# Determining the Castability of Simple Polyhedra

Prosenjit Bose *
Université du Québec à Trois-Rivières

David Bremner†
McGill University

Marc van Kreveld‡
Utrecht University

**Abstract**

A polyhedron $P$ is castable if its boundary can be partitioned by a plane into two polyhedral terrains. Castable polyhedra can be manufactured easily using two cast parts, where each cast part can be removed from the object without breaking the cast part or the object. If we assume that the cast parts are each removed by a single translation, it is shown that for a simple polyhedron with $n$ vertices, castability can be decided in $O(n^2 \log n)$ time and linear space using a simple algorithm. A more complicated algorithm solves the problem in $O(n^{3/2+\epsilon})$ time and space, for any fixed $\epsilon > 0$. In the case where the cast parts are to be removed in opposite directions, a simple $O(n^2)$ time algorithm is presented. Finally, if the object is a convex polyhedron and the cast parts are to be removed in opposite directions, a simple $O(n \log^2 n)$ algorithm is presented.

## 1 Introduction

A growing number of applications of computational geometry are in the area of automated manufacturing, where an engineer can design an object with the aid of a computer, and determine by which manufacturing process the object can be constructed. There are several types of manufacturing processes studied in computational geometry, such as gravity casting [4, 5, 6, 16], NC machining [4, 17], automated welding [22] and layer deposition methods

*Research supported by NSERC Grant OGP0183877 and a FIR Grant. This work was done while the author was at McGill University and the University of British Columbia. Département de Mathématiques et d'Informatique, Université du Québec à Trois-Rivières, C.P. 500, Trois-Rivieres, Québec G9A 5H7, Canada. E-mail: jit@uqtr.uquebec.ca.

†Research supported by an NSERC postgraduate fellowship. Address: School of Computer Science, McGill University, 3480 University St., Montreal, Quebec, H3A 2A7. E-mail: bremner@cs.mcgill.ca

‡Research supported by the ESPRIT Basic Research Action 7141 (project ALCOM II: *Algorithms and Complexity*). Address: Department of Computer Science, Utrecht University, POB 80.089, 3508 TB, Utrecht, The Netherlands. E-mail: marc@cs.ruu.nl

such as stereolithography [4, 3]. For a survey of this application area, the reader is referred to [7].

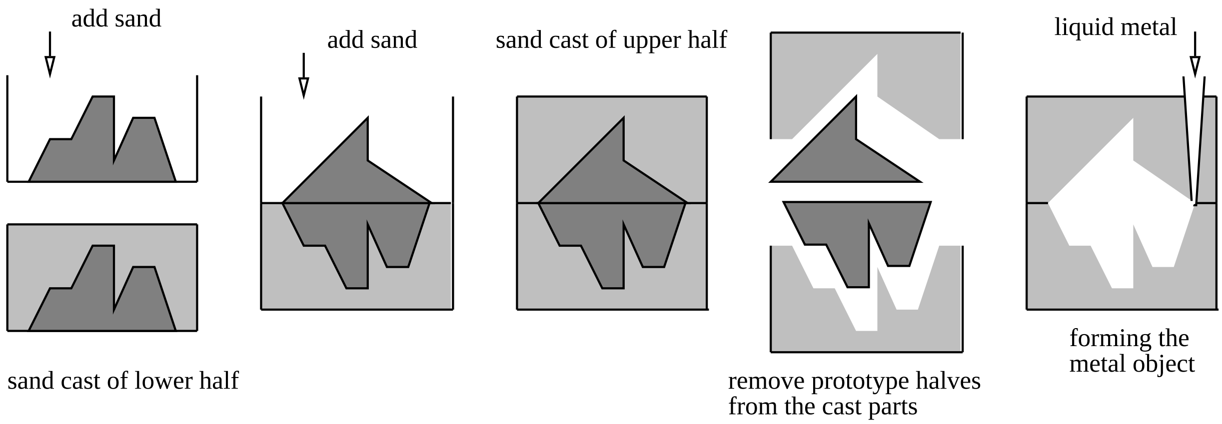


Figure 1: Construction of an object by sand casting, using two halves of the object as prototypes.

In this paper, we study the geometric and computational aspects of casting. Casting consists of filling the open region bounded by two or more cast parts with a material such as molten metal, then removing the cast parts once the metal has hardened. The necessity of removing the cast parts without breaking them imposes certain restrictions on the shape of the objects that can be constructed. For *sand casting* (see e.g. [15, 32]), only two cast parts are used. To construct the cast parts, a prototype of the object is first obtained (see Figure 1). The prototype is then divided into two parts along a plane. The facet of each prototype part adjacent to the cutting plane is referred to as the base. The first cast part is made by placing the base of the first prototype part on a flat surface, and then adding sand around it. The part is then rotated such that the base is facing up, and the other prototype part is placed such that the bases coincide. The second cast part is built by adding sand around this prototype part while maintaining a channel into the cavity. Once the sand hardens, the cast of the prototype object is complete and the prototype parts can be removed. To build a metal rendition of the prototype object with this cast, molten metal is poured into the opening until it fills the cavity. After the metal solidifies, the cast parts are removed from the object. The key to constructing a cast with this process is the ability to remove the prototype object without breaking the cast. This property is not restricted to casts built for manufacturing methods related to sand casting but also applies to other metal casting methods [15, 32], as well as injection molding and blow molding methods for plastics [27, 33]. The ability to remove the prototype object from the cast without breaking the cast allows one to re-use the same cast when mass-producing a particular object. Thus for several different manufacturing methods involving casting, the geometry of the object determines its feasibility of construction.

We note that more complicated objects can be made by using cores and inserts [15, 27, 32, 33]. However, their use slows down the manufacturing process and makes it more costly.

