Determining the Separation of Preprocessed Polyhedra - A Unified Approach

David P. Dobkin *
Computer Science Department
Princeton University

David G. Kirkpatrick †
Computer Science Department
University of British Columbia

Abstract

We show how (now familiar) hierarchical representations of (convex) polyhedra can be used to answer various separation queries efficiently (in a number of cases, optimally). Our emphasis is i) the uniform treatment of polyhedra separation problems, ii) the use of hierarchical representations of primitive objects to provide implicit representations of composite or transformed objects, and iii) applications to natural problems in graphics and robotics.

Among the specific results is an $O(\log |P| \cdot \log |Q|)$ algorithm for determining the separation of polyhedra P and Q (which have been individually preprocessed in at most linear time).

1 Introduction and background

Given pairs of geometric objects A and B the problems of *testing* for non-empty intersection ($A \cap B \neq \emptyset$), together with the *construction* of $A \cap B$ (when $A \cap B \neq \emptyset$) or a description of their *separation* (when $A \cap B = \emptyset$), comprise some of the most fundamental issues in computational geometry [24,20,14] The intrinsic complexity of these tasks is not yet fully understood even for the simplest of geometric objects. We continue here our earlier investigations [9,10,11] of these questions with respect to convex polytopes in two and three dimensions.

One of the essential themes of our earlier work (including also [19]) has been the introduction and exploitation of hierarchical representations of polytopes in this setting. In [9] two (essentially dual) hierarchical representations for (convex) polygons and polyhedra were introduced and some of their basic properties (such as the efficient response to extremal queries)

*Supported in part by NSF Grant Number CCR87-00917 and a Guggenheim Fellowship.

†Supported in part by Natural Sciences and Engineering Research Council Grant A3583 and a BCASI Fellowship.

were set out. (See also [20] for a discussion of these same basic ideas). This representation has been exploited in a series of subsequent papers dealing with separation of polytopes [11], generalized extremal queries and applications [16], intersection of convex and non-convex polyhedra [21], intersection of convex bodies with curved edges and faces *i.e. splinegons and splinehedra* [12,13,25], parallel algorithms for manipulation of polytopes [6,7], applications in computer graphics [8], on line maintenance of polygons [1] and construction of polyhedral intersections [3].

Of particular significance in the background of the present paper is the result of Chazelle and Dobkin [4,5] that (given suitable - and often simple - preprocessing of the objects) intersection testing is less costly than intersection construction for convex objects. That this is also true of the more general problem of polyhedral separation (in which a closest pair of points - a *witness* to the separation - on the given objects is constructed) was subsequently observed. Schwartz [26] and more recently Edelsbrunner [15] and Chan and Wang [2] studied the separation problem for (preprocessed) polygons in the plane. The latter two papers present (optimal) $O(\log |P| + \log |Q|)$ algorithms for finding the separation of arbitrary polyhedra P and Q. Among the main technical contributions of this paper is another optimal algorithm for finding the separation of polygons that (unlike earlier algorithms) works even with a somewhat less explicit representation of polygons. In addition, we present an $O(\log |P| \cdot \log |Q|)$ algorithm for finding the separation of arbitrary polyhedra P and Q. This algorithm assumes that the polyhedra have been individually preprocessed (in at most linear time), and hence provides a generalization of the linear upper bound for determining the separation of (unpreprocessed) polyhedra [11].

In addition to these technical contributions we wish to emphasize the following:

(i) our algorithms provide a *uniform* treatment of polyhedra separation problems; we exploit the fact that we can reduce both the dimension and the combinatorial complexity of the objects under consideration within the same representation;

(ii) our hierarchical representations of polyhedra P provide implicit representations of associated polygons (formed, for example, by projection of P) and of composite objects formed (for example, by extrusion, intersection or convolution) of polyhedra; and

(iii) many natural problems in graphics and robotics (such as occlusion, collision detection and path planning) which involve convex objects can be reduced to instances of geometric intersection/separation queries for implicitly defined polyhedra.

2 Hierarchical representations of polygons and polyhedra

In this section we set out some of our notation and basic definitions concerning polytopes, along with a review of the definition and fundamental properties of hierarchical representations of (low dimensional) polytopes. See [14,20,17] for more detailed treatments.

