A Simple Primal Algorithm for Intersecting 3-Polyhedra in Linear Time

by

Andrew K. Martin
(Email: amartin@cs.ubc.ca)

Technical Report 91-16
July 1991

Department Of Computer Science
University of British Columbia
2366 Main Mall, Vancouver, B.C.
CANADA V6T 1Z4

Abstract

This thesis presents, in full, a simple linear time algorithm for intersecting two convex 3-polyhedra P and Q. This differs from the first such algorithm — due to Chazelle — in that it operates entirely in primal space, whereas Chazelle's algorithm relies heavily on duality transforms. We use the hierarchical representations of polyhedra due to Dobkin and Kirkpatrick to induce a cell complexes between coarse approximations, P^k and P_k of P satisfying $P_k \subseteq P \subseteq P^k$, and similar approximations Q^k and Q_k of Q satisfying $Q_k \subseteq Q \subseteq Q^k$. We show that the structure of such complexes allows intersection queries to be answered efficiently. In particular, the sequence of cells intersected by a ray can be identified in time proportional to the length of the sequence.

The algorithm operates by recursively computing the intersections: $P^k \cap Q_k$ and $P_k \cap Q^k$. Then edges of the union of approximations $P \cap Q^k$ and $Q \cap P^k$ are traversed by tracing their intersection with the two cell complexes. We show that each such edge can be traversed in constant time. In the process, most of the edges of $P \cap Q$ which lie simultaneously on the boundary of P and Q will be traced. We show that the total time needed to construct those which remain is linear in the size of P and Q.

Though based on the same general principles, the algorithm presented here is somewhat simpler than that described by Chazelle, which uses only the cell complexes induced by the inner hierarchical representations of P and Q. By extending Chazelle's search structure to the space exterior to the given polyhedra, we avoid having to operate simultaneously in primal and dual spaces. This permits us to conceptualise the algorithm as traversing the edges of the boundary of $(P \cap Q^k) \cup (Q \cap P^k)$. As a side effect, we avoid one half of Chazelle's recursive calls, which leads to a modest improvement in the asymptotic constants.

Contents

List of Figures

Chapter 1

Introduction

The main contribution of this thesis, is the development in full of a simple linear-time algorithm for computing the intersection of two three dimensional convex polytopes. In the process, cell complexes are described which abstract some of the properties of Dobkin and Kirkpatrick's hierarchical representations. The properties of such cell complexes are described for arbitrary dimensions, although their use is restricted to three dimensions in this thesis.

As shown by Muller and Preparata[7], the problem of computing the intersection of two convex polytopes can be linearly reduced to that of computing a description of the intersection of their boundaries. Intuitively, such a description provides a pattern for 'sewing' together facets of the two polyhedra to form their intersection. Brute force approaches to computing the boundary intersection which simply compute the intersection of each facet of one, with all facets of the other are, of course, doomed to take $\Theta(n^2)$ time, where n is any reasonable measure of the combined complexity of the two polytopes. Muller and Preparata describe the first non-trivial algorithm for computing the intersection of two convex polyhedra. Their algorithm, which runs in $O(n \log n)$ time, relies on the dual relationship between polytope intersection and convex union. Given two polytopes P and Q, the algorithm first identifies a point p in their intersection (or returns the empty set if none exists). Then, by applying an inversion transform[9] about p, P and Q are transformed into their duals, P^δ and Q^δ respectively. The convex hull Z of $P^\delta \cup Q^\delta$ is then computed, and finally the dual of Z is computed to yield a description of $P \cap Q$. Since both inversion and intersection detection are accomplished in linear time the algorithm is dominated by the $O(n \log n)$ cost of computing the convex union. Subsequently Hertel *et al.*[5] give a fundamentally different algorithm based on planar sweep to achieve the same upper bound. Yet another $O(n \log n)$ algorithm follows by direct application of the hierarchical representations of 3-polyhedra introduced in Dobkin and Kirkpatrick[3] (and revisited in Chapter 3 of this thesis).

While linear time solutions to the two-dimensional version of this problem (ie. intersecting convex polygons) have been known for some time[9], until Chazelle's recent result[2], linear time algorithms for intersecting 3-polyhedra have remained elusive. Chazelle's linear time algorithm uses both a truncated version of the hierarchical representations of Dobkin and Kirkpatrick, and the dual transforms used by Muller and Preparata. Even though the induced search structure which underlies Chazelle's algorithm has logarithmic depth, making a running time of $\Theta(n \log n)$ seem unavoidable, Chazelle uses an ingenious dovetailed recursion to avoid using the entire structure thereby reducing the time taken to $O(n)$.

The algorithm presented here was inspired by and bears considerable resemblance to Chazelle's algorithm. In particular Chazelle's observations that an inner hierarchical representation induces a cell complex suitable for ray tracing, and that a truncated version of this can be exploited in a recursive algorithm are fundamental to our algorithm as well. However, the two algorithms differ in several important respects. By extending Chazelle's search structure to the space exterior to the given polyhedra, we manage with only one half of Chazelle's recursive calls which leads to a simplification in the analysis of the algorithm and to a modest improvement in the asymptotic constants.

We consider both the inner and the outer hierarchical representations simultaneously and view their union as a cell complex partitioning both the interior and the exterior of the given polytopes. Truncating the approximation sequences at a fixed depth (k) gives a cell-complex which is bounded by two coarse approximations to the original and in which ray-tracing can be performed in constant $O(k)$ time. More specifically suppose P and Q are two convex polytopes with n and m edges respectively. To construct $P \cap Q$ we begin by building a search structure C_P on the space between two coarse approximations P_k and P^k satisfying $P_k \subseteq P \subseteq P^k$. A similar structure C_Q is built between approximations Q_k and Q^k of Q. Recursively, $Q^k \cap P_k$ and $P^k \cap Q_k$ are computed. The result of these recursive calls is then used to trace the edges of a carefully constructed approximation to $P \cup Q$ through C_P and C_Q taking $O(m+n)$ time in total.

Chapter 2 provides definitions for the terms and symbols used throughout the paper. Chapters 3 and 4 develop a cell decomposition of the space within and outside of each polytope. This structure will allow us to trace the surface of one polytope through the structure induced by the other, thereby exploiting the structure of both to reduce the number of boundary intersections which need to be considered. Chapter 5 shows how to use these structures to achieve an $O(n)$ algorithm. Finally Chapter 6 compares our algorithm with that described by Chazelle.

Chapter 2

Terms and Definitions

We begin by defining some terms and symbols which will be used throughout this paper. Many of the definitions, and in particular the definitions of (oriented) hyperplanes, convex polyhedra, convex polytopes, faces, vertices, edges and facets are exactly as described in[3].

If $a \in \Re^d - \{0\}$ and $c \in \Re$ then the set $H(a,c) = \{x \in \Re^d | \langle x, a \rangle = c\}$ (where $\langle x, a \rangle$ denotes the inner product of vectors x and a) is called a *hyperplane* in $\Re^d$. A hyperplane $H(a,c)$ defines two closed half spaces, $H^+(a,c) = \{x \in \Re^d | \langle x, a \rangle \geq c\}$ and $H^-(a,c) = \{x \in \Re^d | \langle x, a \rangle \leq c\}$. A *(convex) polyhedron* is defined as the intersection of a finite number of such closed half-spaces. Bounded polyhedra are known as *polytopes.* A polytope can be defined equivalently as the convex hull of a finite point set. If S is such a point-set set, we denote the convex hull of S by $\mathcal{CH}(S)$. If $\mathcal{H}$ is a finite set of hyperplanes, we define the associated polyhedra $\mathcal{P}(\mathcal{H}) = \cap_{H \in \mathcal{H}} H^+$.

We say that a hyperplane H *supports* a polytope P if $P \subseteq H^+$ and $H \cap P \neq \emptyset$. We say a hyperplane H supports P *at* x when $P \subseteq H^+$ and $x \in P \cap H$. If H supports P, then $H \cap P$ is called a *face* of P. Zero dimensional faces of are called *vertices*, one dimensional faces are called *edges* and $d-1$-dimensional faces are called *facets.* The set of all vertices (resp. edges and facets) of a polytope P is denoted by $V(P)$ (resp. $E(P)$ and $F(P)$). The union of all the facets of P is called the *boundary* of P and is denoted by ∂P. $P - \partial P$ is called the *interior* of P and denoted $\mathrm{Int}(P)$. Observe that the set $V(P)$ is the smallest set of points S such that $P = \mathcal{CH}(S)$. Similarly, we use the notation $\mathcal{D}(P)$ to denote the smallest set of hyperplanes, $\mathcal{H}$ satisfying $P = \mathcal{P}(\mathcal{H})$. We call such a hyperplane a *defining* hyperplane, and observe that its intersection with P is a facet.

We say two vertices, v_1 and v_2 are *neighbours* in P iff there is an edge, $e \in E(P)$ such that $e = \mathcal{CH}(\{v_1, v_2\})$. We say that they are *independent* if they are not neighbours. The neighbourhood $\mathcal{N}(P; v)$ of a vertex, v is the set of all such neighbours. We say two defining hyperplanes h_1 and h_2 are *neighbours* in P iff their common intersection with P is a $d-2$ dimensional face. Defining hyperplanes are *independent* iff they are not neighbours. The neighbourhood $\mathcal{N}(P;h)$ of a defining hyperplane h, is the set of all such neighbours. As polytopes are closed objects, the compliment of P, denoted $\overline{P}$, specifically excludes the boundary of P. On occasion, it will be convenient to have a way to refer to the set of points on or beyond the boundary of P. Thus we denote $\overline{P} \cup \partial P$ by $\widetilde{P}$. Since $P \cup Q = (P \cap \overline{Q}) \cup (\overline{P} \cap Q) \cup (P \cap Q)$, the notions of faces, vertices, edges and facets can be extended in a natural way to the union of convex polytopes. In particular, $\partial(P \cup Q) = (\partial P \cap \widetilde{Q}) \cup (\partial Q \cap \widetilde{P})$.

Chapter 3

Polytope Sequences Viewed as Cell Complexes

Let $\mathcal{S} = P_1, \ldots, P_\alpha$ be an arbitrary sequence of polyhedra in $\Re^d$ such that, for $1 \leq i < \alpha$, $P_{i+1} \subseteq P_i$. The connected components of $P_i - P_{i+1}$ for each i, together with $\overline{P_1}$ and P_α, will be referred to as *cells*. [1] A cell complex $\mathcal{C}(\mathcal{S})$ arising in this way will be referred to as a *polytopal cell complex*. Such a complex is illustrated in Figure 3.1. We associate with each cell c a level $\mathcal{L}(\mathcal{C}(\mathcal{S}); c)$, such that $\mathcal{L}(\mathcal{C}(\mathcal{S}); c) = i$ iff $c \subseteq P_i - P_{i+1}$, $\mathcal{L}(\mathcal{C}(\mathcal{S}); \overline{P_1}) = 0$ and $\mathcal{L}(\mathcal{C}(\mathcal{S}); P_\alpha) = \alpha$. In cases where the polytope sequence $\mathcal{S}$ is clear from the context, we will omit the first argument, abbreviating the notation to $\mathcal{L}(c)$. Cells c such that $\mathcal{L}(c) > 0$ and $\mathcal{L}(c) < \alpha$ will be called *proper cells* to distinguish them from $\overline{P_1}$ and P_α. The concept of levels can be extended naturally to arbitrary points $x \in \Re^d$ by defining $\mathcal{L}(\mathcal{C}(\mathcal{S}); x) = \mathcal{L}(\mathcal{C}(\mathcal{S}); c)$ where c is the unique cell which contains the point x.

Lemma 3.1 *If an arbitrary line penetrates a sequence of cells $c_1, \ldots, c_n$ of a polytopal cell complex then the sequence of levels, $\mathcal{L}(c_1), \ldots \mathcal{L}(c_n)$ is unimodal.*

Proof: The polytopes which form the complex are convex, and hence each can be intersected by a line at most once. $\square$

We define $|c|$, the *complexity* of a level i cell c as the total number of faces of P_i and P_{i+1} which intersect its boundary. We say that c is *simple* iff its complexity is bounded by a predetermined constant ρ. We say that a polytopal cell complex is *simple* if all of its proper cells are simple. If the innermost (resp. outermost) polytope P_α of a simple polytopal cell complex is also simple, then we

[1] This is somewhat of an abuse of the term used in various branches of topology, since the boundaries of our cells are partially open.

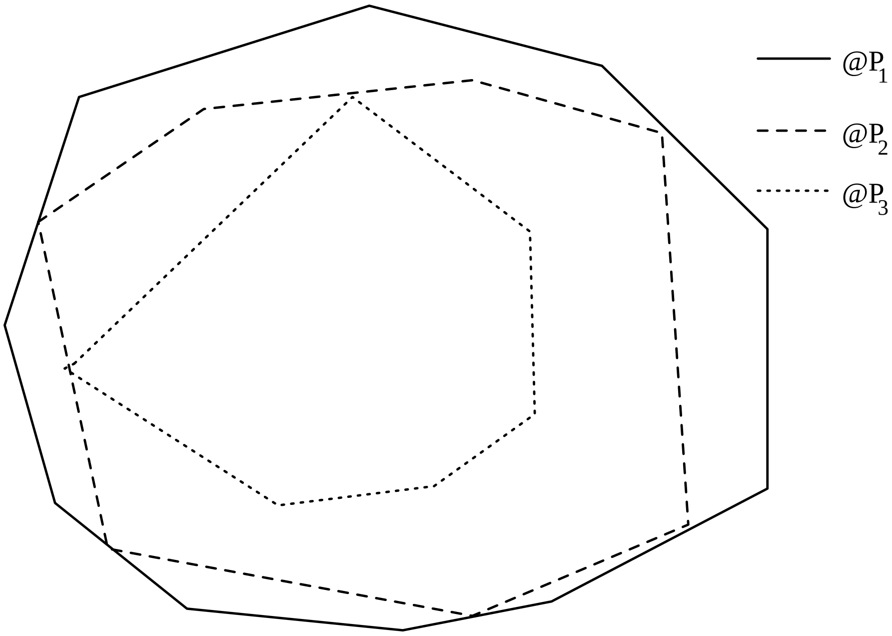

Figure 3.1: A polytopal Cell Complex

say that the complex is i-*simple* (resp. o-*simple*). A cell complex which is both *i*-simple and *o*-simple will be said to be t-*simple*. Throughout the remainder of this chapter, we assume that $\mathcal{C}(\mathcal{S})$ is an arbitrary simple polytopal cell complex. Moreover, throughout the rest of this thesis, we restrict the generality of the term "cell complex" to simple polytopal cell complexes just defined.