Thus to be cost efficient, cores and inserts should be avoided. We do not study the extra possibilities of cores and inserts in this paper. We also omit treatment of issues related to filling the cast, such as whether air bubbles are trapped by the rising liquid. For a geometric treatment of some of these issues, see [5, 6].

An object is *castable* if it can be manufactured by casting as described above. In other words, a cast of the object can be constructed such that each cast part can be removed from the object without breaking the object or any of the cast parts. Geometric and algorithmic issues of the castability of planar objects have been studied by Rosenbloom and Rappaport [28]. This paper addresses casting of objects modelled by polyhedra. A polyhedral surface $S$ is called a *weak terrain with respect to a direction* $\vec{d}$ if any line with orientation $\vec{d}$ intersects $S$ in a point or a line segment. In geometric terms, castability can be defined as follows (for a polyhedron $P$, $\partial P$ denotes the boundary of $P$, and for a plane $h$, $h^+$ and $h^-$ refer to the open half-spaces above and below $h$):

**Definition 1** *A simple polyhedron $P$ is* castable *if there exists a plane $h$ such that $h^+ \cap \partial P$ is a weak terrain w.r.t. some direction and $h^- \cap \partial P$ is a weak terrain w.r.t. some direction. The plane $h$ is called the* casting plane*, and the two directions are called* casting directions*.*

To manufacture a castable object (modelled as a polyhedron $P$), first determine a casting plane $h$. The plane $h$ divides $P$ into two cast parts. Make each cast part from the prototype halves $h^+ \cap P$ and $h^- \cap P$. Since $P$ is castable, the prototype halves can be removed from the cast parts, and later the manufactured object can be removed from the cast parts. We consider three versions of the castability problem. They differ in the way the cast parts may be removed from the polyhedron $P$. Figure 2 shows the three versions for planar polygons.

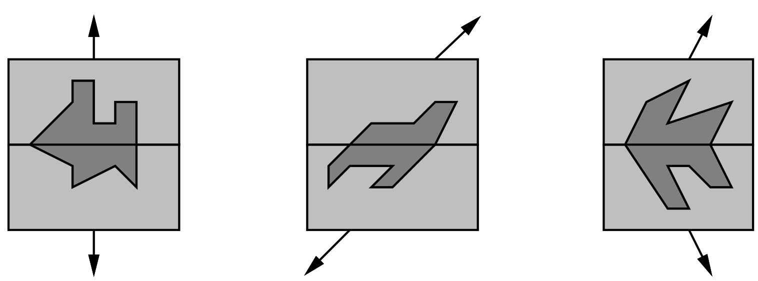

Figure 2: Three versions of the castability problem.

1. The two cast parts must be removed from $P$ by one translation each, in opposite directions, and normal to the casting plane (orthogonal cast removal).

2. The two cast parts must be removed from $P$ by one translation each, and in opposite directions (opposite cast removal).

| | space | | opposite & orthogonal | arbitrary |
|---|---|---|---|---|
| simple algorithms | $O(n)$ | convex polyhedra | $O(n \log^2 n)$ | $O(n^2 \log n)$ |
| | | simple polyhedra | $O(n^2)$ | $O(n^2 \log n)$ |
| best results (in theory) | $O(n^{3/2+\epsilon})$ | convex polyhedra | no improvement | $O(n^{3/2+\epsilon})$ |
| | | simple polyhedra | $O(n^{3/2+\epsilon})$ | $O(n^{3/2+\epsilon})$ |

Table 1: Summary of casting algorithms presented.

3. The two cast parts must be removed from $P$ by one translation each, in arbitrary directions (arbitrary cast removal).

Any convex polygon (in the plane) is castable in any of the three versions. In 3 dimensions, the equivalent property does not hold for convex polyhedra; in fact, many convex polyhedra are not castable in any of the three versions (see Appendix A). In manufacturing, developing machines that perform orthogonal and opposite cast removal is much simpler than developing machines that perform arbitrary cast removal. In fact, opposite cast removal seems to be the most popular technique used [10, 27]. Furthermore, if orthogonal or opposite cast removal is possible, it can be determined more efficiently. From a geometric point point of view, the differences between opposite and orthogonal castability are for the most part straightforward. In the interests of conciseness, we concentrate here on the opposite case, making note of the differences for the orthogonal case where they seem significant.

We summarize the complexity of the different algorithms we developed for the casting problem in Table 1. In the top half of the table, the time bounds of simple, linear space algorithms are shown. The bottom half of the table shows improvements made (in theory) by using $O(n^{3/2+\epsilon})$ storage (for any positive constant $\epsilon$).

We first derive some useful geometric properties of castable objects. These properties are the foundation upon which the algorithms are developed. The differences between the three versions of cast removal for both convex and simple polyhedra are considered (Section 2). The main approach taken by the algorithms is to first compute a set of candidate casting planes and candidate cast removal directions, and then to test each one of the planes and directions. Therefore, in Section 3, we give bounds on the maximum number of distinct (candidate) casting planes and cast removal directions. As a by-product that is interesting in its own right, we prove that the number of planes that intersect a convex polyhedron without intersecting any of its facets properly is $O(n)$, and that the total number of edges contained in these planes is $O(n \log n)$. Algorithms for cast removal in opposite directions are presented in Section 4. Section 5 presents the algorithms for cast removal in arbitrary directions.

## 2 Preliminaries

All polygons and polyhedra in this paper are simple (see [14, 25, 26] for any missing geometric definitions). For a polyhedron $P$, denote by $V$, $E$ and $F$ the set of its vertices, edges and facets. Edges that bound two parallel facets are not allowed; they can be removed without changing the shape of the polyhedron. Edges and facets are open; the closure of an open set $S$ is denoted $cl(S)$. The open interior of a polyhedron $P$ is denoted by $int(P)$, the open exterior by $ext(P)$, and the boundary of $P$ by $\partial P$. A face of $P$ refers to a vertex, edge, or facet of $P$. Although technically the object to be constructed is the interior of $P$, and the boundary of $P$ is part of the cast, with a slight abuse of notation, we nevertheless state that $P$ is castable or not castable.