2.1 Basic definitions

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

If p and $q \in \Re^k$ we denote by L_{pq} the *line* passing through p and q and R_{pq} the *ray* from p passing thru q. If $a \in \Re^k - \{0\}$ and $c \in \Re$ then the set $H(a,c) = \{x \in \Re^d | \langle x, a \rangle = c\}$ is called an *oriented hyperplane* in $\Re^k$. Here, $\langle x, y \rangle$ represents the inner product of vectors x and y. A hyperplane $H(a,c)$ defines two closed half spaces $H^+(a,c) = \{x \in \Re^d | \langle x, a \rangle \geq c\}$ and $H^-(a,c) = \{x \in \Re^d | \langle x, a \rangle \leq c\}$. We say that a hyperplane $H(a,c)$ *supports* a polyhedron P if $H(a,c) \cap P \neq \emptyset$ 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 (resp. 0) are called *edges* (resp. *vertices*) of P. We denote by $V(P)$ the set of vertices of P.

The *1-skeleton* (hereafter simply *skeleton*) of a polytope P is the graph whose vertices (resp. edges) are the vertices (resp. 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 denote by $|P|$ the total number of faces of P (of all dimensions).

We find it convenient to refer to polytopes with size bounded by some fixed constant as *elementary polytopes*. Note that the separation of elementary polytopes can be computed in constant time.

Two convex objects (be they polytopes or linear subspaces) which intersect do so in a convex object. If objects do not intersect, we say that they are separated. For convex objects P and Q, we define their separation $\sigma(P,Q)$ as the distance between their nearest points (not necessarily unique). This separation can be characterized in various ways, as we shall see below. 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 $|p - q| = \sigma(P,Q)$.

2.2 Hierarchical representations

We exploit the same hierarchical representations of polygons and polyhedra introduced in [9] and subsequently studied in [10,11]. As in our earlier papers we describe such representations as abstract data types; see [14] for details of an elegant implementation.

Let P be a d-polytope with vertex set $V(P)$. A sequence of polytopes $hier(P) = P_1, \cdots, P_k$ 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) $V(P_{i+1}) \subset V(P_i)$, for $1 \leq i < k$, and

(iv) the vertices of $V(P_i) - V(P_{i+1})$ form an independent set in P_i, for $1 \leq i < k$.

We refer to k, $\sum_{i=1}^{k} |P_i|$ and $max_i max_{v \in V(P_i) - V(P_{i+1})} deg(v, P_i)$, as the *height*, *size* and *degree* of $hier(P)$ respectively. A hierarchical representation is said to be *compact* if it has height at most $c \log |P|$, size at most $c|P|$, and degree at most c, for some fixed constant c. We recall from [11] that

Property 2.1. (a) Given a standard representation of a 2- or 3- polytope P a compact hierarchical representation of P can be constructed in time $O(|P|)$.

(b) If $P_1, \cdots, P_k$ is a hierarchical representation of P and H is any (oriented) hyperplane such that $P_{i+1} \subset H^+$, for some i, then either i) $P_i \subset H^+$ or ii) there exists a unique vertex $v \in V(P_i)$ such that $v \in H^-$.

Because of property 2.1 (b) it is natural to (explicitly) endow hierarchical representations with some additional structure to reflect the (implicit) relationship between successive elements. Specifically we add the following to the definition of hierarchical representations:

(v) each facet F of P_{i+1} that is not a facet of P_i has associated with it a pointer to the (unique) vertex of P_i that lies in the halfspace opposite to P_{i+1}, with respect to the hyperplane supporting F.

The following is a direct consequence of point iii) of the definition of hierarchical representations.

Property 2.2. If $P_1, \cdots P_k$ is a hierarchical representation of the polytope P and $P_i = \cap_{j \in J} H_j$, where $\{H_j | j \in J\}$ is the set of hyperplanes supporting facets of P_i, then $P_{i-1} \subset \cup_{k \in J}(\cap_{j \in J-\{k\}} H_j)$.

Remark. Property 2.2 restricts the "growth" of polytopes as we move up the hierarchy. It also suggests that hierarchical representations have a natural dual formulation. In fact these dual representations (called inner and outer representations) were presented in parallel in [9].

3 Separation of preprocessed polygons

3.1 Separation of polygons and linear subspaces

The results in [9] for detecting the intersection of polygons and linear subspaces (points or lines) generalize very naturally to the problem of determining their separation. The basic idea is to maintain as we step through a compact hierarchical representation of polygon P, the pair of points (r_i, s_i) which realizes the separation of P_i (the current approximation of P) and the subspace S. It suffices to show how this closest pair can be updated in $O(1)$ time per step. (Note that when $r_i = s_i$, that is intersection has been detected, we can continue stepping through the hierarchical representation of P to actually construct the intersection [9].)