Lemma 3.2 *Suppose that H is a defining hyperplane of some $P_j \in \mathcal{S}$ and i is the smallest integer such that $H \in \mathcal{D}(P_i)$. Then there is a unique cell $C(H)$ such that $P_{i-1} \cap \overline{H^+} \subset C(H)$.*

Proof: Let x_1 and x_2 be two points in $P_{i-1} \cap \overline{H^+}$. We show that they are in the same cell. Clearly, $\overline{x_1 x_2} \cap P_i = \emptyset$ since both x_1 and x_2 lie in $\overline{H^+}$. On the other hand, $\overline{x_1 x_2} \subseteq P_{i-1}$ since both x_1 and x_2 are in P_{i-1}. Since the line connecting them neither enters P_i nor leaves P_{i-1} we must conclude that they are in the same cell. □

Lemma 3.3 *If $\mathcal{C}(P_1, \ldots, P_\alpha)$ is an arbitrary simple (resp.* i-*simple,*o-*simple,*t-*simple) cell complex, and H an arbitrary hyperplane, then the cell complex $\mathcal{C}(P_1 \cap H, \ldots, P_\alpha \cap H)$ is a simple (resp.* i-*simple,* o-*simple,* t-*simple) cell complex.*

Lemma 3.4 *If $\mathcal{C}(P_1, \ldots, P_\alpha)$ is an arbitrary simple cell complex and H is a defining hyperplane of some P_i where $i > 1$, but is not a defining hyperplane of P_1, then the cell complex $\mathcal{C}(P_i \cap H, \ldots, P_\alpha \cap H)$ is a* o*-simple cell complex.*

Proof: The facet $P_i \cap H$ must be simple since it bounds a simple cell. □

3.1 Representation

Before proceeding to make claims about the cost of computing certain things with a polytopal cell complex, we must make explicit some assumptions about representation. We assume that the full facial graph is available for each polytope in the sequence, with each face being represented explicitly. For polytopes of three or less dimensions, this graph will be planar and hence can be represented in linear space using the doubly connected edge list representation of Muller and Preparata[7]. In this representation, edges are oriented and represented explicitly by data structures containing the names of the two vertices which terminate them, the two facets which they separate, and pointers to the next edges encountered when proceeding clockwise around their terminating vertices.

We assume that cells are explicitly represented, or at least that a list of the faces of P_i and P_{i+1} which bound each level i cell can be constructed in constant (depending on ρ) time. We also assume that for each defining hyperplane H of each P_i, we have at hand, the unique cell $C(H)$ whose existence was proved in Lemma 3.2. Moreover for each face f of each P_i, we have a pointer to the highest dimensional face of P_{i+1} which it contains (if one exists). If no such face of P_{i+1} exists, then f's representation includes a pointer to the representation of the level i cell which it bounds.

Lemmas 3.3 discussed the cell complex which results from intersecting a cell complex C with a hyperplane H. While a representation of this complex is implicit in the representation of C, some cost may be incurred in extracting descriptions of its features. On the other hand a description of the cell complex arising in the more restricted situation discussed in Lemma 3.4 is *explicitly* present in the description of the original: Let f be a facet of some $P_k \in \{P_1, \ldots, P_\alpha\}$, let $f_i = P_i \cap f$ for all $k \leq i < \alpha$, and let j be the largest i such that $f_j \neq \emptyset$. Let $C^* = \mathcal{C}(f_k, \ldots, f_j)$. Observe that each f_i for $k \leq i \leq j$ is actually a face of P_i, and thus a description of it is directly available from the description of P_i. Each cell in C^* is merely a cell of C which has been intersected by a hyperplane. Thus, a description of each cell of C^* can be obtained in $O(\rho)$ time from the description of the corresponding cell in C. In fact, all the information which we claimed to have at hand for C, is available for C^* in $O(\rho)$ time from the description of C.

3.2 Point Location and Ray Tracing

Our primary interest in cell complexes, concerns the ease with which point-location and ray-tracing operations can be performed. We first define these operations. The problem of ray-tracing can be described as follows: Given a point x inside a known proper cell c and a direction vector $\vec{v}$, compute the next cell c' entered by a ray r which emanates from p in the direction $\vec{v}$. Since c has a constant sized description, it is a simple matter to compute the coordinates of the point x' at which the ray leaves c in $O(\rho)$ time. Thus, we lose no generality in assuming the the point x' on the boundary of c is also given. We say that r is emanating *outwards* if $\mathcal{L}(c') < \mathcal{L}(c)$. Of course, if r is emanating outwards, x' will actually be a point in the cell c. We say that r is emanating *inwards* if $\mathcal{L}(c') > \mathcal{L}(c)$. If r is emanating inwards, x' will be a point in the cell c' whose identity we are to determine. Unfortunately, in both cases knowing the face by which r leaves c does not immediately identify the new cell c' which it enters.

The problem of point-location is to identify the cell in a cell complex containing a point whose physical co-ordinates are known. For our purposes, it will suffice to consider this problem in the restricted case where cell complex is o-simple.

Lemma 3.5 *If an algorithm exists to solve the point-location problem for any query point x in an arbitrary $d-1$ dimensional* o*-simple cell complex C^* in $O(\mathcal{L}(C^*; x))$ time, then the identity of the next cell c' of an arbitrary d dimensional cell complex $C(\mathcal{S})$ entered by an arbitrary ray r which emanates inwards from a point p in a known proper cell c of $\mathcal{C}(\mathcal{S}))$ can be computed in $O(\mathcal{L}(\mathcal{C}(\mathcal{S}); c') - \mathcal{L}(\mathcal{C}(\mathcal{S}); c))$ time.*

Proof: Let $i = \mathcal{L}(\mathcal{C}(\mathcal{S}); c)$. Let f be the facet of P_i bounding c by which r leaves. Since c has an $O(\rho)$ sized description, we can identify f in $O(\rho)$ time. Let x' be the point at which r touches f, let c' be the cell which contains x', and let $j = \mathcal{L}(\mathcal{C}(\mathcal{S}); c')$. Let f_k represent $P_k \cap f$ for all $i < k \leq j$, and observe that each f_k is a face of P_k. The sequence $f_{i+1}, \ldots, f_j$ forms a $d-1$ dimensional o-simple cell complex, the structure of which — including cell names—is explicitly described by the description of the original. Thus, we can appeal to the point-location algorithm to locate w in the $d-1$ dimensional cell complex, $\mathcal{C}(f_{i+1}, \ldots, f_j)$ in $O(j-i)$ time. □

Lemma 3.6 *If point location for all points x in an arbitrary $d-1$ dimensional* o*-simple cell complex C^* can be accomplished in $O(\mathcal{L}(C^*; x))$ time, then it can be accomplished for all points y in an arbitrary d dimensional* o*-simple cell complex C in $O(\mathcal{L}(C; y))$ time.*

Proof: Suppose that x is the point we wish to locate in a d-dimensional cell complex $C = \mathcal{C}(\mathcal{S})$. Let o be any point in P_α. Let r be a ray from o through x.

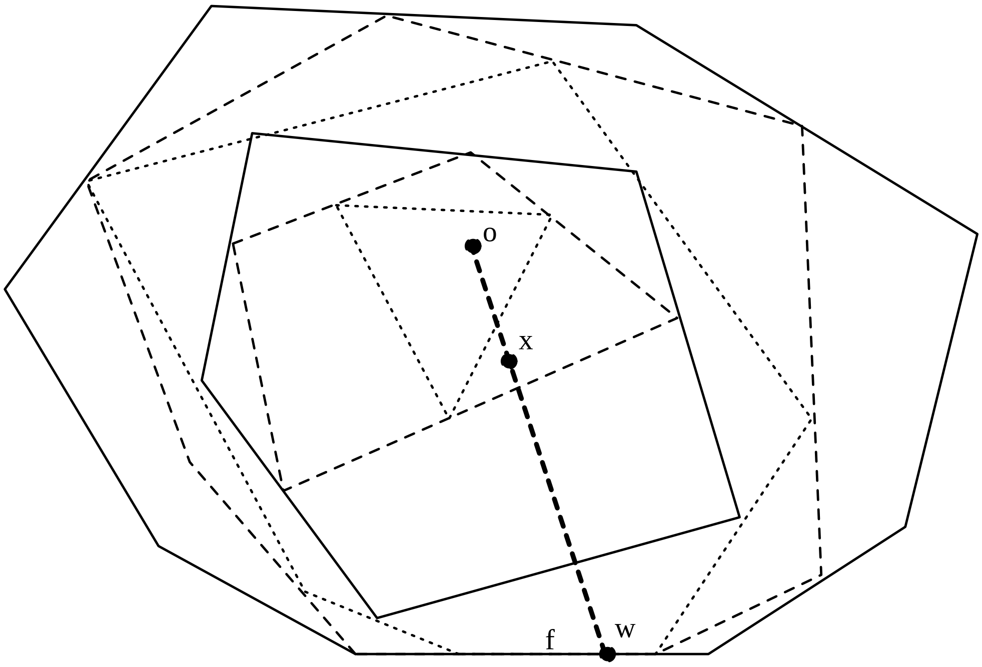

Figure 3.2: Locating a point x in a cell complex

Since P_1 has a simple description, we can identify the co-ordinates of the point w at which r intersects the boundary of P_1, and the facet f of P_1 which contains w in $O(\rho)$ time. This construction is illustrated in Figure 3.2. We now appeal to algorithm for computing point location in $d-1$ dimensions to compute the cell in the o-simple $\mathcal{C}(P_1 \cap f, \ldots, P_\alpha \cap f)$ which contains w. Having identified the cell which contains w, we appeal to Lemmas 3.1 and 3.5 to trace the ray from w to x in the required time. □

Theorem 3.7 *It is possible to locate any point x in an arbitrary d dimensional* o*-simple cell complex, C, in $O(\mathcal{L}(C; x))$ time.*

Proof: We prove the theorem by induction on the dimension of the problem. In one dimension, a simple cell complex is just a set of intervals, and point-location in the required time is trivial. Suppose that the theorem is true for all problems of dimension less than d. Then Lemma 3.6 says that it is true for problems of dimension equal to d. □

Theorem 3.8 *The identity of the next cell c' of a cell complex C entered by a ray r which emanates inwards from a point p in a known proper cell c can be computed in $O(\mathcal{L}(C;c') - \mathcal{L}(C;c))$ time.*

Proof: Follows directly from Theorem 3.7 and Lemma 3.5. □

Having given upper bounds for the costs of doing point location in a o-simple cell complex, and tracing rays inwards in a simple cell complex, we now attend to tracing rays outwards. Algorithm 3.1 finds the next cell c' entered by

Algorithm 3.1

> **Repeat**
>
> > $c := C(H)$
> > Let D be the set of all hyperplanes G satisfying $G \in \mathcal{D}(P_{\mathcal{L}(c)})$, $G \cap c \neq \emptyset$, $r \subseteq G^-$, and $r \cap G = x$.
> > **If** $D \neq \emptyset$ **then**
> >
> > > let H be any element of D.
> >
> > **Endif**
>
> **Until** $D = \emptyset$

a ray r which emanates outwards from a cell c. Its operation is illustrated in Figure 3.3. Lemma 3.9 shows it to be efficient, and Lemma 3.10 shows it to be correct. Initially, we let x be the point on ∂c by which r leaves. We initialise H to be any defining hyperplane of $P_{\mathcal{L}(C)}$ which bounds c and which r crosses by leaving c. More formally, we require: $H \in \mathcal{D}(P_{\mathcal{L}(c)})$,$r \subseteq H^-$ and $r \cap H = x$. Such a hyperplane is guaranteed to exist, since r must cross some bounding hyperplane. After the algorithm terminates, c will identify the next cell entered by r.