If a plane $h$ satisfies Definition 1 for some polyhedron $P$ with cast removal directions $\vec{d_1}$ and $\vec{d_2}$, we say that $h$ is a *valid* casting plane for $P$. If $h$ is a valid casting plane and $\vec{d_1} = -\vec{d_2}$, we say that $h$ is an *opposite* casting plane for $P$.

Recall that a connected polyhedral surface $S$ is called a *weak terrain with respect to a direction* $\vec{d}$ if any line with orientation $\vec{d}$ intersects $S$ in a point or a line segment. A polyhedron $P$ is called a *weak terrain w.r.t. a facet* $Q$ *and a direction* $\vec{d}$ if $\partial P - Q$ is a weak terrain with respect to $\vec{d}$. In the rest of this paper we use *terrain* to mean *weak terrain*

For a non-vertical plane $h$, we denote by $h^+$ and $h^-$ the open half-spaces above and below $h$. If $h$ is vertical but does not contain a line parallel to the $y$-axis, then $h^+$ and $h^-$ denote the open half-spaces bounded by $h$ that contain the points $(0, \infty, 0)$ and $(0, -\infty, 0)$, respectively. If $h$ is vertical and contains a line parallel to the $y$-axis then $h^+$ and $h^-$ denote the open half-spaces bounded by $h$ that contain the points $(\infty, 0, 0)$ and $(-\infty, 0, 0)$, respectively. We use $h_0^+$ and $h_0^-$ to denote $h^+$ and $h^-$ translated so that the bounding plane intersects the origin. Given direction $\vec{d}$ and facet $f$, we say that $f$ is *compatible* with $\vec{d}$ if the inner product between $\vec{d}$ and the outward normal of facet $f$ is non-negative (i.e. $\vec{d}$ makes an angle of at most $\pi/2$ radians with the outward normal of $f$). We say that $f$ is *incompatible* with $\vec{d}$ if it is not compatible.

**Observation 1** *Let $P$ be a polyhedron and let $h$ be a plane that intersects $P$. The surface $\partial P \cap cl(h^+)$ is a terrain for direction $\vec{d}$ if and only if $\partial P \cap cl(h^+)$ is connected and every facet of $P$ that intersects $h^+$ is compatible with $\vec{d}$.*

Therefore, castability with respect to a plane $h$ is only determined by the facets of $P$ that intersect $h^+$ and the ones that intersect $h^-$. If $h$ is a casting plane for $P$, then $h$ can be perturbed if this does not involve new facets intersecting $h$. In the case of orthogonal cast removal, the only perturbation allowed is translation.

**Observation 2** *For castability with orthogonal cast removal, we may assume that the casting plane contains at least one vertex of $P$. For opposite and arbitrary cast removal, we may assume that the casting plane contains at least three vertices of $P$.*

For two polyhedra $P$ and $Q$ whose interiors lie on different sides of a plane $h$, and which are both bounded by the same facet $f$ that lies inside $h$, we define the *union* of $P$ and $Q$ as the polyhedron with all vertices of $P$ and $Q$, with all facets of $P$ and $Q$ except $f$, and with all edges of $P$ and $Q$ except the ones contained in $h$ that bound two parallel facets.

## 2.1 The sphere of directions

We represent the space of all directions in 3-space by the points on the surface of a sphere. Let $\mathcal{S}$ be the unit sphere centered at the origin $o$. Any point $p$ on $\mathcal{S}$ represents the direction $\overrightarrow{op}$. Let *north* and *south* denote the points on $\mathcal{S}$ that represent the $\vec{z}$ and $-\vec{z}$ directions. Let $\mathcal{E}$ denote the equator (the set of points $p \in \mathcal{S}$, such that $\overrightarrow{op} \cdot \vec{z} = 0$). For any point $p \in \mathcal{S}$, let $H(p)$ denote the open hemisphere representing all directions that make an angle less than $\pi/2$ radians with $\overrightarrow{op}$.

Let $P$ be a convex polyhedron, let $h$ be a casting plane and let $\vec{d}_1$ and $\vec{d}_2$ be the two cast removal directions, represented by points $d_1$ and $d_2$ on the sphere of directions. We re-orient $P$ and $h$ such that *north* is normal to $h$, thus $d_1$ and $d_2$ cannot both lie in the upper hemisphere or the lower hemisphere. Without loss of generality, let $d_1 \in cl(H(north))$ and $d_2 \in cl(H(south))$.

**Observation 3** *If a facet $f$ of $P$ intersects $h^+$, and $f$ has its outward normal represented by a point $q$ on $\mathcal{S}$, then $q \in cl(H(d_1))$. Similarly, if $f$ intersects $h^-$, then $q \in cl(H(d_2))$. Therefore, if $f$ intersects the casting plane $h$, then $q \in cl(H(d_1)) \cap cl(H(d_2))$ (recall that $f$ is open).*

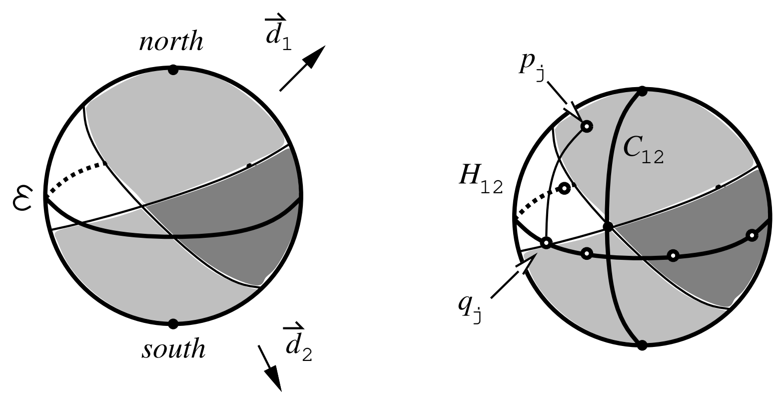


Figure 3: The sphere of directions. The shaded hemispheres are $H(d_1)$ and $H(d_2)$, and the darker shaded region is their intersection.