Suppose (r_i, s_i) realizes $\sigma(P_i, S)$ and assume that $r_i \neq s_i$ (otherwise we are done). Let H_p be the (oriented) line normal to $r_i s_i$ that supports P_i at r_i. Then $P_{i-1} = (P_{i-1} \cap H_p^+) \cup (P_{i-1} \cap H_p^-)$ and

$$\sigma(P_{i-1}, S) = \min \left\{ \begin{array}{l} \sigma(P_{i-1} \cap H_p^+, S) \\ \sigma(P_{i-1} \cap H_p^-, S) \end{array} \right\}$$

But $\sigma(P_{i-1} \cap H_p^+, S)$ is realized by the pair (r_i, s_i) and $P_{i-1} \cap H_p^-$ is elementary, since the representation is compact, and hence $\sigma(P_{i-1} \cap H_p^-, S)$ (and its realization) can be determined in $O(1)$ time. Thus we have shown the following:

Theorem 3.1. The separation $\sigma(P, S)$ (and its realization) of a polygon P and a linear subspace S can be determined in $O(\log |P|)$ time from a compact hierarchical representation of P.

3.2 Separation of two polygons

Let P and Q be polygons and assume that compact hierarchical representations for both P and Q are available. Recall that Edelsbrunner [15] has given an $O(\log|P| + \log|Q|)$ algorithm to determine $\sigma(P,Q)$ and its realization using an array (and hence binary searchable) representation of both P and Q. Our objective in this subsection is to outline another $O(\log|P|+\log|Q|)$ algorithm using hierarchical representations of P and Q. Our motivation is threefold

(i) Edelsbrunner's algorithm requires a preprocessing step to check that $\sigma(P,Q) \neq 0$. We want to show that the full problem is just a direct generalization of this subproblem.

(ii) We wish to provide a unified approach to geometric intersection problems; our algorithm for $\sigma(P,Q)$ follows exactly the same (walking through the hierarchical representations) approach that we use for both simpler (Q is a linear subspace) and more complex (both P and Q are polytopes) cases.

(iii) Though Edelsbrunner's underlying data structure is simpler it is, nevertheless, an *explicit* representation of a polygon. Our algorithm applies even in a situation in which the input polygons are only known implicitly that is, via a sequence of approximations the first of which may contain no information about the actual edges of the final polygon), see section 3.3.

As in the case of the previous section we maintain a pair (r_i, s_j) which realizes the separation of approximations P_i and Q_j (of P and Q respectively) and show how this pair can be updated efficiently as we step through one or the other of the two hierarchies. To update the pair in $O(\log|P_i| + \log|Q_j|)$ time (even if we take a step simultaneously in both hierarchies) is straightforward. We illustrate exactly this approach (which yields an $O(\log|P| \cdot \log|Q|)$ algorithm for the case where P and Q are polytopes, in section 5.3. A more efficient updating scheme seems to require a somewhat subtler approach - in effect a more powerful invariant.

Suppose that (r,s) realizes the separation of P_i and Q_j. Suppose further that it has been determined that $\sigma(P,Q) = \sigma(P \cap W_r, Q \cap W_s)$, where W_r is the concave wedge formed by the vertices r_a and r_b (separated by at most 8 vertices on P_i) and the point r and W_s is the concave wedge formed by the vertices s_a and s_b (separated by at most 8 vertices on Q_j) and the point s (see Figure 1). Since $\sigma(P \cap (W_r - W_r'), Q \cap (W_s - W_s')) \geq |r-s|$ it follows that either $\sigma(P,Q) = \sigma(P \cap W_r', Q \cap W_s)$ or $\sigma(P,Q) = \sigma(P \cap W_r, Q \cap W_s')$, where W_r' is the concave wedge formed by the vertices r_+ and r_- (each separated by one vertex on P_i from point r) and the point r and W_s' is the concave wedge formed by the vertices s_+ and s_- (likewise separated by one vertex on Q_j from point s) and the point s. Furthermore (using a fairly involved case analysis, the details of which will appear in an expanded version of this paper), which of these is the case can be determined in $O(1)$ time. Suppose, without loss of generality, that $\sigma(P,Q) = \sigma(P \cap W_r', Q \cap W_s)$. Since $P_i \cap W_r'$ contains at most five vertices of P_i it follows that it contains at most nine vertices of P_{i-1}. Hence we can compute $\sigma(P_{i-1}, Q_j) = \sigma(P_{i-1} \cap W_r', Q_j \cap W_s)$ together with a realization (r', s') in $O(1)$ time. Since r' is restricted to $P_{i-1} \cap W_r'$ and s' is restricted to $Q_j \cap W_s$ the wedges W_r', formed by r_+, r_- and r', and W_s, formed by s_a, s_b and s' satisfy the invariant.