Lemma 3.9 *Suppose c_i is the initial value of c. Then the algorithm terminates in $O(\mathcal{L}(c_i) - \mathcal{L}(c_r))$ time, where c_r is the final value of c.*

Proof: The algorithm terminates since $\mathcal{L}(c)$ is monotonically decreasing, and $\mathcal{D}(P_1) = \emptyset$. It terminates in the required time, since each step in the iteration requires only constant time. □

Lemma 3.10 *When Algorithm 3.1 terminates, c will be the next cell entered by r.*

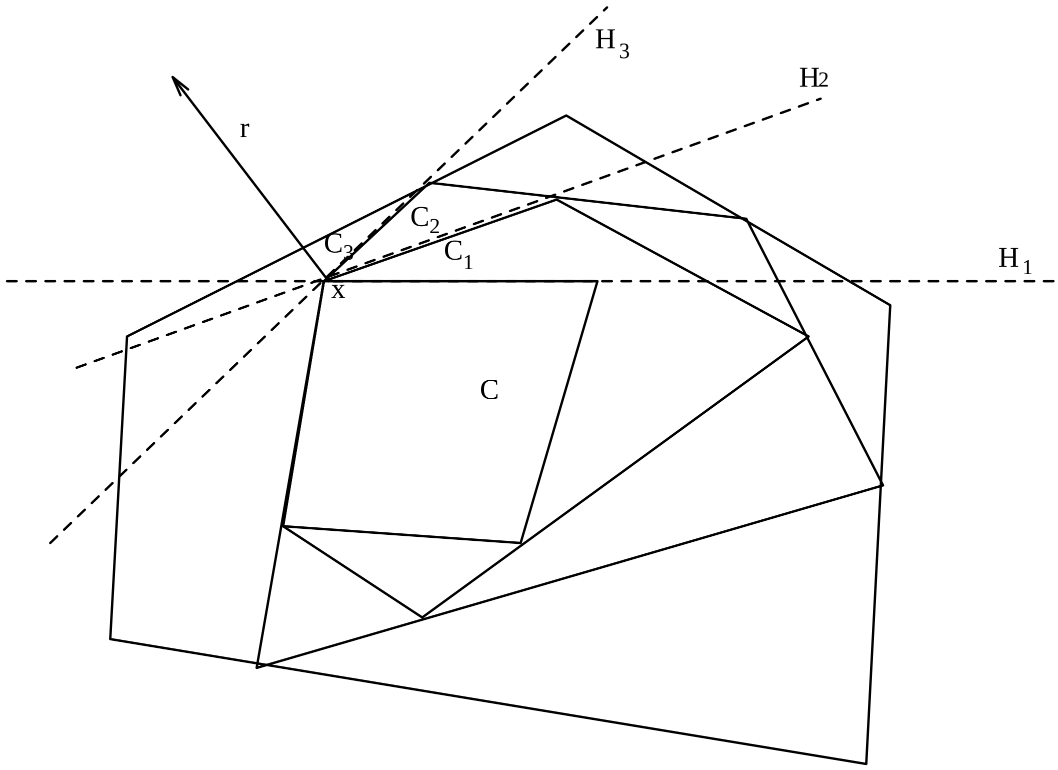

Figure 3.3: Tracing a ray outwards

Proof: We first show that the algorithm does not 'miss' the appropriate cell. Suppose that D is not empty, and hence the algorithm continues for at least one more iteration. Let G be any element of D and observe that $r - \{x\} \subseteq \overline{G+}$, but $c \subseteq G^+$. Thus, c cannot be the next cell entered by r, and the algorithm correctly continues to iterate.

Suppose on the other hand that D is empty. We must show that that c is the next cell entered by r. Let $x = r \cap H$. Let $\vec{v}$ be a unit vector in the direction of r, so that points along r can be expressed as $x + t\vec{v}$ for non-negative scalars t. We say that a point x *violates* a defining hyperplane H of some polytope $P \subseteq H^+$ if $x \in \overline{H^+}$. We say that a ray r immediately *violates* such a hyperplane H if for all positive (non-zero) scalars s we have $x + s\vec{v} \in \overline{H^+}$.

We know that $r \subset H^-$ for some $H \in \mathcal{D}(P_{\mathcal{L}(c)+1})$, and that c is the unique cell $C(H)$ such that $P_{\mathcal{L}(c)} \cap \overline{H^+} \subseteq c$. Now, if a point is not in c, then it clearly must violate a defining hyperplane of $P_{\mathcal{L}(c)}$ which bounds c, or it must be inside $P_{\mathcal{L}(c)+1}$. Since D is empty, we know that the ray does not immediately violate a defining hyperplane of $P_{\mathcal{L}(c)}$ which bounds c. Since r immediately crosses H entering H^-, we must conclude that r immediately enters c. □

Theorem 3.11 *Suppose that $C = \mathcal{C}(P_1, \ldots, P_\alpha)$ is an arbitrary simple cell complex, and suppose that x is a point in $P_1 \cap \overline{P_\alpha}$, or in an edge of P_α. Given the proper cell c of C or the edges of P_α containing x, the next cell c' of C entered by a ray r emanating from x can be computed in $O(|\mathcal{L}(C;c') - \mathcal{L}(C;x)|)$ time.*

Proof: The result is a direct consequence of Theorem 3.8 and Lemmas 3.9 and 3.10. □

This result is the fundamental result of this section, providing a primitive operation used throughout the algorithm for computing $P \cap Q$. If $C = \mathcal{C}(P_1, \ldots, P_\alpha)$, we say that the region

$$C_N = P_1 - \overline{P_\alpha} \bigcup_{e \in E(P_\alpha)} e$$

is the *navigable* region of C. Observe that if l is a line segment, and x is a point in a connected component l' of $l \cap C_N$ then the sequence of cells $c_1, \ldots, c_k$ intersected by l' can be computed in $O(\max\{\mathcal{L}(C;c_i)\} - \min\{\mathcal{L}(C;c_i)\})$ time by repeated application of Theorem 3.11.

3.3 Steepest Descent

We conclude this chapter by describing a procedure that we call "steepest descent," which enables us to find out whether a particular polytope in the sequence intersects a given hyperplane.

Theorem 3.12 *Suppose that $C = \mathcal{C}(\mathcal{S})$ is a simple cell complex and that H is a hyperplane intersecting P_j. If the cell c containing a point $x \in P_j \cap H$ has been identified, then one can decide whether a point in $P_k \cap H$ exists in $O(k - \mathcal{L}(c))$ time, returning such a point if it does.*

Proof: Let $i = \mathcal{L}(c)$. If $i = k$ then we are done. Suppose that $i < k$. We first intersect c with H in constant time and consider $H \cap c$. If H does not intersect a portion of the boundary which belongs to ∂P_{i+1}, then $H \cap c = H \cap P_i$, and thus H cannot intersect P_k. On the other hand, if H does intersect a portion of the boundary which belongs to ∂P_{i+1} then we let x' be any point on the intersection, compute $c' = C(x')$ in $O(\mathcal{L}(c) - i)$ time, and apply the algorithm iteratively to c'.

Now suppose that the algorithm iterates n times identifying a sequence cells $c_1, \ldots, c_n$. Let $c_0 = c$ be the cell given initially. The total time spent is thus given by $\sum_{x=1}^{n} O(\mathcal{L}(c_x) - \mathcal{L}(c_{x-1}))$, which reduces to $O(\mathcal{L}(c_n) - \mathcal{L}(c))$ as required. $\square$

Chapter 4

Hierarchical Representations

We say that a sequence of polytopes $P_1, \ldots, P_k$ in $\Re^d$ is an *inner hierarchical representation* of a polytope P iff

1. $P_1 = P$
2. P_k is a d-simplex
3. $V(P_{i+1}) \subseteq V(P_i)$
4. no two vertices in $V(P_i) - V(P_{i+1})$ are neighbours in P_i.

Similarly a sequence of polytopes, $P_1, \ldots, P_k$ n $\Re^d$ is an *outer hierarchical representation* of a polytope P iff

1. $P_1 = P$
2. P_k is a d-simplex
3. $\mathcal{D}(P_{i+1}) \subseteq \mathcal{D}(P_i)$
4. no two facets in $F(P_i) - F(P_{i+1})$ are neighbours in P_i.

These constructions, which are illustrated in Figures 4.1 and 4.2 are precisely the hierarchical representations of Dobkin and Kirkpatrick[3]. In this chapter, we explore the difference between successive elements of such descriptions, with an eye towards identifying situations under which the resulting cell complexes are simple. We define the *cell* associated with a vertex v of a polytope P as $C(P; v) = P - \mathcal{CH}(V(P) - v)$, and show that the cells associated with independent vertices are disjoint. Moreover, we show that if P_i and P_{i+1} are successive

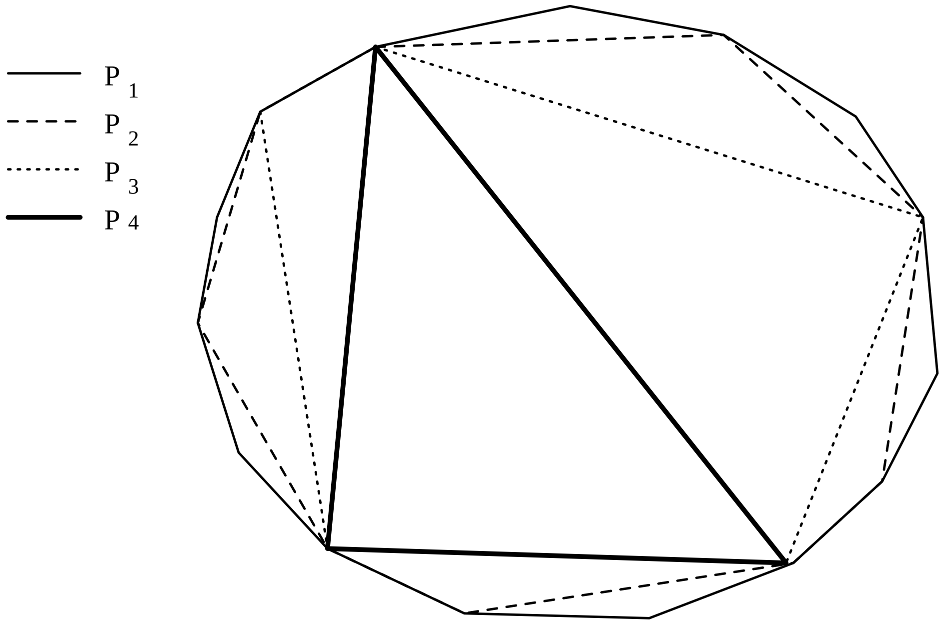

Figure 4.1: An inner hierarchical description of a polytope P

elements in an inner hierarchical description, and $S = V(P_i) - V(P_{i+1})$, then then $P_i - P_{i+1} = \bigcup_{v \in S} C(P_i; v)$.

Similar results hold for outer hierarchical descriptions. We define the *cell* associated with a defining hyperplane h of a polytope P as $C(P; h) = \mathcal{P}(\mathcal{D}(P) - h) - P$. We show that cells associated with independent defining hyperplanes are disjoint, and that if P_i and P_{i+1} are successive elements in an outer hierarchical description, and $S = \mathcal{D}(P_i) - \mathcal{D}(P_{i+1})$ then $P_{i+1} - P_i = \bigcup_{h \in S} C(P_i; h)$.

We also show that the complexity of a cell associated with a vertex or defining hyperplane is functionally dependent (linearly when $d \leq 3$) on the size of that vertex's or hyperplane's neighbourhood. Thus, to form a simple cell complex, all that is required is that the neighbourhood of a vertex or defining hyperplane which appears in P_i and is absent from P_{i+1} be bounded by an appropriate constant.

4.1 Inner Hierarchical Descriptions

This section examines the nature of cells in an inner hierarchical description. Towards this end, we let P be an arbitrary convex polytope and let v be one of its vertices. We begin by describing the cell, $C(P; v)$. Let $P' = \mathcal{CH}(V(P) - \{v\})$; $N = \mathcal{CH}(\mathcal{N}(P; v) \cup \{v\})$ and $N' = \mathcal{CH}(\mathcal{N}(P; v))$.

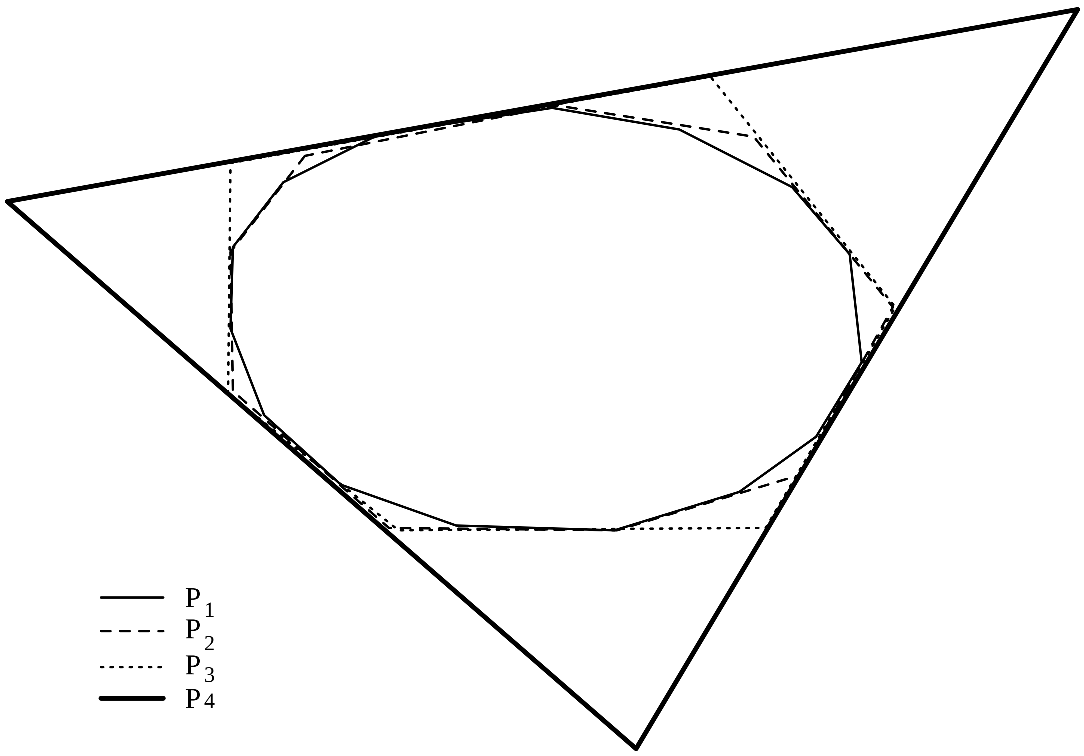

Figure 4.2: An outer hierarchical description of a polytope P

Lemma 4.1 *Let P be a polytope and v one of its vertices. If H is a hyperplane such that $v \in H$ and $\mathcal{N}(P;v) \subseteq H^+$ then $P \subseteq H^+$*

Proof: Suppose $H = H(a,c)$ and let $G = G(a', c')$ be any hyperplane through v which supports P and minimises the angle θ between the normal vectors a and a' . If $\theta = 0$, then $G = H$ and H supports P as required.