Define $C(d_1)$ and $C(d_2)$ to be the great circles that bound $H(d_1)$ and $H(d_2)$. If $\vec{d}_1$ and $\vec{d}_2$ are opposite, then $C(d_1) = C(d_2)$, otherwise, $C(d_1) \cap C(d_2)$ consists of a pair of antipodal points on $\mathcal{S}$ different from *north* and *south*.

For any point $p \in \mathcal{S} - \{north, south\}$, define $\lambda(p)$ to be the nearest point on the equator (i.e., the intersection point of the equator $\mathcal{E}$ with the great circle through *north* and $p$ nearest to $p$). By definition, we have

$$\overrightarrow{op} \cdot \overrightarrow{o\lambda(p)} > 0\,. \tag{1}$$

Furthermore, $p$ and $\lambda(p)$ lie to the same side of every great circle through *north* and *south* except the one containing them.

Assume that $\vec{d_1}$ and $\vec{d_2}$ are non-opposite in the following (see Figure 3). Define $C_{12}$ to be the great circle containing *north*, *south* and the points of $C(d_1) \cap C(d_2)$. Note that $cl(H(d_1)) \cap cl(H(d_2))$ does not intersect one of the (open) hemispheres defined by $C_{12}$. Let $H_{12}$ be this open hemisphere. By the above observation, any facet that has its outward normal in $H_{12}$ cannot be intersected by the casting plane. We use this fact in the following lemma.

**Lemma 1** *If a simple polyhedron $P$ is castable in non-opposite directions with casting plane $h$, then $h$ contains an edge of $P$.*

**Proof:** Let $Q = P \cap h$. If $Q$ consists of more than one connected component, or if $Q$ has polygonal holes, then $h$ cannot be a casting plane for $P$. Therefore, $Q$ is a polygon with no polygonal holes. Let $e_1, \ldots, e_m$ be the clockwise sequence of edges bounding $Q$ and let $q_1, \ldots, q_m$ be the points on $h \cap \mathcal{S}$ that represent the outward normals of $e_1, \ldots, e_m$. Since $h$ is chosen to be horizontal, $q_1, \ldots, q_m \in \mathcal{E}$. Every open half-circle in $\mathcal{E}$ contains at least one point of $q_1, \ldots, q_m$, because $Q$ is a simple polygon.

Given that $P$ is castable with respect to non-opposite directions $\vec{d_1}$ and $\vec{d_2}$, assume that every $e_i$ is the intersection of a facet $f_i$ of $P$ with the casting plane (i.e. no edge of $Q$ is an edge of $P$). Let $C_{12}$ and $H_{12}$ be as defined above, and let $e_j$ be an edge of $Q$ such that $q_j \in \mathcal{E} \cap H_{12}$ (see Figure 3). Let $p_j$ be the point on $\mathcal{S}$ that represents the outward normal of $f_j$. Then $q_j = \lambda(p_j)$, and by (1), we know $p_j$ lies in $H_{12}$. However, $H_{12}$ does not contain any point in $cl(H(d_1)) \cap cl(H(d_2))$, so by Observation 3 the facet $f_j$ cannot intersect the casting plane, which is a contradiction. Thus $h$ contains an edge of $P$. ∎

**Lemma 2** *If a simple polyhedron $P$ is castable with casting plane $h$ and in non-opposite directions, then $h$ contains an edge of the convex hull of $P$.*

**Proof:** Let $P$ be castable with respect to non-opposite directions $\vec{d_1}$ and $\vec{d_2}$. Let $Q = P \cap h$. The convex hull of $Q$ is the closure of a facet bounding both $CH(P \cap h^+)$ and $CH(P \cap h^-)$ (note that the convex hull is defined as a *closed* set). As in the proof of the previous lemma, there exists an edge $e_j$ of the convex hull of $Q$ where the outward normal of the edge on plane $h$ lies in $H_{12}$. We need to prove that $e_j$ is also an edge of $CH(P)$. Let $f_1$ be the facet of $CH(P \cap h^+)$ incident to $e_j$ and not in $h$. Define $f_2$ analogously for $CH(P \cap h^-)$. Let $q_j$, $p_1$ and $p_2$ be the points on $\mathcal{S}$ that represent the outward normals of $e_j$ (in $h$), $f_1$ and $f_2$, respectively. Since $f_1$ and $f_2$ are incident to $e_j$, we have $\lambda(p_1) = \lambda(p_2) = q_j$, so $p_1$, $p_2$ and $q_j$ lie on a half-circle between *north* and *south* and in $H_{12}$. Since $p_1 \in H(d_1)$ and $p_2 \in H(d_2)$ are both contained in $H_{12}$, the half-circle through *north*, *south*, $p_1$ and $p_2$ must contain a point $r$

that is not in $cl(H(d_1))$ nor in $cl(H(d_2))$. The plane $h'$ with normal $\overrightarrow{or}$ and containing $e_j$ has $CH(P \cap h^+)$ completely to the one side, with the exception of $cl(e_j)$. Similarly, $CH(P \cap h^-)$ lies completely to the one side of $h'$ with the exception of $cl(e_j)$. Since these convex hulls lie to the same side, it follows that $P$ lies completely to the one side of $h'$ with the exception of the endpoints of $e_j$, and possibly $e_j$ itself (if $e_j$ is an edge of $P$). Therefore, $e_j$ is an edge of $CH(P)$. ■

Notice that the above two lemmas imply that if a polyhedron is castable, but not with opposite cast removal, then the casting plane contains both an edge of $P$ and an edge of the convex hull of $P$ (this might be the same edge). This will aid considerably to determine castability with arbitrary cast removal.