In this fashion the separation of P_i and Q_j can be updated in $O(1)$ time, and so the separation of P and Q can be reduced to a constant number of polygon/linear subspace separation queries in $O(\log|P| + \log|Q|)$ time. We summarize this result in the following.

Theorem 3.2. The separation $\sigma(P,Q)$ (and its realization) of polygons P and Q can be determined in $O(\log|P| + \log|Q|)$ time from their hierarchical representations.

3.3 Separation of implicitly defined polygons

As we mentioned earlier it is of interest to ask the extent to which our algorithm for polygon separation depends on explicit knowledge of the input polygons. Inspection of the algorithm reveals that two basic properties of hierarchical representations are used:

(i) if a wedge cuts off a segment of the boundary of P_i of size s then the same wedge cuts off a segment of P_{i-1} of size $O(s)$.

(ii) in the (omitted) case analysis the property that the vertices of P_i are vertices of P_{i-1} is used to restrict the growth of P_{i-1} in terms of P_i. Specifically, we use the fact (an immediate consequence of Property 2.2) that if P_i has vertices $v_i \cdots v_t$ and H_j^+ denotes the halfspace through v_j and v_{j+1} supporting P, then

$$P_{i-1} - P_i \subset \cup_j (H_{j-1}^+ \cap H_j^- \cap H_{j+1}^+)$$

Properties i) and ii) (or slight variants) would, of course, continue to hold if P_i is formed from P_{i-1} by removing connected clusters of points each of size $O(1)$. In fact appropriate analogues hold if P_i is formed from P_{i-1} by replacing segments of length $O(1)$ by new segments of length $O(1)$ (so long as $P_i \subset P_{i-1}$). In the next section we present a new (more general) definition of hierarchical representation of polytopes motivated by these observations.

4 Implicitly defined polygons and their representation

In this section we show that certain natural operations on polyhedra that give rise to (implicit) polygons are well reflected by our hierarchical representations in the sense that the representation of the polyhedron embodies an (implicit) representation of the associated polygon. To achieve the greatest generality we relax our definition of hierarchical representations of polytopes appropriately.

4.1 Hierarchical representations with granularity greater than one

As we have seen, conditions (iii) and (iv) of our definition of hierarchical representations imposes a locality of change property (essentially Property 2.1(b) and Property 2.2) on successive elements of such representations. This locality of change is preserved (along with a relaxation of the corresponding properties) if the definition itself is relaxed by replacing conditions (iii) and (iv) by the following:

(iii′) The vertices of $V(P_{i+1}) - V(P_i)$ induce a subgraph of P_{i+1} each of whose connected components has size bounded by some constant g; and

(iv′) The vertices of $V(P_i) - V(P_{i+1})$ induce a subgraph of P_i each of whose connected components has size bounded by g.

We refer to the bound g as the *granularity* of the associated representation. Note that our standard definition describes representations of granularity 1.

4.2 Projections of polyhedra

Given a polyhedron P and plane H and a point $p \notin H$, the *projection* of P onto H through p, denoted $proj_H(P,p)$ is the set $\{q \in H | R_{pq} \cap P \neq \emptyset\}$. The projection of P is a (possibly unbounded) polygon whose vertices are projections of vertices of P. It is natural to ask how faithfully a hierarchical representation of P represents an arbitrary projection of P. More concretely, can one answer queries about a projection of P using only the representation of P as efficiently as one could using an explicit representation of the projection? The answer to this, and similar such questions, is yes; it suffices to ask how the projections of successive elements in the hierarchical representation of P relate to one another.

Lemma 4.1. If $P_1, P_2, \cdots P_k$ is a hierarchical representation with granularity g of the polyhedron P, then $P'_1, P'_2 \cdots P'_k$, where $P'_i = proj_H(P_i, p)$, is a hierarchical representation with granularity g of $proj_H(P,p)$.

