorem 3.1 summarizes our results for the hyperplane-polyhedron intersection detection problem.

Theorem 3.1: 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 $T(d,n)$ with $S(d,n)$ space and $PP(d,n)$ preprocessing:

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

Proof: follows from the preceding discussion. □

4. Polyhedron-Polyhedron Intersection Detection

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 either P or Q, but there are

* Note that we assume that it takes time $O(d)$ to determine on which side of a given hyperplane a point is located. Dobkin and Lipton [Dobk76] assume in their analysis that this can be done in constant time and consequently obtain a time complexity of $O(2^d \log n)$.

hyperplanes h' and h'' parallel to h, such that h' is above h and h'' is below h, and either h' intersects P and h'' intersects Q, or vice versa. More formally, a non-vertical hyperplane h, given by the equation $x_d = a_1x_1 + \ldots + a_{d-1}x_{d-1} + a_d$, separates the polyhedra P and Q if and only if

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

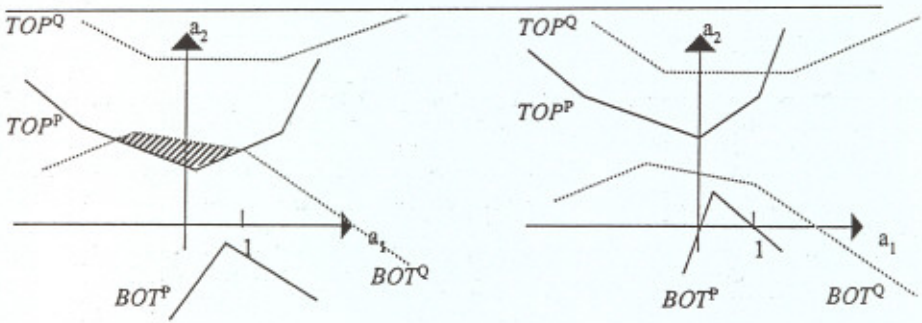
$$TOP^Q(a_1 \ldots a_{d-1}) < a_d < BOT^P(a_1 \ldots a_{d-1}).$$

Therefore, two polyhedra P and Q intersect if and only if*

(i) $\min_{(x_1 \ldots x_{d-1}) \in \mathbb{E}^{d-1}} (TOP^P - BOT^Q)(x_1 \ldots x_{d-1}) \geq 0$, and

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

See figure 4.1 for two examples. If both conditions are only met as equalities, then only the boundaries of P and Q intersect, but not their interiors.



(a) no intersection: the points in the shaded area are the duals of the separating hyperplanes.

(b) intersection

Figure 4.1

With the definitions of TOP^P and BOT^P, these conditions form a linear programming problem with no more than $2n$ constraints. According to Megiddo [Megi84], the time complexity to solve this problem is bound by $2^{O(2^d)} \cdot 2n$. Hence, the conditions can be tested in linear time $O(n)$ if the dimension is fixed. With some preprocessing, however, the conditions can be tested in polylogarithmic time as follows.

Without loss of generality, we only show how to test condition (i). 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.

In the two-dimensional case, condition (i) 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 $b_1 \ldots b_l$ $(k+l \leq 2n)$; 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 the minimum. The algorithm proceeds recursively, eliminating at least one quarter of the remaining edges at each recursion level. Therefore, the minimum is detected in time $O(\log n)$ without any preprocessing or extra space. 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. 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 $M = (M_1 \ldots M_d)$ of TOP^P's graph TG or BOT^Q's graph BG. Let $(v_1 \ldots v_{|TG|})$ denote the sequence of vertices in V_{TG}, sorted by increasing x_1-coordinate. We consider the

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

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 $\varepsilon_0 > 0$, such that for all ε with $0 < \varepsilon < \varepsilon_0$

$$\begin{array}{ll} & TOP^P - BOT^Q(m_1 - \varepsilon, m_2 \ldots m_d) \\ >(<) & TOP^P - BOT^Q(m_1 \ldots m_d) \\ >(<) & TOP^P - BOT^Q(m_1 + \varepsilon, m_2 \ldots m_d). \end{array}$$