## 2.2 Relation to linear programming

Let $P$ be a polyhedron and let $h$ be a plane. The plane $h$ partitions the set $V$ of vertices of $P$ into three subsets $V_h$, $V_h^+$ and $V_h^-$ of vertices in, above and below $h$, respectively. Similarly, $h$ partitions the set $E$ of edges of $P$ in four subsets $E_h$, $E_h^\times$, $E_h^+$ and $E_h^-$ of edges contained in $h$, intersecting $h$, above $h$ and below $h$, respectively. The set $F$ of facets is partitioned in the same way. For any facet $f \in F$, denote by $\Psi(f)$ the closed half-space bounded by a plane supporting $f$, and such that for any point in $f$, $\Psi(f)$ does not intersect the interior of $P$ in an $\epsilon$-neighborhood of the point. Denote by $\Psi_0(f)$ the same half-space, but translated such that the bounding plane contains the origin. We define

$$\xi^+(h) \equiv cl(h_0^+) \cap \left\{ \bigcap_{f \in F_h^+ \cup F_h^\times} \Psi_0(f) \right\} \quad \text{and} \quad \xi^-(h) \equiv cl(h_0^-) \cap \left\{ \bigcap_{f \in F_h^- \cup F_h^\times} \Psi_0(f) \right\}.$$

The intersection of a set of half-spaces is called *non-trivial* if it contains more than a single point, *trivial* otherwise. Denote by *refl*$(b)$ the reflection of an object $b$ through the origin (i.e. every point in $b$ is negated). We make the following observations.

**Observation 4** *The plane $h$ is a casting plane for polyhedron $P$ for arbitrary cast removal if and only if $\xi^+(h)$ and $\xi^-(h)$ are both non-trivial.*

**Observation 5** *The plane $h$ is a casting plane for polyhedron $P$ for opposite cast removal if and only if $\xi^+(h) \cap \mathit{refl}(\xi^-(h))$ is non-trivial.*

With the above observations, we can test efficiently whether a given plane $h$ is a casting plane for $P$ (for orthogonal cast removal, we merely intersect the cone of Observation 5 with a line perpendicular to $h$). Since the casting problem for a plane $h$ and a polyhedron $P$ can be transformed in linear time to a linear programming problem in 3 dimensions, the test requires only linear time [23, 31].

**Lemma 3** *Given a polyhedron $P$ and a plane $h$, one can test in linear time whether $h$ is a casting plane for $P$ in any of the three versions for removing the cast.*

Similarly, given a polyhedron and two cast removal directions (but not a casting plane), one can test using linear programming whether the polyhedron is castable with respect to those cast removal directions. This test is accomplished by reducing the problem to finding a weak separating plane for two sets of points. A *weak separating plane* for two sets of points is one that does not contain points from both point sets in either induced open halfspace (for a more thorough study of such problems, the reader is referred to Houle [18]). In the rest of this paper we use *separating plane* to mean weak separating plane.

The following lemma uses a well known duality between points and planes (see e.g. [14]). The duality transform $\mathcal{D}$ maps a point $p = (p_1, p_2, p_3)$ to the plane $h = \{x \in R^3 \mid 2p_1x_1 + 2p_2x_2 - x_3 = p_3\}$, and vice versa, that is, it maps a non-vertical plane $h$ to the point $\mathcal{D}(h)$ such that $\mathcal{D}(\mathcal{D}(h)) = h$. Note that $\mathcal{D}$ is not defined for vertical planes. The transform $\mathcal{D}$ has the following two fundamental properties:

1. A point $p$ belongs to plane $h$ if and only if point $\mathcal{D}(h)$ belongs to plane $\mathcal{D}(p)$.
2. A point $p$ lies in half-space $h^+$ ($h^-$) if and only if point $\mathcal{D}(h)$ lies in half-space $\mathcal{D}(p)^+$ ($\mathcal{D}(p)^-$).

**Lemma 4** *Given a set of red points $R$ and a set of blue points $B$ in $E^3$, let $H$ be the set of non-vertical planes $h$ such $R \subset cl(h^+)$ and $B \subset cl(h^-)$ and $h$ contains at least 3 elements of $R \cup B$. Let $n$ denote $|R \cup B|$.*

*(a)* $|H| \in O(n)$

*(b) A single element of $H$ can be computed in $O(n)$ time.*

*(c) All elements of $H$ can be computed in $O(n \log n)$ time.*

**Proof:** For each red point $r$, we generate a linear constraint $\mathcal{D}(r)^+$, and for each blue point $b$, we generate a linear constraint $\mathcal{D}(b)^-$. The set of (duals of) separating planes is precisely the convex polyhedron defined by these constraints. Since we insist that solution planes contain three members of $R \cup B$, our solution planes correspond to the vertices of this polyhedron. Part (a) of the lemma then follows from Euler's formula. Part (b) requires solving a 3-dimensional linear program (see e.g. [31, 23]), and Part (c) requires the intersection of $n$ 3-dimensional halfspaces (see e.g. [14, 26]). ■

**Lemma 5** *Given a polyhedron $P$ and two cast removal directions, one can test in linear time whether there exists a casting plane $h$ that allows removing the cast parts in the given directions.*

**Proof:** Let the two cast removal directions be $\vec{d_1}$ for $\partial P \cap h^+$ and $\vec{d_2}$ for $\partial P \cap h^-$. For every facet $f$ of $P$, one can determine whether $f$ should lie above the casting plane $h$ (is compatible only with $\vec{d_1}$), below $h$ (is compatible only with $\vec{d_2}$), may intersect $h$ (is compatible with both

$\vec{d_1}$ and $\vec{d_2}$) or is incompatible with the cast removal directions. If there is a facet of $P$ that is incompatible, then there does not exist any casting plane for directions $\vec{d_1}$ and $\vec{d_2}$.