Proof. It suffices to observe that each edge e' of P'_i corresponds to an edge e of P_i. Edge e separates two faces f_L and f_R of P_i and (by Property 2.1(b)) each of these faces has associated with it (at most) one vertex of P_{i-1} that lies on the opposite side of the plane supporting this face, from P_i. At most one of these vertices lie on the opposite side of the plane through p and e, from P_i and hence at most one vertex of P_{i-1} projects onto the opposite side of e' from P'_i. Since the hierarchical representation of P provides access (starting from e') to each such vertex in $O(1)$ time, the hierarchical representation of P contains an implicit hierarchical representation of $proj_H(P,p)$. □

4.3 Plane intersections of polyhedra

If P is a polyhedron and H is a plane, then the intersection $P \cap H$ is a (convex) polygon in H. As before we wish to demonstrate that a hierarchical representation of P serves as an efficient implicit representation of all polygons formed in this way. Note that the vertices of $P \cap H$ can not be put into correspondence with vertices of P (as in the case of projections). Indeed, vertices of $P \cap H$ arise from the intersection of *edges* of P with H, and edges of $P \cap H$ arise from the intersection of *faces* of P with H. However, using our relaxation of the definition of a hierarchical representation of polygons, we can establish the desired linkage. Specifically,

Lemma 4.2. If $P_1, \cdots, P_k$ is a hierarchical representation with bounded degree of the polyhedron P, then $P'_1, \cdots, P'_k$, where $P'_i = P_i \cap H$, is a hierarchical representation of bounded granularity of the polygon $P \cap H$.

Proof. Suppose the hierarchical representation of P has degree d. By convexity $P'_i \subset P'_{i-1}$. Hence it suffices to show that, for each i, every sequence of d vertices on the boundary of P'_i contains at least one vertex of P'_{i-1}, and every sequence of d vertices on the boundary of P'_{i-1} contains at least one vertex of P'_i. But a vertex of P'_{i-1} does not appear on P'_i if and only if it corresponds to an edge of P_{i-1} that is removed in constructing P_i. Any sequence of such vertices must correspond to edges of P_{i-1} that are eliminated by the removal of a single vertex. Hence such sequences are restricted in length by the degree of the removed vertex. Since vertices of P'_i that do not belong to P'_{i-1} correspond to edges introduced in the construction of P_i from P_{i-1} and all such edges that share a face in P_i arise from the removal of the same vertex of P_{i-1}, sequences of vertices P_i' that do not belong to P_{i-1}' are restricted in length by the degree of the removed vertex. □

In the next section we will see how Lemma 4.2 can be applied to give efficient algorithms for the separation of polyhedra and polygons and for arbitrary pairs of polyhedra.

4.4 Application: occlusion of polyhedra

A familiar problem in computer graphics is to determine, for a scene consisting of two or more objects, whether a particular object A when viewed from a point p occludes another object B. This question is directly reducible to the question does $proj_H(A, p)$ intersect $proj_H(B, p)$, for an arbitrary plane H that does not include p. By Lemma 4.2 we know that hierarchical representations of convex polyhedra P and Q induce (implicit) hierarchical representations of $proj_H(P, p)$ and $proj_H(Q, p)$. By Theorem 3.2 and the remarks of section 3.3 such implicit representations suffice to test for the intersection of $proj_H(P, p)$ and $proj_H(Q, p)$ (in fact, to determine their separation). Hence we have the following:

Theorem 4.1. Given a point p and two disjoint polyhedra P and Q, whether P occludes Q from viewpoint p can be determined in $O(\log|P| + \log|Q|)$ time, from compact hierarchical representations of P and Q.

Proof. Choose any plane H with $p \notin H$. Polyhedron P occludes polyhedron Q from viewpoint p if and only if $proj_H(P, p) \cap proj_H(Q, p) \neq \emptyset$ and for any point $r \in proj_H(P, p) \cap proj_H(Q, p)$ the line segment $P \cap L_{pr}$ lies between point p and line segment $Q \cap L_{pr}$ on the line L_{pr}. Since $|proj_H(P, p)| \leq |P|$ and $|proj_H(Q, p)| \leq |Q|$ the intersection of $proj_H(P, p)$ and $proj_H(Q, p)$ can be tested (and if non-empty a witness r produced) in $O(\log|P| + \log|Q|)$ time. By convexity and disjointness a non-empty intersection implies that either P occludes Q or Q occludes P. Since $P \cap L_{pr}$ and $Q \cap L_{pr}$ can be constructed in $O(\log|P| + \log|Q|)$ time from hierarchical representations of P and Q [9], the entire occlusion problem can be solved in this same time. □

Remark. Consider the cone defined by polyhedron P and point p, $cone(P, p) = \{q | L_{pq} \cap P \neq \emptyset\}$. Theorem 4.1 can be interpreted as asserting that detecting the intersection of $cone(P, p)$ with polyhedron Q can be determined in $O(\log|P| + \log|Q|)$ time. If p is the point at infinity in direction d, $cone(P, p)$ corresponds to the volume of space swept out as P is translated (in an unbounded fashion) along direction d. Thus Theorem 4.1 asserts that it is possible to efficiently detect collision (or more generally to determine the minimum separation realized) between P and Q as polyhedron P is translated along some (unbounded) vector. Similar results for collision detection under bounded (and semi-unbounded) translation are discussed in section 6.