Suppose that $|\theta| > 0$. Since G minimises θ, there must be another vertex v' of P which lies on $G \cap \overline{H^+}$. Moreover, the closest such vertex to v must be a neighbour of v, contradicting the assumption that $\mathcal{N}(P;v) \subseteq H^+$. □

Lemma 4.2 *Suppose P is a polytope and v is one of its vertices. If H is a hyperplane such that $v \in H^-$ and $\mathcal{N}(P;v)) \subseteq H^+$ then $V(P) - \{v\} \subseteq H^+$*

Proof: Suppose that $H = H(h,c)$. If $v \in H$ then Lemma 4.1 applies directly. If $v \notin H$, then translate H to $H' = H(h,c')$ so that $v \in H'$. By Lemma 4.1 we know that $V(P) \subseteq H'^+$. Let W be the set of vertices of P which lie in $\overline{H^+} \cap H'^+$. We shall show that $W = \{v\}$. Clearly, $v \in W$. Suppose with an eye towards contradiction that w is another vertex in W. Let θ be the maximum angle formed by the ray $\vec{vw}$ and the ray $v + zh$ ($z >= 0$) for any $w \in W$, and consider the cone defined by all points x such that the angle between $\vec{xv}$ and $v + zh$ equals θ. Observe that any hyperplane containing x and tangent to the cone supports P, and that any vertex $w \in W$ is a neighbour of v by virtue of such a hyperplane containing w and v, and lying tangent to the cone. Of course, no such point w can be a neighbour of v since $W \subseteq \overline{H^+}$ and neighbours of P are constrained to lie in H^+. Thus, no vertices of P can lie in $\overline{H^+}$ as required. □

Lemma 4.3 $P = P' \cup N$

Proof: It should be clear that both P' and N are subsets of P, and hence so is their union. Suppose that x is a point in P, but in neither P' nor N. There exists a hyperplane H_1 such that $x \notin H_1^+$, but $P' \subseteq H_1^+$. Similarly there exists a hyperplane, H_2 such that $x \notin H_2^+$ but $N \subseteq H_2^+$. Notice that $v \notin H_1^+$ since if it were, H_1 would separate P from x. Thus, $v \in \overline{H_1^+} \cap H_2^+$. Consider the hyperplane G through v and $H_1 \cap H_2$. Observe that it separates x from v and $\mathcal{N}(P;v)$, and by Lemma 4.1 from P, contradicting the assertion that $x \in P$. Thus, no such point x exists. □

Theorem 4.4 $C(P;v) = \mathcal{CH}(\mathcal{N}(P;v) \cup \{v\}) - \mathcal{CH}(\mathcal{N}(P;v))$

Proof: From Lemma 4.3, we know that $P - P' \subseteq N$. From their definitions we know that $N' \subseteq P'$. Thus we conclude that $P - P' \subseteq N - N'$. On the other hand let x be a point in $N - N'$. Since $x \notin N'$ there exists a hyperplane H such that $x \notin H^+$, and $N' \subseteq H^+$. Since no hyperplane can separate x from $V(N)$, we must have $v \notin H^+$. We have thus established a situation, in which $v \in H^-$, and $\mathcal{N}(P;v) \subseteq H^+$. From Lemma 4.2 we conclude that $V(P) - \{v\} \subseteq H^+$ and hence $P' \subseteq H^+$. Since $x \notin H^+$, we must also conclude that $x \notin P'$ proving that $N - N' \subseteq \overline{P'}$. Since $N \subseteq P$ it follows that $N - N' \subseteq P - P'$. □

Corollary 4.5 *In $\Re^3$, if $|\mathcal{N}(P;v)| \leq \kappa$ then $|C(P;v)| \leq 6(\kappa + 1)$*

Proof: The boundary of the resulting cell could be represented as a planar graph. Thus, the total number of facets cannot exceed six time the number of vertices. □

Unlike cells in an arbitrary cell complex, those that arise from inner hierarchical representations are star shaped.

Lemma 4.6 *If $x \in C(P;v)$ then $\overline{vx} \subseteq C(P;v)$*

Proof: Let x be any point in $C(P;v)$. Since $x \notin P'$, there must exist a hyperplane H such that $P' \subseteq H^+$, and $x \notin H^+$. Since $x \in P$ we must have $v \notin H^+$, otherwise H would separate x from $V(P)$. Since P is convex, and both x and v are in P, we can be sure that $\overline{vx} \subseteq P$. On the other hand, since both v and x are not in H^+, and $P' \subseteq H^+$, we can be equally sure that $\overline{vx} \cap P' = \emptyset$. Thus, we can conclude that $\overline{vx} \subseteq C(P;v)$. □

Up until now, we have examined the effect of removing a single vertex v from a convex polytope, and forming the convex hull of those which remain. We have shown that the resulting cell is star-shaped with all points visible from v, and that its complexity depends only on the size of v's neighbourhood. We now show that if two vertices are not neighbours, then the cells resulting from their removal are disjoint. Moreover the order in which they are removed in no way affects the resulting cells.

Lemma 4.7 *If x is a vertex of P such that $x \notin \mathcal{N}(P;v)$ and H is a hyperplane supporting P' at x then H supports P.*

Proof: Suppose that H does not support P. Then there must be a vertex of P which is not in H^+. The only vertex of P which is not also a vertex of P' is v, so $v \notin H^+$. However, $v \notin \mathcal{N}(P;x)$, and thus we have $x \in H$ and $\mathcal{N}(P;x) \subseteq H^+$ and hence $P \subset H^+$ by Lemma 4.1 □

Throughout the remainder of this section, we let P be an arbitrary convex polytope, and v_1 and v_2 be two distinct vertices of P. We let $P_1 = \mathcal{CH}(V(P) - \{v_1\}$ and $P_2 = \mathcal{CH}(V(P) - \{v_2\})$.

Theorem 4.8 *Disjointness: If $v_1 \notin \mathcal{N}(P; v_2)$ then $C(P; v_1) \cap C(P; v_2) = \emptyset$*

Proof: Suppose, with an eye towards contradiction, x is a point in the intersection of the two cells. Consider the ray r_1 starting at v_1 and proceeding through x. Such a ray must penetrate a face f_1 of P_1, which is contained in a hyperplane H_1 such that $V(P_1) \subseteq H_1^+$. Similarly the ray r_2 starting at v_2 and proceeding through x must penetrate a face f_2 of P_2 which is contained in a hyperplane H_2 such that $V(P_2) \subseteq H_2^+$.

Let z be the point at which r_2 penetrates f_2, and notice that $v_2 \in H_1^+$, $x \notin H_1^+$, and hence $z \notin H_1^+$. Since z lies on f_2, there must be a vertex of f_2 which is not in H_1^+. Such a vertex must also be a vertex of P_2, and the only possible such vertex is v_1.

So here we have a situation in which H_2 does not support P since $v_2 \notin H_2^+$, H_2 supports P_2 at v_1, and $v_1 \in V(P)$. From Lemma 4.7 we can conclude that $v_1 \in \mathcal{N}(P; v_2)$, contradicting the original premise. □

Lemma 4.9 *If $v_1 \notin \mathcal{N}(P; v_2)$ then $\mathcal{N}(P; v_2) = \mathcal{N}(P_1; v_2)$.*

Proof: Suppose $x \in \mathcal{N}(P; v_2)$. Then there is a hyperplane H which supports P such that $H \cap P = \overline{xv_2}$. Now $P_1 \subseteq P$, so $H \cap P_1 \subseteq H \cap P$. The vertex $v_1 \notin \mathcal{N}(P; v_2)$ so $x \neq v_1$, $\overline{xv_2} \subset P_1$, and $H \cap P_1 = \overline{xv_2}$.

On the other hand, suppose that $x \in \mathcal{N}(P_1; v_2)$. Then there is a hyperplane H supporting P_1 such that $H \cap P_1 = \overline{xv_2}$. From Lemma 4.7 we know that H supports P. If $v_1 \in H$ then $H \cap V(P) = \{v_1, v_2, x\}$, and there must be an edges between v_1 and v_2. If $v_1 \notin H$, then $H \cap P$ must remain unchanged. □

Theorem 4.10 *If $v_1 \notin \mathcal{N}(P; v_2)$ then $C(P_1; v_2) = C(P; v_2)$*

Proof: Follows as a direct consequence of Theorem 4.4 and Lemma 4.9 □

Corollary 4.11 *Suppose that R is an independent subset of $V(P)$ and $P' = \mathcal{CH}(V(P) - R)$. Then $P = P' \cup (\cup_{v \in R} C(P; v))$.*

Consider now the inner hierarchical representation, $P_1, \ldots, P_k$ of a polytope P. Let $S_i = V(P_i) - V(P_{i+1})$ for $1 \leq i < k - 1$. Moreover, suppose that the polytopes $P_2, \ldots, P_k$ are constructed so that if $v \in S_i$, then $|\mathcal{N}(P_i; v)| \leq \kappa$ for some predetermined constant κ. From Theorem 4.8 and Corollary 4.11 we conclude that the cells $C(P_i; v)$ for each $v \in S_i$ are the connected components of $P_i - P_{i+1}$. In $\Re^3$, if we choose $\rho > 6\kappa$ then $\mathcal{C}(P_1, \ldots, P_k)$ forms a simple polytopal cell complex. Moreover, Corollary 4.5 assures us that a description of the cell associated with each vertex $v \in S_i$ is available in $O(\kappa \log \kappa)$ time from the description of P_i. Thus the cell complex is adequately represented by DCEL representations (described on page 9) of each polytope in the sequence.

4.2 Outer Hierarchical Descriptions

In the previous section, we described the cell complex induced by an inner hierarchical description of a polytope. Here we develop a complimentary description of the cell complex induced by an outer hierarchical description. Recall that we define a polytope as the intersection of a finite number of closed half-spaces in R^d. If H is a set of hyperplanes in R^d, we define the polyhedra $\mathcal{P}(H) = \cap_{h \in H} h^+$. Of course there is a certain duality between the construction of inner and outer hierarchical representations. In particular, taking the geometric dual of the polytopes $P_1, \ldots, P_k$ in an inner hierarchical representation of P about a point in the interior of P_k yields an outer hierarchical representation $P_1^\delta, \ldots, P_k^\delta$ of P^δ, the dual of P about the same point. Thus it should not be surprising that the results of Section 4.1 have analogues for the cell complexes arising from outer hierarchical representations. However, one of the goals of this thesis is to avoid the use of dual transforms hence we develop separately the results for outer hierarchical representations. When h is a hyperplane, we define the associated cell as $C(P; h) = \mathcal{P}(\mathcal{D}(P) - \{h\}) - P$. Throughout the following section, we assume that P is an arbitrary convex polytope, $h \in \mathcal{D}(P)$, and that $P' = \mathcal{P}(\mathcal{D}(P) - h)$.

Lemma 4.12 *Let g be a hyperplane in $\mathcal{D}(P)$. If $g \notin \mathcal{N}(P; h)$ then $g \cap P = g \cap P'$ hence $g \cap C(P; h) = \emptyset$*

Proof: Observe that $g \cap P$ is a convex polytope in $d-1$ dimensions. Since h and g are not neighbours, we know that $h \cap g \cap P$ is not a $d-2$ face of P. Thus, $h \cap P \notin \mathcal{D}(g \cap P)$ and hence $g \cap P' = g \cap P$. □

Theorem 4.13 $C(P; h) = \bigcap_{x \in \mathcal{N}(P;h)} x^+ \cap \overline{h^+}$

Proof: Let $W = \bigcap_{x \in \mathcal{N}(P;h)} x^+ \cap \overline{h^+}$. It follows directly from their definitions that $C(P; h) \subseteq W$. It remains to show that $W \subseteq C(P; h)$. Let x be a point in w. Since $x \notin h^+$, we know that $x \notin P$. Suppose with an eye towards contradiction that $x \notin P'$. There must be some $g \in \mathcal{D}(P')$ such that $x \notin g^+$. Since $C(P; h) \subseteq W$, there must be some point in $C(P; h)$ which is in g^+, thus g^+ must partition $C(P; h)$. But, according to Lemma 4.12, g cannot intersect $C(P; h)$, thus establishing a contradiction and proving that $x \in P'$. Since $x \notin P$ and $x \in P'$, we have also proven that $x \in C(P; h)$ and hence $W \subseteq C(P; h)$. □

Corollary 4.14 *In $\Re^3$, if $|\mathcal{N}(P; v)| \leq \kappa$ then $|C(P; v)| \leq 6(\kappa + 1)$*

Proof: The boundary of the resulting cell could be represented as a planar graph. Thus, the total number of faces cannot exceed six times the number of facets. □

Throughout the remainder of this section, we assume that P is an arbitrary convex polytope, that h_1 and h_2 are two distinct defining hyperplanes. We let $P_1 = \mathcal{P}(\mathcal{D}(P) - \{h_1\})$ and $P_2 = \mathcal{P}(\mathcal{D}(P) - \{h_2\})$.