The classification of the facets as "above", "below", and "intersect" imposes a classification of the edges. Any edge is classified either as "above/above" (a/a), "above/below" (a/b), "above/intersect" (a/i), "below/below" (b/b), "below/intersect" (b/i) or "intersect/intersect" (i/i), corresponding to the classification of the two facets incident to that edge.

Similarly, the classification of an edge determines where both endpoints of the edge must lie. For example, if an edge is classified as (a/a) then both endpoints must lie in $h^+ \cup h$. We summarize the implications that the classification of the edges has on their endpoints in Table 2.

| edge class. | endpoints |
|---|---|
| (a/a) | $h^+ \cup h$ |
| (a/i) | $h^+ \cup h$ |
| (b/b) | $h^- \cup h$ |
| (b/i) | $h^- \cup h$ |
| (a/b) | $h$ |
| (i/i) | anywhere |

Table 2: Classification of vertices from edge classification.

The classification of the endpoints of edges, in turn, determines where the vertices of $P$ must lie. If a vertex is adjacent only to (i/i) edges, then it can be anywhere with respect to the casting plane and plays no role in determining the position of the casting plane. These vertices can be ignored. However, for every other vertex, one can decide whether it must be contained in $h$, lie in $h^+ \cup h$, or lie in $h^- \cup h$. Let $V^{\subset}$, $V^+$, and $V^-$, respectively, represent these sets. We note that a vertex is adjacent only to (i/i) edges if and only if the cast removal directions are not opposite. In the case of opposite cast removal, every vertex is classified.

A valid casting plane $h$ is one that contains $V^+ \cup V^{\subset}$ in $cl(h^+)$ and $V^- \cup V^{\subset}$ in $cl(h^-)$. By Observation 2, we may assume $h$ contains 3 vertices of $V^+ \cup V^- \cup V^{\subset}$. By Lemma 4b, such a plane can be computed in $O(n)$ time. ∎

## 2.3 Convexity properties

In this subsection we derive some additional geometric properties of convex polyhedra that form the basis of faster algorithms. We show that for convex polyhedra, the linear programs defined in Section 2.2 need consider only a subset of the facets of the polyhedron. We also characterize the intersection of an opposite casting plane and a facet of a convex polyhedron.

If $P$ is a convex polyhedron, then the linear programming problems defined by $P$ and a candidate casting plane $h$ need not consider all facets of $F$, but only those intersecting $h$ and those adjacent to $h$. We make this more precise. For the subset $E_h$ of the edges of $P$ contained in $h$, let $F^+(E_h)$ denote the subset of $F^+$ of facets that contain at least one edge of $E_h$ in their closure. Define $F^-(E_h)$ analogously. Furthermore, we define

$$\phi^+(h) \equiv cl(h_0^+) \cap \left\{ \bigcap_{f \in F_h^\times \cup F^+(E_h)} \Psi_0(f) \right\} \quad \text{and} \quad \phi^-(h) \equiv cl(h_0^-) \cap \left\{ \bigcap_{f \in F_h^\times \cup F^-(E_h)} \Psi_0(f) \right\}.$$

**Lemma 6** *If $P$ is convex, $\xi^+(h) = \phi^+(h)$ and $\xi^-(h) = \phi^-(h)$.*

**Proof:** We only prove that $\xi^+(h) = \phi^+(h)$; the other proof is similar. Furthermore, that $\xi^+(h) \subseteq \phi^+(h)$ is trivial, so we prove $\phi^+(h) \subseteq \xi^+(h)$.

If $\phi^+(h)$ only contains the origin then so does $\xi^+(h)$. Otherwise, let $r$ be a half-line originating at the origin and inside $\phi^+(h)$. If $r \notin \xi^+(h)$, then there is a facet $f \in F_h^+ \setminus F^+(E_h)$ for which $r \notin \Psi_0(f)$. Let $\overline{\Psi}(f)$ denote the (closed) half-space supporting $f$ distinct from $\Psi(f)$. Since $P$ is convex,

$$f \subset cl(h^+) \cap \left\{ \bigcap_{f \in F_h^\times \cup F^+(E_h)} \overline{\Psi}(f) \right\}.$$

Since $r \in \phi^+(h)$, it follows that the projection of any point in $f$ in direction $-r$ onto $h$ will lie in $h \cap P$. But since $r \notin \Psi_0(f)$, the line segment connecting a point in $f$ with this projection will be (partially) outside $P$, namely, in the neighborhood of $f$. This contradicts the convexity of $P$. ∎

With Lemma 6, we conclude the following:

**Lemma 7** *The plane $h$ is a casting plane for a convex polyhedron $P$ for opposite cast removal if and only if $\phi^+(h) \cap refl(\phi^-(h))$ is non-trivial.*

We now consider the intersection of an opposite casting plane with a facet of a convex polyhedron.

Let $Q$ be a convex polygon. Let $L_0$ and $L_1$ be distinct parallel oriented lines of support for $Q$. We call the faces (vertices or edges) $q_0$ and $q_1$ of $Q$ contained, respectively, in $L_0$ and $L_1$ *$\vec{d}$-antipodal*, where $\vec{d}$ is the orientation of $L_0$ and $L_1$. We define the *$\vec{d}$-upper chain* of $Q$ as the chain from $q_0$ to $q_1$ where a ray with orientation $\vec{d}$ from $\partial Q$ is outside $Q$ in the neighbourhood of its endpoint. We call the other chain from $q_0$ to $q_1$ the $\vec{d}$-lower chain. Faces $q_0$ and $q_1$ are called *antipodal* if they are $\vec{d}$-antipodal for some direction $\vec{d}$.