5 Separation of preprocessed polyhedra

5.1 Separation of polyhedra and linear subspaces

We begin by noting that Theorem 3.1, concerning the separation of polygons and linear subspaces, can be directly generalized to apply to hierarchically represented polyhedra. Not only is the methodology the same, in fact the proof of Theorem 3.1 was presented in such a way that (reinterpreting P as an arbitrary polyhedron and S as an arbitrary 3-dimensional linear subspace, i.e. point, line or plane) it proves the following as well:

Theorem 5.1. The separation $\sigma(P,S)$ (and its realization) of a polyhedron P and a linear subspace S can be determined in $O(\log|P|)$ time from a hierarchical representation of P.

Remark. In the event that $\sigma(P,S) \neq 0$ and S is a point or a line it is straightforward to *construct* $P \cap S$ within the same time bound (cf. [9]). If S is a plane this is clearly impossible. However, as we noted in section 4.2, an implicit representation of $P \cap S$ (sufficient to answer other intersection queries) is readily available in this case.

Note that the tools developed in section 4 suggest an alternative approach to the separation of polytopes and either lines or planes (actually points as well if we permit ourselves to dualize). For example, the separation of polyhedron P and line L is just the separation of the projections of P and L onto a plane orthogonal to L, from a point at infinity. Since representations of these projections are implicit in representations of the 3-dimensional counterparts, a reduction to the two dimensional separation algorithms of section 3.1 is immediate.

5.2 Separation of polyhedra on a given plane

Suppose we wish to determine the separation of those parts of two polyhedra P and Q that intersect a common plane H. By the results of section 4.2 this separation $\sigma(P \cap H, Q \cap H)$ and its realization can be determined in $O(\log|P| + \log|Q|)$ time. (As an immediate corollary we get the same time bound for determining the separation of a polyhedron P and a polygon Q, in the plane of the polygon.)

5.3 Separation of arbitrary polyhedra

We turn now to the general case where P and Q are arbitrary polyhedra and we wish to determine $\sigma(P,Q)$. Our approach is similar in spirit to the polygon/linear subspace and polygon/polygon separation algorithms of sections 3.1 and 3.2 respectively. It can also be viewed as a refinement of the (unpreprocessed) polyhedron/polyhedron separation algorithm presented in [10].

Let $P_1, \cdots, P_r$ be a hierarchical representation of P and $Q_1 \cdots Q_s$ be a hierarchical representation of Q, and assume without loss of generality that $r \leq s$. Since $\sigma(P_r, Q_r)$ and its realization can be determined by a constant number of polyhedron/linear subspace queries (and hence $O(\log|Q_r|)$ time in total), it suffices to show how to efficiently update the pair (p_i, q_i), $p_i \in P_i$ and $q_i \in Q_i$, realizing $\sigma(P,Q)$, as i is decremented from r down to 1.

We can assume that $p_i \neq q_i$ (otherwise, it suffices to set $p_{i-1} = q_{i-1}$). Let H_p and H_q be planes normal to the line $L_{p_i q_i}$, such that H_p supports P_i at p_i and H_q supports Q_i at q_i (see Figure 2). Then

$$P_{i-1} = (P_{i-1} \cap H_p^+) \cup (P_{i-1} \cap H_p^-)$$

$$Q_{i-1} = (Q_{i-1} \cap H_q^+) \cup (Q_{i-1} \cap H_q^-)$$

and

$$\sigma(P_{i-1}, Q_{i-1}) = min \left\{ \begin{array}{c} \sigma(P_{i-1} \cap H_p^+, Q_{i-1} \cap H_q^+) \\ \sigma(P_{i-1} \cap H_p^-, Q_{i-1}) \\ \sigma(P_{i-1}, Q_{i-1} \cap H_q^-) \end{array} \right\}$$

But since $\sigma(P_{i-1} \cap H_p^+, Q_{i-1} \cap H_q^+)$ is realized by the pair (p_i, q_i), and $P_{i-1} \cap H_p^-$ and $Q_{i-1} \cap H_q^-$ are both elementary, it follows that $\sigma(P_{i-1}, Q_{i-1})$ can be constructed using $O(1)$ polyhedron/linear subspace separation queries, using a total of $O(\log|P_{i-1}| + \log|Q_{i-1}|) = O(\log(\max\{|P|, |Q|\}))$ time. Since the entire process completed in $r = O(\log(\min\{|P|, |Q|\}))$ reduction steps, we have the following:

Theorem 5.2. The separation $\sigma(P, Q)$ (and its realization) of polyhedra P and Q can be determined in $O(\log|P| \cdot \log|Q|)$ time from their hierarchical representations.