Theorem 4.15 $C(P; h_1) \cap C(P; h_2) = \emptyset$

Proof: Let x be a point in $C(P; h_1)$. Theorem 4.13 says that $x \in \overline{h_1^+}$. On the other hand, $C(P; h_2) \subseteq P_2 \subseteq h_1^+$, proving that the two cells are disjoint. □

Theorem 4.16 *If* $h_1 \notin \mathcal{N}(P; h_2)$ *then* $C(P_1; h_2) = C(P; h_2)$

Proof: To prove the theorem we need only show that $\mathcal{N}(P_1; h_2) = \mathcal{N}(P; h_2)$. For a hyperplane to be in such a neighbourhood, it must be a defining hyperplane. Since the defining hyperplanes if P_1 and P differ only by the inclusion or omission of h_1 which is known not to be in the neighbourhood of h_2, the possible candidates are the same. From Lemma 4.12 we know that $h_2 \cap P_1 = h_2 \cap P$, so for any hyperplane x, it must follow that $x \cap h_2 \cap P_1 = x \cap h_2 \cap P$. □

Corollary 4.17 *Suppose that* R *is an independent subset of* $\mathcal{D}(P)$ *and* $P' - \mathcal{P}(\mathcal{D}(P) - R)$. *Then* $P = P' \cup (\bigcup_{h \in R} C(P; h)$.

As with inner hierarchical descriptions, we now consider the outer hierarchical representation $P_1, \ldots, P_k$ of a polytope P, constructed so that the defining hyperplanes removed from successive polytopes have no more than κ neighbours for some predetermined constant κ. From Theorem 4.15 and Corollary 4.17 we conclude that the cells $C((; P)_i, h)$ where $h \in \mathcal{D}(P_i) - \mathcal{D}(P_{i+1})$ are the connected components of $P_i - P_{i+1}$. In $\Re^3$, if we choose $\rho \geq 6\kappa$ then the resulting cell will be simple. Thus the sequence $P_k, P_{k-1}, \ldots, P_1$ forms a simple polytopal cell complex. Moreover, Theorem 4.15 assures us that a description of each cell $c \subseteq P_i - P_{i-1}$ can be obtained from the description of P_i in $O(\kappa \log \kappa)$ time.

Chazelle[2] was the first to observe that the inner hierarchical representation of a polytope P induces a spatial sub-division on its interior. In fact his polytope intersection algorithm relies heavily on the ability to trace rays efficiently in such a structure. Here we have extended this technique, by showing that both the inner and outer hierarchical representations result in such a structure. Suppose that $P_1, \ldots, P_x$ is an inner hierarchical representation of a polytope P, constructed so that $\mathcal{N}(P_i; v) \leq \kappa$ for all $v \in V(P_i) - V(P_{i+1})$ Suppose further that $P^1, \ldots, P^y$ is an outer hierarchical representation of the same polytope P, constructed so that $\mathcal{N}(P_i; h) \leq \kappa$ for all $h \in \mathcal{D}(P_i) - \mathcal{D}(P_{i+1})$. Let $\mathcal{S} = P^y, \ldots, P^2, P, P_2, \ldots, P_x$ and observe that $\mathcal{C}(\mathcal{S})$ is a t-simple polytopal cell complex, and thus inherits all the properties developed in Chapter 3. Kirkpatrick[6] provides a simple argument to show that such hierarchical descriptions can be constructed for arbitrary polyhedra with an appropriately chosen constant κ, in $O(n)$ space and time, and that such descriptions have depth at most $O(\log n)$ depth, where n is the number of edges in P.

As discussed in the introduction, hierarchical descriptions with logarithmic depth immediately give rise to an $O(n \log n)$ algorithm for computing 3-polyhedral intersection. To replace the $\log n$ term by a constant, we truncate the hierarchical descriptions, so that they contain at most k polytopes for some appropriately chosen constant k. Throughout the remainder of this thesis, we assume that such a polytopal cell complex has been constructed. We use the notation P_k and P^k to refer to the k^{th} polytopes of the inner and outer hierarchical representations of P respectively.

4.3 The Size of P^k and P_k

In this final section on hierarchical representations, we consider the question of how quickly the size of successive polytopes in a hierarchical description can be made to diminish. In particular we are interested in relating the combined size of P^k and P_k to the choice of k and κ.

Before embarking on such a discussion, it behooves us to define precisely what is meant by a polytope's "size". For polytopes in three dimensions or less, the incidence structure on the faces (edges, vertices and facets) can be described by a planar graph. Thus, Euler's formula $f + v - e = 2$ relates the number of vertices, edges and facets. If $f \geq 2$ and $v \geq 2$ then $e \geq v$ and $e \geq f$. Thus, for the remainder of this paper, we shall use the number of edges as a measure of the size of a three-dimensional polytope. This metric has the advantage of being invariant under duality, and thus provides no bias towards either inner or outer hierarchical representations. Thus, we define

$$|P| = |E(P)|$$

for polytopes in three or less dimensions.

Inner (resp. outer) hierarchical representations are build by removing successive sets of independent vertices (resp. defining hyperplanes) with bounded degree. The rate at which the size of the polytope diminishes is thus related to the size of the independent set of low degree vertices (resp. defining hyperplanes) that can be found. Kirkpatrick[6] gives a very straightforward argument which shows that an independent set of at least $\frac{n}{24}$ vertices whose degree does not exceed 11 can be found in $O(n)$ time, where n is the original number of vertices. Algorithm 4.1, given by Edelsbrunner[4] improves this fraction. The algorithm is initialised with a list L of the vertices of P in non-decreasing order of degree. He leaves as an exercise the problem of showing that it finds an independent set of at least $\frac{n}{7}$ vertices when $\kappa \geq 12$.

Edelsbrunner's result makes no assumptions about the sparsity of edges in the original polytope. Here we generalise his result to include some sensitivity

Algorithm 4.1

> **Let** $I := \emptyset$
>
> **Let** v be the first vertex in L.
>
> **while** $deg(v) \leq \kappa$ **do**
>
> > **if** v is not marked **then**
> >
> > > Set $I := I \cup \{v\}$
> > > Mark all vertices adjacent to v.
> >
> > **endif;**
> >
> > Set v to the next vertex in L
>
> **endwhile.**

to the ratio of facets to vertices, and hence the number of edges. We begin by defining some terms.

- Let $l = |L|$.
- Let $L_i = \{v \in L | deg(v) = i\}$ and let $l_i = |L_i|$.
- Let $A_i = \{v \in I | deg(v) = i\}$ and let $a_i = |A_i|$.
- Let I_n be $\bigcup_{i=1}^{n} A_i$.
- Let c be the average degree of the vertices in L.
- Let b_i be the number of vertices in L_i which have edges connecting them to vertices in I_{i-1}.
- Let x_i be the number of edges connecting vertices in A_i to vertices of higher degree.
- Let v, f, and e be the number of vertices (resp. faces, edges) on the polytope.
- Let $\beta = (f/v) + 1$

An immediate consequence of Euler's formula $v + f - e = 2$ is that

$$e < \beta v \tag{4.1}$$

Since $2e$ represents the sum of the degrees of all the vertices of P, we can conclude that

$$c \leq 2\beta. \tag{4.2}$$

Lemma 4.18 $\sum_{i=1}^{n} \frac{x_i - b_i}{i+1} \geq 0$

Proof: For the purpose of discussion, we associate a direction with edges connecting vertices of unequal degree. We call the vertex of lower degree the *tail*, and the other the *head*. We can establish an upper bound on b_n, the number of vertices of degree n with edges connecting them to accepted vertices of smaller degree. Each such vertex must be the head of an edge whose tail is an accepted vertex. The number b_n of such vertices cannot exceed the number of such edges. Clearly there are at most $\sum_{i=1}^{n-1} x_i$ such edges available. Of these, however, at least $\sum_{i=1}^{n-1} b_i$ have heads of degree less than n. Thus, we can conclude that

$$b_n \leq \sum_{i=1}^{n-1} x_i - \sum_{i=1}^{n-1} b_i \tag{4.3}$$

We show by induction on n that any non-negative decreasing sequence of coefficients, $z_1, \ldots, z_n$, satisfies the following equation:

$$\sum_{i=1}^{n} z_i(x_i - b_i) \geq 0 \tag{4.4}$$

For a basis, observe that $b_i = 0$ for $i \leq 3$. Assume that Equation 4.4 holds for $n \leq (m-1)$.

$$\begin{aligned} \sum_{i=1}^{m} z_i(x_i - b_i) &= \sum_{i=1}^{m-1} (z_i - z_m)(x_i - b_i) + z_m \sum_{i=1}^{m} (x_i - b_i) \\ &\geq \sum_{i=1}^{m-1} (z_i - z_m)(x_i - b_i) + z_m b_{m+1} \end{aligned} \tag{4.5}$$

Now, the first term, $\sum_{i=1}^{m-1}(z_i - z_m)(x_i - b_i)$ is non-negative by our inductive hypothesis. The second term $z_m b_{m+1}$ is non-negative since both factors are required to be non-negative. Thus, we can conclude that

$$\sum_{i=1}^{m} z_i(x_i - b_i) \geq 0 \tag{4.6}$$

Substituting $\frac{1}{i+1}$ for z_i in equation 4.6 gives

$$\sum_{i=1}^{m} \frac{x_i - b_i}{i+1} \geq 0 \tag{4.7}$$

as required. □

Lemma 4.19 $|I_n| \geq \sum_{i=1}^{n} \frac{l_i}{i+1}$

Proof: The algorithm will reject a vertex only if it is connected to a vertex already accepted. Consider the $l_i - a_i$ rejected vertices of degree i. Of these, b_i are rejected because there is an edge connecting them to a vertex in I_{i-1}. The remaining $l_i - a_i - b_i$ vertices are rejected because of edges connecting them to vertices in A_i. There are ia_i edges incident upon vertices in A_i. However, x_i of them lead to vertices of higher degree. This leaves $ia_i - x_i$ edges available to cause the rejection of a degree i vertices. Thus: $l_i - a_i - b_i \leq ia_i - x_i$ giving

$$a_i \geq \frac{l_i + x_i - b_i}{i+1} \tag{4.8}$$

Now, combining Equation 4.8 and Lemma 4.18 we obtain

$$\begin{aligned} |I_n| &\geq \sum_{i=1}^{n} \frac{l_i + x_i - b_i}{i+1} \\ &= \sum_{i=1}^{n} \frac{l_i}{i+1} + \sum_{i=1}^{n} \frac{x_i - b_i}{i+1} \\ &\geq \sum_{i=1}^{n} \frac{l_i}{i+1} \end{aligned} \tag{4.9}$$

as required. □

Lemma 4.20 *For any sequence of non-negative coefficients, $z_1, \ldots, z_n$, and any positive real coefficients $\lambda_1, \ldots, \lambda_n$, the following inequality holds:*

$$\left(\sum_{i=1}^{n} z_i\right)^2 \leq \left(\sum_{i=1}^{n} \lambda_i z_i\right)\left(\sum_{i=1}^{n} z_i/\lambda_i\right)$$

Proof: We first observe the fact that for any positive real λ_i and λ_j we have,

$$\frac{\lambda_i^2 + \lambda_j^2}{\lambda_i \lambda_j} \geq 2 \tag{4.10}$$

Now

$$\left(\sum_{i=1}^{n} z_i\right)^2 = \sum_{i=1}^{n} (z_i)^2 + 2 \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} z_i z_j \tag{4.11}$$

and,

$$\left(\sum_{i=1}^{n} \lambda_i z_i\right)\left(\sum_{i=1}^{n} z_i/\lambda_i\right) = \sum_{i=1}^{n} (z_i)^2 + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{(\lambda_i)^2 + (\lambda_j)^2}{(\lambda_i)(\lambda_j)} z_i z_j \tag{4.12}$$

Combining Equations 4.10, 4.11 and 4.12 yields the required result. $\square$

Lemma 4.21 [1] $|I| \geq l/(c+1)$

Proof: By definition, $|I| = |I_\kappa|$. From Lemma 4.19 we know that

$$|I| \geq \sum_{i=1}^{\kappa} \frac{l_i}{i+1}. \tag{4.13}$$

From the definitions of c and l, we have

$$\frac{l}{c+1} = \frac{\left(\sum_{i=1}^{\kappa} l_i\right)^2}{\sum_{i=1}^{\kappa}(i+1)l_i}. \tag{4.14}$$

From Lemma 4.20 (when $\lambda_i = i+1$) we have

$$\frac{\left(\sum_{i=1}^{\kappa} l_i\right)^2}{\sum_{i=1}^{\kappa}(i+1)l_i} \leq \sum_{i=1}^{\kappa} \frac{l_i}{i+1}. \tag{4.15}$$

Combining equations 4.13, 4.14 and 4.15 gives

$$|I| \geq \frac{l}{c+1} \tag{4.16}$$

as required. $\square$

We now turn our attention to establishing a lower bound on $|I|$, which respects the ratio between the number of vertices, v, and the number of faces, f, on the given polytope by proving the following theorem:

Lemma 4.22 *If* $\kappa \geq 12$ *then* $|I| \geq \frac{v}{2\beta+1}$

Proof: Recall that $\beta = (f/v)+1$. The sum of the degrees of all vertices is given by $2e$, and can be no less than $cl + (\kappa+1)(v-l)$. Combining with equation 4.1 and regrouping gives

$$l \geq \frac{(2\beta - \kappa - 1)v}{c - \kappa - 1}$$

which can be combined with Lemma 4.21 to yield

$$|I| \geq \frac{(2\beta - \kappa - 1)v}{(c-\kappa-1)(c+1)}$$

[1] Stated without proof as a "hint" to problem 9.9 in[4]

Of course, we would like to be able to substitute 2β for c in the above equation, while preserving the inequality. Consider the expression as a function of x so that

$$f(x) = \frac{(2\beta - \kappa - 1)v}{(x - \kappa - 1)(x + 1)}$$

Observe that the derivative of f with respect to x is given by

$$\frac{d}{dx} f(x) = -\frac{(2\beta - \kappa - 1)v}{(x - \kappa - 1)^2 (x + 1)^2}(2x - \kappa)$$