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 $\varepsilon_0 > 0$, such that for all ε with $0 < \varepsilon < \varepsilon_0$ exactly one of the following conditions holds:

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

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

(iii) $TOP^P - BOT^Q(m_1 - \varepsilon, m_2 \ldots m_d) \geq TOP^P - BOT^Q(m_1 \ldots m_d)$
$\wedge\ TOP^P - BOT^Q(m_1 + \varepsilon, m_2 \ldots m_d) \geq TOP^P - BOT^Q(m_1 \ldots m_d)$

If condition (iii) holds, then m is a local minimum. Because $TOP^P - BOT^Q$ is convex, m also has to be a global minimum. Conversely, if m is a global minimum, condition (iii) clearly has to be true.

We now show indirectly that $M_1 > m_1$ implies condition (i). Suppose that $M_1 > m_1$, but (i) does not hold. Because m is not a global minimum, condition (ii) has to be true. Let $r = (r_1, r_2 = m_2, r_3, \ldots, r_d)$ denote the minimum of $TOP^P - BOT^Q$ along the hyperplane $x_2 = m_2$. Due to (ii) and to the convexity of $TOP^P - BOT^Q$, it is $r_1 < m_1$ and $r_d < m_d$. Therefore, the line segment (M, r) intersects the hyperplane $x_1 = m_1$ in some point $s = (s_1 = m_1, s_2, \ldots, s_d)$. Because of $M_d \leq r_d$, it is $s_d \leq r_d$, and because of $r_d < m_d$, 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.

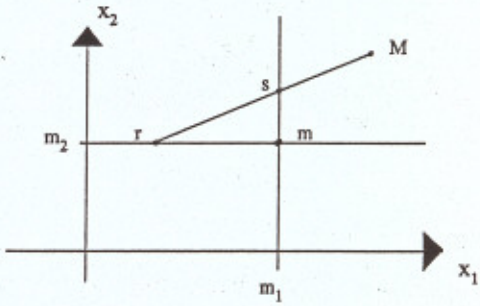


Figure 4.2

Hence, $M_1 > m_1$ implies condition (i). Similarly, it can be shown that $M_1 < m_1$ implies condition (ii). 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 $TOP^P - BOT^Q(m_1 \pm \varepsilon, m_2 \ldots m_d)$ for some suitable $\varepsilon > 0$ 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 $\log(|TG|) + \log(|BG|)$ 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 $|TG|+|BG| \le n^d$, i.e. the algorithm requires no more than $2d \log n$ iterations. Each iteration involves a $(d-1)$-dimensional minimization and the four function lookups necessary to obtain $TOP^P - BOT^Q(m_1 \pm \varepsilon, m_2 \ldots m_d)$. As shown in section 3, each lookup can be carried out in no more than $2^{d+1} d \log n$ steps. We obtain a total time complexity

$$
\begin{aligned}
& T(d,n) \\
& \le d \log n \cdot (4 \cdot 2^{d+1} d \log n + T(d-1,n)) \\
& \le 2^{d+3} d^2 \log^2 n + d \log n T(d-1,n) \\
& \le 2^{d+3} d^2 \log^2 n + d \log n \, 2^{d+2} (d-1)^2 \log^2 n + d(d-1) \log^2 n T(d-2,n) \\
& \le \ldots \\
& \le \sum_{i=2}^{d-1} 2^{d+5-i} d^i \log^i n = O((2d)^{d-1} \log^{d-1} n).
\end{aligned}
$$

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

For $d=3$, the space and preprocessing requirements of this algorithm are as follows. The equations of the $O(n)$ faces of P and Q require space $O(n)$. For the multidimensional binary search one has to store (a) a subdivision of the x_1-axis into no more than $n+1$ partitions, and (b) a sequence of $O(n)$ boundary segments for each one of the partitions. The total space requirements are therefore $O(n^2)$. The preprocessing can be done in time $O(n^2)$ by means of a plane sweep as described in [Prep85], pp. 47-48.