**Lemma 8** *Let $P$ be a convex polyhedron, and $Q$ a facet of $P$. Let $\vec{d}$ be a direction contained in the plane induced by $Q$. Any facet adjacent to an edge on the $\vec{d}$-upper chain of $Q$ is compatible with $\vec{d}$ and incompatible with $-\vec{d}$.*

**Proof:** Let $e$ be an edge on the $\vec{d}$-upper chain of $Q$. We represent the space of directions in 3-space as a sphere of directions with $\vec{d}$ as *north*. Let $n_e$ be the point on the sphere of directions corresponding to the outward normal of edge $e$ (in the plane of $Q$). Since $e$ is on the $\vec{d}$-upper chain of $Q$, $n_e$ is in the open northern hemisphere. Let $n_q$ be the point on the sphere of directions corresponding to the outer normal of facet $Q$. Since $\vec{d}$ is contained in the plane induced by $Q$, $n_q$ and $-n_q$ are on the equator $\mathcal{E}$. Let $F$ be the facet adjacent to $e$ distinct from $Q$. Let $n_f$ be the point on the sphere of directions corresponding to the outward normal of facet $F$. By the convexity of $P$, we know that $n_f$ must be contained on the open arc of a great circle from $n_q$ to $-n_q$ through $n_e$. Every point on this arc is in the open northern hemisphere, so the outward normal of $F$ is compatible with $\vec{d}$ and incompatible with $-\vec{d}$. ■

**Lemma 9** *Let $h$ be a casting plane with casting directions $\vec{d}$ and $-\vec{d}$. Let $f$ be a facet of $P$ intersected by $h$. Let $q_0$ and $q_1$ be faces (edges or vertices) of $f$ intersected by $h$. Faces $q_0$ and $q_1$ are $\vec{d}$-antipodal.*

**Proof:** By Observation 3, $\vec{d}$ must be parallel to $f$. By Lemma 8, a casting plane with casting direction $-\vec{d}$ cannot intersect the $\vec{d}$-upper chain of $Q$, since that would cause a facet incompatible with $\vec{d}$ to intersect the casting plane. Similarly, interchanging the roles of $\vec{d}$ and $-\vec{d}$ in Lemma 8, a casting plane with casting direction $\vec{d}$ cannot intersect the $\vec{d}$-lower chain of $Q$. ■

## 2.4 Relation between simple polyhedra and convex polyhedra

In this subsection, we develop crucial links between simple polyhedra and convex polyhedra in terms of castability. In essence, we show that many of the problems concerning a simple polyhedron $P$ can be solved by first considering the convex hull of $P$. The main theorem that we establish is the following.

**Theorem 1** *If a simple polyhedron $P$ is castable with a plane $h$ that does not intersect any edge of $P$ properly and directions $\vec{d}$ and $-\vec{d}$, then $CH(P)$ is also castable with plane $h$ and directions $\vec{d}$ and $-\vec{d}$; furthermore $h$ does not properly intersect any edge of $CH(P)$ either.*

To prove the theorem, we first establish a few relevant lemmas.

**Lemma 10** *A convex polyhedron $P$ is a terrain with respect to a facet $Q$ and a direction $\vec{d}$ if and only if the vertices of $P$ project into $cl(Q)$ when projected in direction $-\vec{d}$ onto the supporting plane of $Q$.*

**Proof:**

($\Rightarrow$) If $P$ is a terrain with respect to a direction $\vec{d}$ and a facet $Q$, then every point of $P$ projects into $cl(Q)$ in direction $-\vec{d}$.

($\Leftarrow$) Suppose every vertex of $P$ projects into $cl(Q)$ in direction $-\vec{d}$. Since $P$ is convex, the line segment from every vertex $v$ to $Q$ in direction $-\vec{d}$ must be inside $P$. It follows that a ray with direction $\vec{d}$ from every vertex is outside $P$. By Observation 1, $P$ is a terrain with respect to $\vec{d}$ and $Q$. ∎

**Lemma 11** *If a polyhedron $P$ is a terrain with respect to a direction $\vec{d}$ and facet $Q$ then $CH(P)$ is a terrain with respect to $\vec{d}$ and $CH(Q)$.*

**Proof:** Every vertex of $P \setminus Q$ is on one side of the plane induced by $Q$; it follows that the convex hull of $Q$ must be a facet of $CH(P)$. Since every vertex of $CH(P)$ is a vertex of $P$, every vertex of $CH(P)$ must project into $CH(Q)$ in direction $\vec{d}$. By Lemma 10, $P$ is a terrain with respect $\vec{d}$ and $CH(Q)$. ∎

**Proof:** (of Theorem 1) Let $P$ be a simple polyhedron, and let $h$ be a casting plane for $P$ with cast removal directions $\vec{d}$ and $-\vec{d}$. Plane $h$ does not intersect any edge of $P$ properly. Let $Q$ be the simple polygon $P \cap h$. By definition, every vertex of $Q$ is a vertex of $P$, and every edge of $Q$ is either an edge of $P$ or the intersection of a facet of $P$ with the casting plane.

Let $P^+$ and $P^-$ denote $CH(P \cap h^+)$ and $CH(P \cap h^-)$ respectively. By Lemma 11, the polyhedra $P^+$ and $P^-$ are also castable for casting plane $h$ and directions $\vec{d}$ and $-\vec{d}$ respectively. Let $P'$ denote $P^+ \cup P^-$.