Remark. The complexity bound in Theorem 4.2 is comparable to that achieved for detecting polyhedron/polyhedron intersections in [10]. However, the earlier algorithms used a rather cumbersome representation of polyhedra that requires $O(|P|^2)$ space (and preprocessing time). It is, however, interesting to note that the general approach for detecting polyhedral intersections used in this earlier algorithm, namely testing in a binary searching fashion - for the intersection of a succession of polyhedral cross sections, can be emulated in a straightforward way using (slightly augmented) hierarchical representations. Specifically, if we record, as part of our representation of a polyhedron P, the sequence of vertices of P sorted along an arbitrary axis, then we have available, in an implicit form, all cross sections of P normal to this axis. Using these we can perform a sweep or binary search along this axis, which lends itself to the implementation of another class of algorithms.

6 Separation of implicitly defined polytopes

In this section we outline some of the evidence for our claim that our hierarchical representations lend themselves not only to the representation of geometric primitives but also to composite or transformed objects formed (in natural ways) from those primitives. Since many natural operations on polyhedra preserve convexity, it is natural to ask if the hierarchical representation of the operands somehow embody an (implicit) hierarchical representation of the result. This turns out to be the case for the operations of extrusion, intersection, and convolution.

6.1 Extrusions of polygons and polyhedra

If P is a polytope and v is a vector then $extr(P, v)$ is the polytope formed by translating (extruding) P along the vector v. This is a special case of convolution (which we discuss in section 6.3) but is interesting and instructive to study in its own right. It is clear that $extr(P, v)$ is a polytope. In fact we claim the following:

Lemma 6.1. If $P_1, P_2, \cdots, P_k$ is a hierarchical representation of the polyhedron P, then $P_1', P_2', \cdots, P_k'$, where $P_i' = extr(P_i, v)$, is a relaxed hierarchical representation of $extr(P, v)$.

As a corollary of lemma 6.1 (and the results of sections 3 and 5) we have the following:

Corollary 6.2. Given polygons (respectively polyhedra) P and Q and a vector v, the separation $\sigma(extr(P, v), Q)$ between the extrusion of P and Q can be determined in $O(\log|P| + \log|Q|)$ (respectively $O(\log|P| \cdot \log|Q|)$) time, from hierarchical representations of P and Q.

6.2 Common intersections and convolutions of polyhedra

The intersection $P \cap Q$ of two polyhedra P and Q is a polyhedron. It is natural to ask the extent to which hierarchical representations of P and Q embody a representation of $P \cap Q$. Chazelle [3] has recently shown that an explicit description of $P \cap Q$ can be constructed from hierarchical representations of P and Q in $O(|P| + |Q|)$ time. It turns out (as we claim below) that sufficient information concerning $P \cap Q$ is implicit in the representations of P and Q, that with no additional preprocessing many queries concerning $P \cap Q$ can be answered as efficiently as they can with an explicit representation of $P \cap Q$.

The *convolution* $P * Q$ of polyhedra P and Q is defined by

$$P * Q = \{p + q | p \in P \& q \in Q\}$$

(where points are added like their associated vectors). (See [18] for a careful treatment of convolutions in the context of geometric intersection problems.) Note that $P*Q = \cup_{q \in Q} extr(P, q)$, provided the origin belongs to Q.

Though the convolution $P * Q$ of polygons P and Q can be constructed simply from (and is linear in the size of) P and Q, this is not the case for polyhedra. It is well known that the convolution of two polyhedra of size n can have size $\Theta(n^2)$. As a consequence there is even more motivation in this case to avoid explicit construction of $P * Q$ when information implicit in the representations of P and Q suffice. Our results concerning common intersections and convolutions of polyhedra can be (weakly) summarized as follows:

Theorem 6.3. Given compact hierarchical representations of P, Q and R, both $\sigma(P \cap Q, R)$ and $\sigma(P * Q, R)$ can be determined in time polylogarithmic in $n = max(|P|, |Q|, |R|)$.

6.3 Applications

We have already observed that our results have application to certain basic questions in computer graphics. We close this section with some remarks concerning additional applications of our results on determining the separation of implicitly defined polyhedra. We will state the applications in terms of polygons; similar results hold for polyhedra.

1. If we have two polygons positioned in the plane and one is translated in a specified direction a specified distance it may or may not collide with the other polygon. The techniques of this paper suffice to give logarithmic time answers to all of the following questions:

   (i) Do the polygons collide?