For $d>3$, the data structures required to do the search are essentially the same as the ones required to do the point location as described in section 3. Therefore, 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 $T(d,n)$ with $S(d,n)$ space and $PP(d,n)$ preprocessing:

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

Proof : follows from the preceding discussion. □

5. Extensions

5.1. Unbounded Polyhedra

Clearly, there exist functions TOP^P and BOT^P for an unbounded polyhedron P, such that a hyperplane h intersects P if and only if the dual $D(h)$ lies between TOP^P and BOT^P. The question is how to define these functions in a way that allows to construct their graphs easily by dualization of the original polyhedron P. In the case of bounded polyhedra, we base our definition on the notion of vertex, which is obviously not sufficient for the unbounded case. One simple way to generalize our definitions of TOP^P and BOT^P,

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

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

to an unbounded polyhedron P, is to enhance V_P by some *virtual* vertices at infinity. In particular, let C_P denote a d-dimensional cube with edge length $E(C_P)$ that contains all vertices of P. The bounded polyhedron $P \cap C_P$ has a set of vertices $V_{P \cap C_P} = V_P \cup \bar{V}$, where $\bar{V}$ contains those vertices that are formed by intersections of C_P with edges of P. As $E(C_P)$ goes to infinity, so do the vertices in $\bar{V}$. The dual $D(\bar{v})$ of any vertex $\bar{v} \in \bar{V}$ goes towards a vertical hyperplane with a corresponding function $F_{D(\bar{v})}: E^{d-1} \to \pm\infty$. Now the functions $TOP^P, BOT^P: E^{d-1} \to E^1 \cup \{\pm\infty\}$ are defined as

$$TOP^P(x_1 \ldots x_{d-1}) = \lim_{E(C_r)\to\infty} \max_{v \in V_r \cup \bar{V}} F_{D(v)}(x_1 \ldots x_{d-1})$$

$$BOT^P(x_1 \ldots x_{d-1}) = \lim_{E(C_r)\to\infty} \min_{v \in V_r \cup \bar{V}} F_{D(v)}(x_1 \ldots x_{d-1})$$

Again, there is an isomorphism between the upper hull of P and the graph of TOP^P, as well as between the lower hull of P and the graph of BOT^P [Brow79]. Note that the virtual vertices are only a conceptual aid. They do not have to be taken into account when constructing the graphs by dualization. If the dual of P's upper hull does not yield a finite value $a_d = TOP^P(a_1 \ldots a_{d-1})$, then the functional value at $(a_1 \ldots a_{d-1})$ is assumed $+\infty$. Similarly, the default for $BOT^P(a_1 \ldots a_{d-1})$ is $-\infty$. The algorithms to detect intersections do not have to be modified, except for the possibility that TOP^P and BOT^P may now assume the values $\pm\infty$. A two-dimensional example is given in figure 5.1.