Recall from Equation 4.2 that $c \leq 2\beta$. If we choose κ, so that $\kappa \geq 4\beta$ ($\kappa \geq 12$ will suffice since $\beta < 3$) then the derivative is negative for $x \leq c$. As a result, we can be sure that $f(c) \geq f(2\beta)$, and hence

$$|I| \geq \frac{v}{2\beta + 1}$$

as required. □

Suppose we are given a polytope P with v vertices, f facets and e edges. Previously we defined $\beta = (f/v) + 1$. We can find and remove an independent set of low degree vertices of size at least $\frac{v}{2\beta+1}$. The value of β is not known for the resulting polytope, but Euler's equation prohibits it from exceeding 3. Thus, at worst we will be able to remove $k - 1$ more sets each consisting of at least $\frac{1}{7}$ of the remaining vertices. The result will be a polytope, P_k, such that

$$|V(P_k)| < v \left(\frac{2\beta}{2\beta + 1} \right) \left(\frac{6}{7} \right)^{k-1}$$

Now, $e = \beta v$, and $|P_k| < 3|V(P_k)|$, so we can conclude that

$$|P_k| < 3|P| \left(\frac{6}{7} \right)^{k-1} \left(\frac{2}{2\beta + 1} \right) \tag{4.17}$$

Similarly, we obtain an expression for the size of P^k, the polytope which is formed by removing k independent sets of defining hyperplanes. Let $\beta^\delta = (v/f) + 1$. Then

$$|P^k| < 3|P| \left(\frac{6}{7} \right)^{k-1} \left(\frac{2}{2\beta^\delta + 1} \right) \tag{4.18}$$

Adding equations 4.17 and 4.18 gives a bound on the combined size of P^k and P_k.

$$|P^k| + |P_k| < |P|(6) \left(\frac{6}{7} \right)^{k-1} \left(\frac{1}{2\beta + 1} + \frac{1}{2\beta^\delta + 1} \right). \tag{4.19}$$

To continue the simplification we first maximise the expression,

$$\frac{1}{2\beta+1}+\frac{1}{2\beta^{\delta}+1}.$$

To accomplish this, let $\alpha = f/v$, and substitute $\alpha+1$ for β, and $(\alpha^{-1}+1)$ for β^{δ}. After expansion, this gives

$$\frac{1}{2\beta+1}+\frac{1}{2\beta^{\delta}+1}=\frac{2(\alpha+\alpha^{-1})+6}{6(\alpha+\alpha^{-1})+13}. \tag{4.20}$$

Substituting $x = \alpha + \alpha^{-1}$ into equation 4.20 gives

$$\frac{1}{2\beta+1}+\frac{1}{2\beta^{\delta}+1}=\frac{2x+6}{6x+13}. \tag{4.21}$$

For positive x, this is a strictly decreasing function, and is thus maximised when x is minimised. Of course, x is minimised when $\alpha = \alpha^{-1} = 1$, giving $x \geq 2$, and hence

$$\frac{1}{2\beta+1}+\frac{1}{2\beta^{\delta}+1}\leq\frac{2}{5} \tag{4.22}$$

Substituting equation 4.22 back into equation 4.19 yields

$$|P^k|+|P_k|<6|P|\left(\frac{2}{5}\right)\left(\frac{6}{7}\right)^{k-1}. \tag{4.23}$$

The results of this chapter are summarised by the following theorem:

Theorem 4.23 *Given a (DCEL) representation of polytope $P \subseteq \Re^3$, it is possible to construct a sequence of nested polytopes $\mathcal{S} = P^k, P^{k-1}, \ldots, P^2, P, P_2, \ldots, P_k$ so that $\mathcal{C}(\mathcal{S})$ is a simple polytopal cell complex with $\rho < 72$, and so that $|P^k| + |P_k| < 6|P|\left(\frac{2}{5}\right)\left(\frac{6}{7}\right)^{k-1}$.*

Chapter 5

Computing $P \cap Q$

We now show how to use the cell complexes which are induced by the inner and outer hierarchical representations of two polytopes P and Q to compute their intersection. We assume that polytopes are represented in a (now standard) way using the doubly connected edge list (DCEL) representation described on page 9. To build a description of $P \cap Q$, it will suffice to compute a description of the edges of $\partial P \cap \partial Q$, with appropriate pointers into the DCEL descriptions of P and Q, along with new pointers for truncated edges of P and Q. Thus, we add two new slots to the representation of an edge to accommodate these two new pointers.

Suppose we were to begin by constructing inner and outer hierarchical representations of both P and Q, so that we have available descriptions of the t-simple cell complexes, $\mathcal{C}(P^w, \ldots, P, \ldots, P_x)$ and $\mathcal{C}(Q^y, \ldots, Q, \ldots, Q_z)$, with w (resp. x,y,z) large enough so that P^w (resp. P_x,Q^y,Q_z) is a simplex. We call these the *complete* hierarchical representations of P. Approaches to computing $P \cap Q$ which attempt to traverse the boundary of either P or Q, while keeping track of their location in the cell complex induced by the complete hierarchical representations of the other seem doomed to produce $\Theta(n \log n)$ algorithms at best. The problem is that the complete hierarchical representations have logarithmic depth, all of which might be used during the traversal. To avoid this difficulty, Chazelle limits the depth of his (inner) hierarchical representations to an appropriately chosen constant k. We adopt the same approach, by computing $C_p = \mathcal{C}(P^k, \ldots, P, \ldots, P_k)$ and $C_q = \mathcal{C}(Q^k, \ldots, Q, \ldots, Q_k)$. We then traverse the edges of the boundary, Σ, of the specially designed non-convex polytope, $(P^k \cap Q) \cup (Q^k \cap P)$. Observe that $(P^k \cap Q) \cap (Q^k \cap P) = P \cap Q$, and that any point on $\partial P \cap \partial Q$ lies on Σ. Note that the facets of Σ might not be simply connected since, as illustrated in Figure 5.1, facets of P may be perforated by facets of Q and vice versa. To traverse all the edges of Σ our algorithm will have to cross such perforated facets.

There is an alternate dual construction involving the surface $\Sigma^\delta = \partial((P \cup$

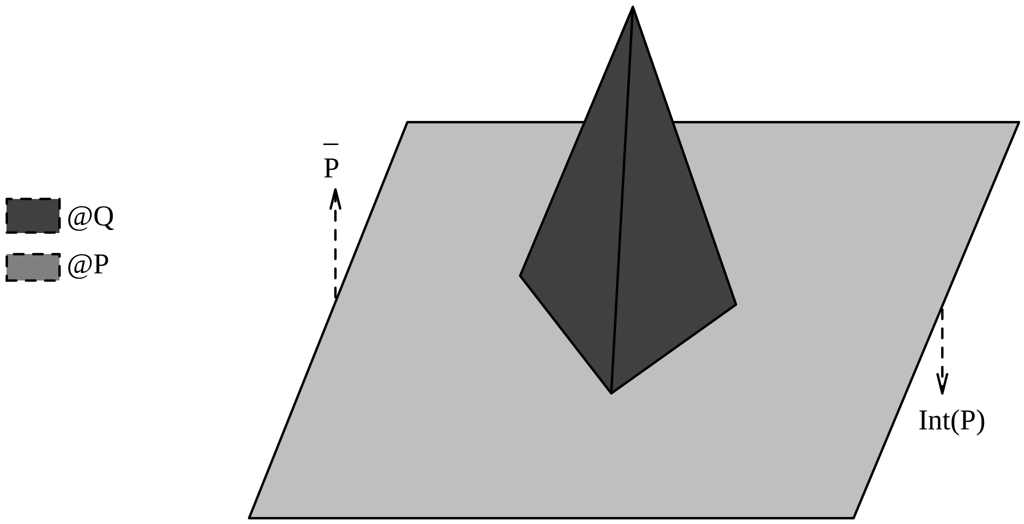

Figure 5.1: A perforated facet of $P \cup Q$

$Q_k) \cap (Q \cup P_k))$ which also appears to be easy to explore and yields a suitable approximation to $P \cap Q$. Σ^δ might appear to yield the desired result more directly, since it is the boundary of the intersection of approximations to P and Q, whereas Σ is the boundary of the union of such approximations. However, Σ^δ has the draw-back that its composite approximations are non-convex and hence turn out to be somewhat tedious to describe. In light of this, we confine our attention to the surface, Σ.

We begin by recursively computing $P^k \cap Q_k$ and $Q^k \cap P_k$. In section 5.6 we show how the constant k can be chosen so that the overall recurrence remains linear. We then make use of the description of these intersections, and the cell complexes $C_p = \mathcal{C}(P^k, \ldots, P_k)$ and $C_q = \mathcal{C}(Q^k, \ldots, Q_k)$ to traverse the edges of Σ taking constant time per edge, thus taking $O(|P| + |Q|)$ time in total. During the course of this traversal, most of $\partial P \cap \partial Q$ is explored. It may happen, however that some edges of $\partial P \cap \partial Q$ intersect the interior of facets of Σ, and hence are not explored during the edge traversal.

An example of such an edge arises when a facet g of Q is co-planar with a facet g' of Q^k, and intersects a facet f of P which is not co-planar with a facet of P^k. In this event, which is illustrated in Figure 5.2, $g \cap f$ is an edge of $Q \cap P$, but crosses the interior of $(g' \cap \widetilde{P}) \cup (g \cap P)$ [1], a facet of Σ. As it turns out, the edge in this example cannot be traversed in constant time, and thus we would like to be able to delay its treatment until section 5.5. The following lemma says more formally that such an edge is not an edge of Σ and thus permits us

[1]Recall the notation $\widetilde{P} = \overline{P} \cup \partial P$.

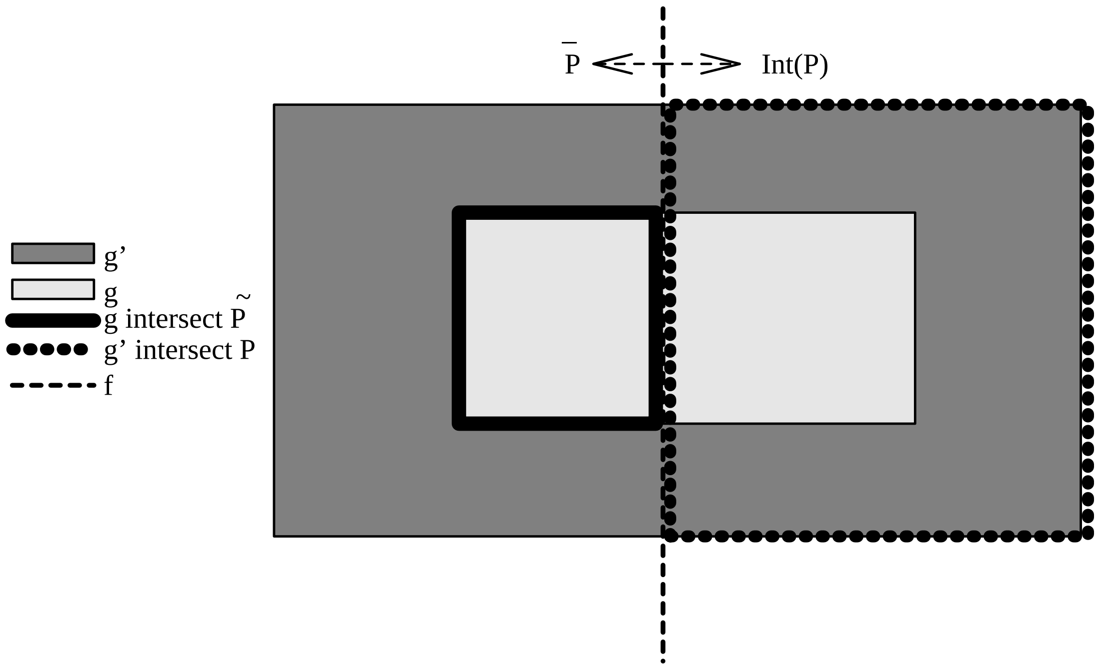

Figure 5.2: Coplanar facets of $P \cap Q^k$ and $Q \cap P^k$

to postpone its treatment:

Lemma 5.1 *Suppose that f and g are facets of P and Q respectively, that $f \cap g$ is not contained in an edge of f or g, and that there is no $f' \in F(P^k)$ such that $f \subseteq f'$. If $f \cap g$ is an edge of Σ, then there is no $g' \in F(Q^k)$ such that $g \subseteq g'$.*

Proof: Suppose that such a g' exists. Let H be the hyperplane containing g and g'. Observe that $a = g' \cap P$ is a facet of A, and that $b = g \cap P^k$ is a facet of B. Thus, H supports both A and B, and $a \cup b$ is a facet of Σ. The line segment $f \cap g$ crosses the interior of b and hence the interior of $a \cup b$ and hence cannot contain an edge of Σ. □

Although it may seem unfortunate that an edge traversal of Σ may not cross all the edges of $\partial P \cap \partial Q$, section 5.5 shows how to use the cell complexes, the descriptions of $P_k \cap Q^k$ and $Q_k \cap P^k$ and information obtained during the edge-traversal of Σ to traverse those which remain. In fact, implementations need not delay traversing these edges at all. They are treated separately here only to simplify the analysis of the algorithm.

To simplify the descriptions of the faces of Σ, we introduce the following notation:

Definition 5.1 *If f is a d-dimensional polytope, we let $(f)_\Sigma$ be the closure of the interior of $f \cap \Sigma$, in $\Re^d$.*

In particular, if f is a two-dimensional polytope (say a facet of P), and $f \cap \Sigma \subseteq \partial f$, then $(f)_\Sigma = \emptyset$, since $\mathrm{Int}(\partial f) = \emptyset$ in R^2.