   (ii) if so, at what distance (in the translation) do they collide and what are the points of contact?

   (iii) if not, at what point in the translation do they come closest to one another and what are the points that realize this closest distance?

2. If we have three polygons P, Q and R positioned in the plane it may or may not be possible by a sequence of translations to move polygon R between polygons P and Q. It is well known that this is reducible to a question about convolutions, specifically the translation is possible if and only if $(P * R) \cap (Q * R) = \emptyset$. Thus we are able to provide polylogarithmic time solutions to the following queries:

   (i) Is the translation possible?;

   (ii) if so, what is a description of the translation path that maximizes the clearance?;

   (iii) if not, what are the points of contact when R becomes "stuck"?

References

[1] Avis,D., ElGindy, H. and Seidel, R., A simple online algorithm for convex polyhedra, in *Computational Geometry*,G. Toussaint, ed., Springer Verlag, 1985, 23-42.

[2] Chan, F. and Wang, C., 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, August, 1982.

[3] Chazelle, B., An optimal algorithm for intersecting three-dimensional convex polyhedra, IEEE Sympos. on Found. of Comp. Sci. 30 (1989), 586-591.

[4] Chazelle, B. and Dobkin, D. Detection is easier than computation, *ACM Sympos. on Theory of Comput.* 12 (1980), 146-153.

[5] Chazelle, B. and Dobkin, D. Intersection of convex objects in two and three dimensions, *JACM* 34 (1987), 1-27.

[6] Dadoun, N. and Kirkpatrick, D., Parallel construction of subdivision hierarchies, *JCSS* 39 (1989), 153-165.

[7] Dadoun, N. and Kirkpatrick, D., Cooperative subdivisions algorithms with applications, Allerton Conference, 1989, to appear.

[8] Dadoun, N., Kirkpatrick, D. and Walsh, J. The geometry of beam tracing, *ACM Sympos. on Comp. Geom.* 1 (1985), 55-61.

[9] Dobkin, D. and Kirkpatrick, D., Fast detection of polyhedral intersections, *Lecture Notes in Computer Science* 140 (1982) 154-165.

[10] Dobkin, D. and Kirkpatrick, D., Fast detection of polyhedral intersection, *TCS* 27 (1983), 241-253.

[11] Dobkin, D. and Kirkpatrick, D., A linear algorithm for determining the separation of convex polyhedra, *J. Algorithms* 6 (1985) 381-392.

[12] Dobkin, D. and Souvaine, D., Detecting the intersection of convex objects in the plane, Princeton University Technical Report CS-TR-231-89, October, 1989.

[13] Dobkin, D. and Souvaine, D., Detecting the intersection of convex objects in 3 dimensions, in preparation.

[14] Edelsbrunner, H., *Algorithms in Combinatorial Geometry*, Springer Verlag, Berlin, 1987.

[15] Edelsbrunner, H., Computing the extreme distance between two convex polygons, *J. Algorithms* 6 (1985), 213-224.

[16] Edelsbrunner, H. and Maurer, H., Finding extreme points in three dimensions and solving the post-office problem in the plane, *IPL* 21 (1985), 39-47.

[17] Grunbaum, B., *Convex Polytopes*, John Wiley&Sons, London, 1987.

[18] Guibas, L., Ramshaw, L., and Stolfi, J., A kinetic framework for computational geometry, *IEEE Sympos. Found. Comp. Sci.* 24 (1983), 100-111.

[19] Kirkpatrick, D., Optimal search in planar subdivisions, *SIAM J. Comput.* 12 (1983), 28-35.

[20] Mehlhourn, K., *Data Structures and Algorithms 3: Computational Geometry* Springer Verlag, Berlin, 1984.

[21] Mehlhourn, K. and Simon, K., Intersecting two polyhedra one of which is convex, Univ. Saarland Technical report, Saarbrucken, West Germany, 1986.

[22] Mueller, D. and Preparata, F., Finding the intersection of two convex polyhedra, *TCS* 7 (1978) 217-236.

[23] Preparata, F. and Hong,S., Convex hulls of finite sets of points in two and three dimensions, *CACM* 20 (1977), 87-93.

[24] Preparata, F. and Shamos, M., *Computational Geometry – an Introduction* Springer Verlag, New York, NY, 1985.

[25] Souvaine, D., Computational geometry in a curved world, Ph.D. dissertation, Princeton University, 1986.

[26] Schwartz, J., Finding the minimum distance between two convex polygons, *IPL* 13 (1981), 168-170.