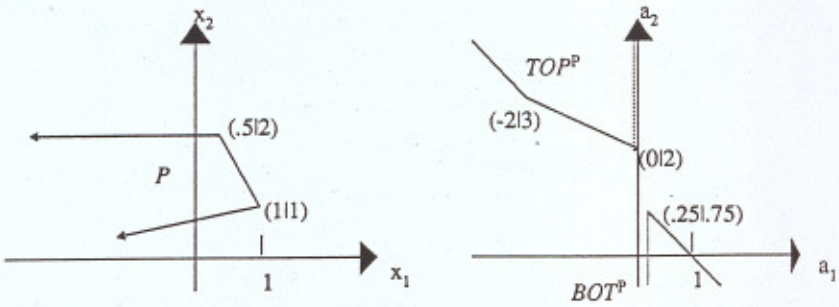


Figure 5.1

5.2. Vertical Hyperplanes

Vertical hyperplanes pose a problem for the dual scheme because they do not have a dual point with finite coordinates. However, for each vertical hyperplane h there is a virtual dual point at infinity. Let (h_n) denote a sequence of non-vertical hyperplanes that converges towards h, such that all hyperplanes h_n have the same $(d-2)$-dimensional point set Q in common (i.e. Q is the intersection of any two hyperplanes h_{n_1} and h_{n_2}).

Let $F_{h_n}(x_1 \ldots x_{d-1}) = a_1^n x_1 + \ldots + a_{d-1}^n x_{d-1} + a_d^n$. As described in section 3, $TOP^P(a_1^n \ldots a_{d-1}^n)$ is obtained as follows. First, one performs a $(d-1)$-dimensional point location in the projection of TOP^P's graph on the hyperplane $J: a_d = 0$ to find the partition $E \subseteq J$ that contains the point $(a_1^n \ldots a_{d-1}^n)$. Then, one computes the functional value $F_{D(v(E))}(a_1^n \ldots a_{d-1}^n)$, where $v(E)$ is the vertex of P that corresponds to the partition E.

Lemma 5.1: There is an $n_0 \in \mathbb{N}$ such that for all $n > n_0$ all duals $D(h_n)$ belong to the same partition $\bar{E} \subseteq J$.

Proof: Because Q is a subset of each hyperplane h_n, each dual point $D(h_n)$ lies on the dual non-vertical straight line $D(Q)$. Clearly, $(D(h_n))$ goes to infinity as n goes to infinity. On the other hand, each partition $E \subseteq J$ is convex, and the number of partitions is finite. From there, the lemma follows. □

In order to check h for intersection with some polyhedron P, one can now proceed similarly as in the case of a non-vertical hyperplane. The partition $\bar{E}$ can be obtained by a point location. Then, one computes the two limits $\lim_{n\to\infty}(TOP^P(a_1^n \ldots a_{d-1}^n) - F_{D(v(\bar{E}))}(a_1^n \ldots a_{d-1}^n))$ and $\lim_{n\to\infty}(F_{D(v(\bar{E}))}(a_1^n \ldots a_{d-1}^n) - BOT^P(a_1^n \ldots a_{d-1}^n))$. h intersects P if and only if both limits are greater or equal zero. Moreover, h supports P if and only if at least one of the limits is finite.

6. 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 space 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. In particular, we suspect that lower bounds for space and preprocessing may be achieved at the expense of slightly higher time bounds for the detection algorithms.

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.

[Chaz87] Chazelle, B. and D. P. Dobkin, Intersection of Convex Objects in Two and Three Dimensions, *J. ACM* **34**, 1 (Jan. 1987), pages 1-27.

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

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

[Dobk80] Dobkin, D. P. and J. I. Munro, Efficient uses of the past, in *Proc. 21st Annual Symposium on Foundations of Computer Science*, Syracuse, NY, 1980, pages 200-206.

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

[Edel86a] Edelsbrunner, H., L. J. Guibas, and J. Stolfi, Optimal point location in a monotone subdivision, in *SIAM J. Comput.* **15**, 1986, pages 317-340.

[Edel86b] Edelsbrunner, H., J. O'Rourke, and R. Seidel, Constructing arrangements of lines and hyperplanes with applications, in *SIAM J. Comput.* **15**, 1986, pages 341-363.

[Gunt87] Gunther, O., *Representation of geometric data*, Ph.D. dissertation, University of California, Berkeley, Ca., in preparation, December 1987.

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

[Megi84] Megiddo, N., Linear programming in linear time when the dimension is fixed, *J. ACM* **31**, 1 (Jan. 1984), pages 114-127.

[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.