There are four problems which need to be addressed, before we can claim to have an algorithm for computing $\partial P \cap \partial Q$. We must show that individual edges of Σ can be traversed in sufficiently little time. Since the facets of Σ need not be simply connected (some facets may be perforated), we must show how to navigate across such perforations. We must be able to locate a starting point on an edge of Σ. Finally, we need to demonstrate that we can efficiently compute descriptions of all of the edges of $\partial P \cap \partial Q$ which are not edges of Σ. To provide a basis for attacking these problems, we begin by classifying and describing the facets of Σ. Then each problem is addressed in turn.

5.1 Facets of Σ

Let $A = P \cap Q^k$ and let $B = Q \cap P^k$. Facets of Σ are of one of the following three types.

1. facets of A restricted to lie in $\widetilde{B}$
2. facets of B restricted to lie in $\widetilde{A}$
3. The union of co-planar facets, one from A and one from B.

Exploiting the obvious symmetry of Σ in A and B, we can ignore facets of type 2, which are obviously symmetric with type 1, and concentrate on facets of type 1 and type 3.

Facets of A, of course, are either facets of P restricted to lie inside Q^k, or they are facets of Q^k restricted to lie inside P. Since they turn out to have somewhat different properties, we consider separately the facets of P which are co-planar with facets of P^k from those which are not. Thus we identify three kinds of facets of A which might contribute two-dimensional components to type 1 or type 3 facets of Σ. These are the portions of facets of Q^k which lie in P^k, the portions of those facets of P which are co-planar with facets of P^k and lie in Q^k, and the portions facets of P which are not co-planar with any facet of P^k and lie in Q^k. The following three equations describe the portion of each which intersects Σ.

Facets $f \in F(Q^k)$:

$$\begin{aligned} f \cap \Sigma &= f \cap P \cap \widetilde{B} \\ &= f \cap P \cap (\widetilde{Q} \cup \widetilde{P^k}) \\ &= f \cap P \end{aligned} \tag{5.1}$$

Facets $f \in F(P)$ which are co-planar with a facet of P^k:

$$\begin{aligned} f \cap \Sigma &= f \cap Q^k \cap \widetilde{B} \\ &= f \cap Q^k \cap (\widetilde{Q} \cup \widetilde{P^k}) \\ &= f \cap Q^k \end{aligned} \tag{5.2}$$

Facets $f \in F(P)$ which are are not co-planar with a facet of P^k:

$$\begin{aligned} f \cap \Sigma &= f \cap Q^k \cap \widetilde{B} \\ &= f \cap Q^k \cap (\widetilde{Q} \cup \widetilde{P^k}) \\ &= (f \cap Q^k \cap \widetilde{Q}) \cup (f \cap Q^k \cap \widetilde{P^k}) \end{aligned} \tag{5.3}$$

In the first two of these cases, $f \cap \Sigma$ is simply a convex polytope. Thus, provided that $f \cap \Sigma$ is two dimensional, equations 5.1 and 5.2 accurately describe $(f)_\Sigma$. The final case, described by Equation 5.3 requires further study. Observe that since $f \in F(P)$, $f \cap \widetilde{P^k} \subseteq \partial f$. Thus,

$$\begin{aligned} \operatorname{Int}(f \cap \Sigma) &= \operatorname{Int}((f \cap Q^k \cap \widetilde{Q}) \cup (f \cap Q^k \cap \widetilde{P^k})) \\ &= \operatorname{Int}((f \cap Q^k \cap \widetilde{Q}) \cup (\partial f \cap Q^k \cap \widetilde{P^k})) \\ &= \operatorname{Int}(f \cap Q^k \cap \widetilde{Q}) \end{aligned} \tag{5.4}$$

and $(f)_\Sigma$ consists of the two-dimensional components of $(f \cap Q^k \cap \widetilde{Q})$. Thus, equations 5.1, 5.2 and 5.4 describe the portions of facets of A which contribute two dimensional components of facets of Σ.

5.2 Tracing Edges of Σ

We say an edge has been *traced* through a cell complex C if the sequence of cells which it intersects has been determined. It is our intention to show that given a starting point located in the navigable regions of C_p and C_q, any edge of Σ can be traced through C_p and C_q in $O(k)$ time by simulating a walk along the edge, while keeping track of our location in the two cell complexes simultaneously. In light of Theorem 3.11, there is no problem provided during the course of the walk, we remain in the navigable regions of C_p and C_q. Since $P \cap Q_k$ and $Q \cap P_k$ are both subsets of P^k and Q^k the only way such a walk can leave the navigable regions is to enter P_k or Q_k.

Let e be an edge of Σ. If $e \cap P_k$ is confined to an edge of P_k we call the intersection *trivial*. Trivial intersections with P_k or Q_k present no difficulty since they remain within the navigable regions of C_p and C_q. Through the following case analysis, it will be shown that any non-trivial intersection of an edge e with P_k (resp. Q_k) lies on an edge of $P_k \cap Q^k$ (resp. $Q_k \cap P^k$), or on a simple facet of P_k (resp. Q_k). Moreover, the edge or simple facet on which it lies can be identified in $O(k)$ time. If e is an edge of Σ and e is contained in an edge of P^k, then $e \cap Q_k$ will clearly be an identifiable edge of $P^k \cap Q_k$, and thus can be followed without difficulty since its description is available from the recursively computed description of $P^k \cap Q_k$. Similarly if e is contained in an edge of Q^k then e can be followed through P_k.

Let $\mathcal{F} = F(P) \cup F(Q) \cup F(P^k) \cup F(Q^k)$. Edges of Σ are the intersection of two non-coplanar facets $(f_1)_\Sigma$ and $(f_2)_\Sigma$, for some $f_1, f_2 \in \mathcal{F}$. Let us classify edges, according the origin of the facets which give rise to them by defining $E(X_1, X_2)$ to be the set of edges of Σ satisfying $e = (f_1)_\Sigma \cap (f_2)_\Sigma$ for some $f_1 \in F(X_1)$ and $f_2 \in F(X_2)$. Under such a classification scheme, there are 10 possible types of edges: $E(P,P)$, $E(P,P^k)$, $E(P,Q)$, $E(P,Q^k)$, $E(P^k,P^k)$, $E(P^k,Q)$, $E(P^k,Q^k)$, $E(Q,Q)$, $E(Q,Q^k)$ and $E(Q^k,Q^k)$.

Of course, $E(Q,Q^k)$ and $E(P,P^k)$ do not require separate treatment since they are subsets of $E(Q^kQ^k)$ and $E(P^k,P^k)$ respectively. We can also exploit the symmetry of Σ in P and Q, and hence discuss only the following five types of edges: $E(P,P)$, $E(P,Q)$, $E(P,Q^k)$, $E(P^k,P^k)$ and $E(P^k,Q^k)$.

5.2.1 Edges in $E(P,Q)$

Let f be a facet of P, and g be a facet of Q. Let $e = f \cap g$. Lemma 5.1 says that if $(e)_\Sigma$ is an edge of Σ, then either both f and g are co-planar with facets f' and g' of P^k and Q^k respectively, or neither are. If neither are, then both f and g must be simple, and hence point-location can be accomplished in both cell complexes in constant time over the entire edge.

Suppose, on the other hand, that both f and g are co-planar with facets f' and g' of P^k and Q^k respectively. If $e \cap P_k$ is non trivial, then there exists $f_0 \in P_k$ such that f is coplanar with f_0, and $e \cap P_k \subseteq f_0$. Thus, $e \cap P_k \subseteq g' \cap f_0$, which is, of course, an edge of $P_k \cap Q^k$. Similarly if $e \cap Q_k$ is non-empty, then there is a $g_0 \in F(Q_k)$ such that $e \cap Q_k = g_0 \cap f'$, which is an edge of $Q_k \cap P^k$. In both cases, the problem remains of identifying (by name) the appropriate edge of $P_k \cap Q^k$ or $Q_k \cap P^k$. Suppose that we are following e and that it intersects P_k. In this event, it crosses the boundary of f_0, at an edge x which can be determined from the description of C_p. The edge of $P_k \cap Q^k$ which we must identify, is one of the two edges bounding $f_0 \cap Q^k$ and incident upon $x \cap Q^k$.

5.2.2 Edges in $E(P, P)$

Let f_1 and f_2 be facets of P, and let $e = f_1 \cap f_2$. Then $e \in E(P)$, and $e \cap P_k$ is trivial. If $(e)_\Sigma \cap Q$ is confined to the boundary of Q, then it is contained in the intersection of a facet of P with a facet of Q, and is subsumed by the discussion of $E(P, Q)$ in section 5.2.1 above. Suppose, on the other hand that $(e)_\Sigma$ actually punctures Q's interior, and let $e' = (e)_\Sigma \cap \mathrm{Int}(Q)$. We shall show, that e' is contained in an edge of P^k.

Since e' is on Σ, it must be on the boundary of P^k. If it were interior to both P^k and Q then it would be interior to B and hence interior to $A \cup B$. Moreover, e' cannot intersect the interior of a facet of P^k. Suppose that e' did intersect the interior of a facet g' of P^k. From Equation 5.1 we know that $(g')_\Sigma = g' \cap Q$. Since $e' \subseteq \mathrm{Int}(Q)$, we can conclude that e' intersects the interior of $(g')_\Sigma$, contradicting the original supposition that e' is part of an edge of Σ.

Thus, e' is on the boundary of P^k, but does not intersect the interior of a facet, and hence is confined to the edge of P^k, which contains e.

5.2.3 Edges in $E(P, Q^k)$

Let f be a facet of P, g' be a facet of Q^k, and $e = f \cap g'$. Clearly, $e \cap P_k$ is an edge of $P_k \cap Q^k$. Suppose e has a non-trivial intersection with Q_k. Then there is a facet $g \in F(Q)$ which is co-planar with g', and $e \cap Q_k \subseteq g \cap f$, which has been discussed in section 5.2.1 above.

5.2.4 Edges in $E(P^k, P^k)$

Let $f_1{}'$ and $f_2{}'$ be facets of P^k. Let $e = f_1{}' \cap f_2{}'$. Since e is an edge of Σ, its intersection with P_k must be trivial, and its intersection with Q_k must be an edge of $Q_k \cap P^k$.

5.2.5 Edges in $E(P^k, Q^k)$

Let f' be a facet of P^k, g' be a facet of Q^k, and let $e = f' \cap g'$. If e intersects P_k (resp. Q_k), such an intersection is confined to the boundary of P_k (resp. Q_k), and hence is contained in $g' \cap f_0$ (resp. $f \cap g_0$) where f_0 (resp. g_0) is the facet of P_k (resp. Q_k) with which the edges intersects. Clearly $g' \cap f_0$ (resp. $f' \cap g_0$) is an edge of $Q^k \cap P_k$ (resp. $P^k \cap Q_k$).

We have shown that it is possible to trace each type of edge through C_p and C_q in $O(k)$, provided the recursively computed descriptions of $P_k \cap Q^k$ and $Q_k \cap P^k$ are available. Given a starting location on each connected component of the edges of Σ, this would immediately yield an $O(|P| + |Q|)$ algorithm for traversing all the edges.

5.3 Crossing Perforated Facets

Since, some facets of Σ are not simply connected, a mechanism for traversing individual edges does not immediately lead to a mechanism for traversing entire facet boundaries.

Lemma 5.2 *The only facets of Σ which are not simply connected are comprised of facets of P which are not co-planar with facets of P^k, or facets of Q which are not co-planar with facets of Q^k.*

Proof: Facets of Σ which are comprised of the union of two facets of A and B will be edge-connected provided the constituent facets are. From our earlier analysis of facets of A (equations 5.1, 5.2 and 5.3) we can see that if f is a facet of Q^k or a facet of P which is co-planar with a facet of P^k, then $(f)_\Sigma$ is a convex polytope, and hence must be edge-connected. □

Consider a facet f of P which is not co-planar with any facet of P^k. Then $(f)_\Sigma = f \cap Q^k \cap \widetilde{Q}$. Of course $f \cap Q^k$ is still convex and edge-connected. The disconnectedness of $(f)_\Sigma$ arises from the exclusion of the interior of Q, effectively taking a bite out of the facet. Such a bite could partion the facet into several components. Provided that the boundaries of Q and $f \cap Q^k$ intersect, however, the boundaries of each component will be connected.

The more difficult problem arises when Q intersects only the interior of $f \cap Q^k$. In this case $(f)_\Sigma$ is perforated by Q, and thus has two disconnected boundaries, namely $\partial(f \cap Q^k)$, and $\partial(Q \cap f)$. Given the location of a vertex v of $\partial(Q \cap (f)_\Sigma)$ in C_q, we can discover the location of a point on the boundary of $f \cap Q^k$ in $O(k)$ time. To accomplish this, we simply start walking in a straight line on f away from Q, tracking our location in the two cell complexes. On the other hand if we have traversed the boundary of $Q^k \cap f$, and would like to decide whether Q intersects $(f)_\Sigma$, locating a point in the boundary of the intersection if it does, we can rely on the "steepest descent" algorithm given in the proof of Theorem 3.12 to accomplish this in $O(k)$ time.

5.4 An Edge Traversal Algorithm for Σ

All that remains is to find a starting point for the edge traversal of Σ. To accomplish this, we begin by constructing the complete inner hierarchical representations of P and Q taking $O(|P|+|Q|)$ time. We can rely on the algorithm given in[3] (which uses these hierarchical representations) to provide a witness to the intersection of P and Q in $O(\log|P|+\log|Q|)$ time. We then navigate through the cell complexes induced by the complete inner hierarchical descriptions of P and Q in any direction until we puncture a facet σ of Σ. This will take at most $O(\log|P|+\log|Q|))$ time. We then continue to navigate in any direction on σ until we cross one of its edges. In total, this initialisation takes at most $O(|P|+|Q|)$ time.

To traverse all the edges of Σ, we start with a located point on the boundary of one of Σ's facets f. From there, we traverse the edges bounding f, taking $O(k)$ time per edge, and possibly $O(k)$ time to search for a possible perforation of f. We then proceed to explore the facets adjacent to f in the same way.

This algorithm takes constant $(O(k))$ time per edge (actually each edge will be explored twice since it bounds two facets), plus an additional $O(k)$ time per facet to check for perforations. Thus, in total we take $(e+f)O(k)$ time to complete the entire traversal where e and f are the number of edges and facets in Σ, plus $O(|P|+|Q|)$ time to find a starting point. Of course there are at most $O(|P|+|Q|)$ facets and at most $O(|P|+|Q|)$ edges in Σ, so that in total, the entire traversal takes only $O(|P|+|Q|)$ time since k is a fixed constant.

5.5 Filling in the Gaps

In the previous section, we have shown that the edges of Σ can be traversed in linear time. Of course the end goal is to traverse all of the edges of $\partial P\cap\partial Q$. Unfortunately some of these edges may have been missed during the traversal of Σ, and will need to be dealt with separately. To identify these edges, we examine the treatment of each facet of P with an eye towards intersections with facets of Q which may have been overlooked.

We first consider degenerate cases, in which co-planar facets of P and Q intersect. Suppose that f and g are co-planar facets of P and Q respectively, and that $f\cap g\neq\emptyset$. In this case, we need to ensure that a description of the facet $\sigma=f\cap g$ in $F(P\cap Q)$ has been constructed. Once the intersection of f and g has been detected, constructing their intersection is easily accomplished by one of the known linear time polygon intersection algorithms such as[8]. Thus, the real problem is to ensure that all such intersections are detected.

Let H be the hyperplane containing the two facets. If H separates P and Q, then $\sigma=P\cap Q$. This case can be identified at the outset, when a witness to the intersection of P and Q is computed. Suppose on the other hand that P and Q lie on the same side of H. If (without loss of generality) $f\subseteq g$, then locally, the boundary of $P \cap Q$ is simply the boundary of P. Hence this situation need not even be detected. If neither is strictly interior to the other, then their boundaries will intersect at a vertex of Σ and hence this situation can be detected during the traversal of Σ.

We now show that non-degenerate cases always involve facets of P and Q, at least one of which is co-planar with a facet of P^k or Q^k respectively. Suppose f is a facet of P which is not co-planar with a facet of P^k. The portion of f which contributed to a facet of Σ was given by $(f)_\Sigma = f \cap Q^k \cap \tilde{Q}$. If f is not co-planar with a facet of Q, and hence that intersections between f and ∂Q are one-dimensional in nature, we will have traced all intersections of f with ∂Q except for those which were co-incident with ∂Q^k. Thus, when looking for edges of $\partial P \cap \partial Q$ which are not edges of Σ, it suffices to examine only facets of P which are co-planar with facets of P^k and facets of Q which are co-planar with facets of Q^k. Without any loss of generality, we shall confine our attention to facets of P.

Let f be a facet of P which is coplanar with a facet f' of P^k. Assume that f (and hence f') intersects Q, since if it does not, then there is no edge of $\partial P \cap \partial Q$ to discover on f. Recall from Equation 5.2 that $(f)_\Sigma = f \cap Q^k$ and from Equation 5.1 that $(f')_\Sigma = f' \cap Q$. Observe that $f^* = (f)_\Sigma \cup (f')_\Sigma$ is a facet of Σ. While tracing the edges of Σ we have traversed the entire boundary of f^*, while maintaining our location in C_q. As part of this process, any intersection of ∂Q with ∂f, will have been discovered. Thus, if ∂Q intersects ∂f, then for each sequence of edges of $\partial P \cap \partial Q$ inside f we will have identified the two points at which it intersects the boundary of f, and these points will be located in C_q.

To trace the remainder of each of these edge sequences, we appeal to an algorithm for intersecting two convex polygons due to O'Rourke *et al.*[8] Let H be the hyperplane containing f and let $g = H \cap Q$. Observe that the edges we wish to traverse are the edges of g which lie inside $(f)_\Sigma$. Our traversal of $\partial(f)_\Sigma$ has identified the points, $\partial g \cap \partial(f)_\Sigma$. It remains to traverse the edges of g inside $(f)_\Sigma$ which lie between pairs of such points, taking $O(|f| + n)$ time in total, where n is the number of edges traversed.

The algorithm of O'Rourke *et al.* accomplishes precisely this, but relies on explicit descriptions of both f and g. Unfortunately, we have an explicit description only of f and computing the required description of g would take an excessive amount of time. To perform the required reduction, we must show that any question which might be asked of g can be answered from its implicit description $(H \cap Q)$ in constant time. Conveniently, given a point on $\partial f \cap \partial g$, the algorithm of O'Rourke *et al.* only requires a description of the portion of ∂g which actually intersects f. Since f is co-planar with $f' \in F(P^k)$, we can compute the endpoints of such edges simply by tracing through C_q the intersection of relevant facets of Q with H. If we should enter Q_k and hence leave the navigable region of C_q we can rely on the description of $f' \cap Q_k$ which must be available from the recursively obtained description of $P_k \cap Q^k$ to guide us.

The above discussion assumed that ∂Q actually intersects the boundary of

f and thus that starting points were available. Suppose on the other hand that it does not. If it intersects f at all, ∂Q must intersect the hyperplane H containing f at points strictly interior to f. To detect this situation, we simply start from any known point on $\partial(f)_\Sigma$ and perform a 'steepest descent' search (see Theorem 3.12) for ∂Q while remaining on the hyperplane containing f. Once this has been accomplished, we can use the C_q and the recursively obtained description of $f' \cap Q_k$ to compute $\partial Q \cap H$ taking $O(k)$ time per edge (facet of Q) traced.

5.6 Time Analysis

To analyse the time required by our algorithm as a function of the size of P and Q, we let $T(|P|, |Q|)$ represent the running time. The algorithm proceeds by first computing the inner and outer hierarchical representations of P and Q, using at most $O(|P| + |Q|)$ time. Then we make two recursive calls, requiring running time of $T(|P^k|, |Q_k|)$ and $T(|P_k|, |Q^k|)$ respectively. Finally, the intersection is computed using an additional $O(|P| + |Q|)$ time. Thus, the entire recurrence is expressed by

$$T(|P|, |Q|) = T(|P^k|, |Q_k|) + T(|P_k|, |Q^k|) + O(|P| + |Q|).$$

To conclude that $T(|P|, |Q|) = O(|P| + |Q|)$ we must choose k so that

$$|P^k| + |Q_k| + |P_k| + |Q^k| < \delta(|P| + |Q|) \tag{5.5}$$

for some constant $\delta < 1$.

It suffices, of course, to choose k such that $|P^k| + |P_k| < \delta|P|$ and $|Q^k| + |Q_k| < \delta|Q|$. Recall from Theorem 4.23 that

$$|P^k| + |P_k| < 6|P| \left(\frac{2}{5}\right) \left(\frac{6}{7}\right)^{k-1}.$$

Similarly,

$$|Q^k| + |Q_k| < 6|Q| \left(\frac{2}{5}\right) \left(\frac{6}{7}\right)^{k-1}.$$

Thus we need $(12/5)(6/7)^{k-1} \leq 1$, or $(6/7)^{k-1} \leq 5/12$. Letting $k = 7$ gives $(6/7)^{k-1} < 0.4 < 5/12$ as required.

Chapter 6

Conclusions

This thesis has described in detail a simple algorithm for computing the intersection of two three-dimensional convex polyhedra using $O(n)$ time and $O(n)$ space, where n is the combined number of edges in the polyhedra. In the process, we have fully exploited the structure of a polyhedral subdivision of $\Re^3$ induced by hierarchical representations of a given polytope. We expect that this induced cell complex will have other applications and hence have set out its properties separately in some detail.

To intersect two polyhedra, we use inner and outer hierarchical representations of Dobkin and Kirkpatrick to build simple polyhedral cell complexes around each of their boundaries. This provides a mechanism for detecting intersections between the two boundaries while traversing the edges of one. To avoid logarithmic cost per edge traversed, and hence an $O(n \log n)$ algorithm, we limit the depth of the cell complexes to a fixed constant k. This allows us to traverse each edge of the union of approximations to P and Q in constant ($O(k)$) time, using recursively obtained descriptions of $P^k \cap Q_k$ and $Q^k \cap P_k$. In this way, the problem of intersecting P and Q is reduced to one of navigating in shallow cell complexes close to the boundaries of P and Q.

The algorithm which has been presented bears considerable resemblance to that described by Chazelle[2]. Chazelle proceeds by identifying a point in $P \cap Q$, and then computing the geometric duals P^δ and Q^δ of P and Q about that point. One of the two polytopes is called the "anchor". Assume without loss of generality that it is P. He then triangulates the boundaries of P and P^δ, before computing their inner hierarchical representations for a sequence of at most k approximating polytopes. Recursively, he computes $P_k \cap Q$ and $P_k^\delta \cap Q^\delta$, being sure that Q is chosen as the anchor for the recursive calls. Once these preliminaries have been accomplished, the algorithm proceeds to traverse the boundary of P inside Q, and the boundary of P^δ inside Q^δ in search of intersections with the boundary of Q (laces). To accomplish such a traversal, it is necessary to pass from a lace of $P \cap Q$ to a corresponding lace on $P^\delta \cap Q^\delta$. A considerable portion of his paper is devoted to the description of such a correspondence, and the corresponding algorithm for computing such a passage. The algorithm presented here represents a simplification largely because it operates entirely in primal space, avoiding these complexities. Our algorithm is not just a re-interpretation of Chazelle's however. To stay entirely in primal space it is necessary to have two different kinds of navigation: ray tracing, and 'steepest descent'.

A second improvement over Chazelle's algorithm is a reduction in the number of recursive calls, and hence the linear constants. As described in his paper, Chazelle's algorithm gives rise to the following recurrence describing its running time where p and q represent the number of edges in P and Q respectively.

$$T(p,q) = 4T(3(6/7)^k p, 3(6/7)^k q) + O(p+q)$$

To compare the two algorithms fairly, we must refine the analysis of Chazelle's slightly. In the above recurrence, the constant factor 3 in the expressions $3(6/7)^k p$ and $3(6/7)^k q$ represents the worst case cost of triangulating the boundaries of P, P^δ, Q and Q^δ. Of course, in reality this worst case cannot be exhibited both by triangulating P and by triangulating P^δ. To capitalise on this observation we further remark that the running time is really dependent on the sum of p and q rather than on their independent values. The algorithm recursively computes four sub-problems: $P_k^\delta \cap Q_k^\delta$, $P_k^\delta \cap Q_k$, $P_k \cap Q_k^\delta$ and $P_k \cap Q_k$. To run in $O(n)$ time, the size of the sub-problems must diminish so that $|P_k| + |P_k^\delta| < \sigma|P|$ and $|Q_k| + |Q_k^\delta| < \sigma|Q|$ for some fixed positive σ less than $(1/2)$.

Suppose that P is a polytope with v, e and f vertices, edges and facets respectively. Let $\alpha = e/v$ and let $\alpha^\delta = e/f$. Observe that $\frac{1}{\alpha} + \frac{1}{\alpha^\delta} = 1 + \frac{2}{e}$. Let P_0 and P_0^δ be triangulated versions of P and P^δ respectively. Let f' and e' be the number of facets and edges of P_0. Since P_0 is triangulated, we have $e' = \frac{3}{2}f'$. Combining with Euler's equation gives $e' = 3v - 6 = \frac{3e}{\alpha} - 6$. Similarly, we obtain complimentary results for the size of P_0^δ. Combining gives

$$|P_0| + |P_0^\delta| = 3e\left(\frac{1}{\alpha} + \frac{1}{\alpha^\delta}\right) - 12.$$

Simplifying gives

$$|P_0| + |P_0^\delta| = 3e - 6 < 3e.$$

Thus, we can conclude that $|P_k| + |P_k^\delta| < 3(6/7)^k e$. Letting $k = 12$ gives $|P_{12}| + |P_{12}^\delta| < 0.47|P|$, guaranteeing a linear recurrence. On the other hand, our algorithm gives rise to the recurrence

$$T(|P|,|Q|) = T(|P^k|,|Q_k|) + T(|P_k|,|Q^k|) + O(|P| + |Q|)$$

which avoids one half of Chazelle's recursive calls, and becomes linear when $k \geq 7$.

Bibliography

[1] B. Chazelle and D. P. Dobkin. Intersection of convex objects in two and three dimensions. *Journal of the Association for Computing Machinery*, 34(1):1–27, 1987.

[2] Bernard Chazelle. An optimal algorithm for intersecting three-dimensional convex polyhedra. Technical report, Department of Computer Science, Princeton University, February 1989.

[3] David P. Dobkin and David G. Kirkpatrick. Fast detection of polyhedral intersection. *Theoretical Computer Science*, 27:241–253, 1983.

[4] H. Edelsbrunner. *Algorithms in Combinatorial Geometry.* Springer-Verlag, Heidelberg, Germany, 1987.

[5] S. Hertel, M. Mäntylä, K. Mehlhorn, and J. Nievergelt. Space sweep solves intersection of convex polyhedra. *Acta Informatica*, 21(5):501–519, 1984.

[6] David Kirkpatrick. Optimal search in planar subdivisions. *SIAM Journal on Computing*, 12, No. 1, 1983.

[7] D. E. Muller and F. P. Preparata. Finding the intersection of two convex polyhedra. *Theoretical Computer Science*, 7:217–236, 1978.

[8] Joseph O'Rourke, Chi-Bin Chien, Thomas Olson, and David Naddor. A new linear algorithm for intersecting convex polygons. *Computer Graphics and Image Processing*, 19:384–391, 1982.

[9] Preparata and Shamos. *Computational Geometry.* Springer-Verlag, New York, 1985.