We begin by showing that $P'$ is convex. To do this, we show that the line segment between every pair of points in $P'$ is contained in $P'$. Since both $P^+$ and $P^-$ are convex, we simply need to show that for any point $a$ in $P^+$ and any point $b$ in $P^-$, the line segment $\overline{ab}$ is contained in $P'$. Let $a$ be an arbitrary point in $P^+$ and $b$ an arbitrary point in $P^-$. Let $a'$ be the projection of $a$ onto $h$ in direction $-\vec{d}$ and let $b'$ be the projection of $b$ onto $h$ in direction $\vec{d}$. Note that $a, a', b, b'$ are coplanar since both line segments, $\overline{aa'}$ and $\overline{bb'}$, are parallel to direction $\vec{d}$. By Lemma 11, both $a'$ and $b'$ are contained in $CH(Q)$. If $a'$ and $b'$ coincide, then the line segment $\overline{ab}$ is contained in $P'$. If $a'$ and $b'$ do not coincide, then line segment $\overline{ab}$ intersects line segment $\overline{a'b'}$. Let $x$ denote this intersection point. Since $x$ must be contained in $CH(Q)$, by construction, we have that $\overline{ax} \in P^+$ and $\overline{bx} \in P^-$. Therefore, $\overline{ab} \in P'$.

From the construction of $P'$ we observe that $P^+ \subseteq CH(P)$, $P^- \subseteq CH(P)$, and $P \subseteq P'$. Given that $P' \subseteq CH(P)$ and $P'$ is convex, we conclude that $P'$ is the convex hull of $P$.

Since every vertex of $CH(Q)$ is a vertex of $P$, $h$ does not intersect any edge of $CH(P)$ properly.

∎

**Lemma 12** *Let $h$ be plane, let $C_1$ and $C_2$ be convex polygons in $h$ such that $C_1 \subseteq C_2$, and let $S$ be a set of points entirely contained in one of the half-spaces bounded by $h$. If $CH(C_1 \cup S)$ is a terrain with respect to a direction $\vec{d}$ and facet $C_1$, then $CH(C_2 \cup S)$ is a terrain with respect to $\vec{d}$ and $C_2$.*

**Proof:** Suppose that $CH(C_1 \cup S)$ is a terrain with respect to $\vec{d}$ and $C_1$. By Lemma 10, $S$ projects inside $C_1$ in direction $-\vec{d}$. Since $C_1 \subseteq C_2$, $S$ also projects inside $C_2$ in direction $-\vec{d}$. By Lemma 10, $CH(C_2 \cup S)$ is a terrain with respect to direction $\vec{d}$ and facet $C_2$. ∎

**Theorem 2** *If a simple polyhedron $P$ is castable, then the convex hull of $P$ is also castable using the same casting plane and cast removal directions.*

**Proof:** Let $P$ be a simple polyhedron, and let $h$ be a casting plane for $P$ with casting directions $\vec{d_1}$ for $\partial P \cap cl(h^+)$ and $\vec{d_2}$ for $\partial P \cap cl(h^-)$. The polyhedron $CH(P \cap h^+) \cup CH(P \cap h^-)$ is also castable for casting plane $h$ and directions $\vec{d_1}$ and $\vec{d_2}$ by Lemma 11. Denote $P^+ = CH(P \cap h^+)$ and $P^- = CH(P \cap h^-)$.

We need to show that $P_H = CH(P)$ is castable with casting plane $h$ and casting directions $\vec{d_1}$ and $\vec{d_2}$. Let $P_H^+ = CH(P_H \cap h^+)$ and $P_H^- = CH(P_H \cap h^-)$. Since $P^+$ is contained in $P_H^+$ and $P^-$ is contained in $P_H^-$, the theorem follows from Lemma 12. ∎

Although initially one might believe Theorem 2 to be more important than Theorem 1, Theorem 2 in itself is not quite as useful as one might suspect. By Observation 2, we see that if a convex polyhedron has a valid casting plane, it must have a valid plane which contains at least three of its vertices. However, if a simple polyhedron $P$ is castable with casting plane $h$, even though $h$ is a valid casting plane for $CH(P)$, $h$ does not necessarily contain at least three vertices of $CH(P)$. As such, Theorem 2, although interesting, is not computationally useful.

# 3 The number of distinct casting planes and directions

The properties of castable polyhedra uncovered in the previous sections provide the framework for our solution to the casting problem. In the case of opposite cast removal, the approach taken by the algorithms to determine if a polyhedron $P$ is castable is to first compute a set of *candidate* casting planes $\mathcal{H}(P)$ and a set of *candidate* casting directions $\Delta(P)$ such that if $P$ is castable, then either $\mathcal{H}(P)$ contains a valid opposite casting plane, or there is a $\vec{d} \in \Delta(P)$ such that $\vec{d}$ and $-\vec{d}$ are valid casting directions. Next, each of the planes in $\mathcal{H}(P)$ and each of the directions in $\Delta(P)$ is tested for validity. In order for this approach to be efficient, the size of $\mathcal{H}(P)$ and $\Delta(P)$ must be bounded, which is the topic of this section.

Given a polyhedron $P$ with vertex set $V$, two planes $h_1$ and $h_2$ are *(combinatorially) distinct* if the partitioning of the facets into $F^+$, $F^-$, $F^{\subset}$ and $F^{\times}$ they define is different. We denote the set of all distinct valid casting planes of a polyhedron $P$ by $H(P)$. It is important to note the difference between the set $H(P)$ and $\mathcal{H}(P)$. A plane in set $H(P)$ is a valid casting plane by definition whereas a plane in the set $\mathcal{H}(P)$ may or may not be a valid casting plane.

By Observation 2, a trivial upper bound on the number of distinct casting planes for a polyhedron with $n$ vertices is $O(n^3)$. For non-opposite cast removal, Lemma 1 tightens this bound to $O(n^2)$. In this section we provide tighter bounds for opposite cast removal. Notice that a casting plane either intersects no edge of $P$ properly or intersects at least one edge of $P$ properly. This divides the set of distinct casting planes $H(P)$ into two groups, denoted $H^{\subset}(P)$ and $H^{\times}(P)$, respectively. Similarly, the set of candidate casting planes $\mathcal{H}(P)$ is divided into two sets $\mathcal{H}^{\times}(P)$ and $\mathcal{H}^{\subset}(P)$ with the following property: If the set $H^{\subset}(P)$ (resp. $H^{\times}(P)$) is non-empty then at least one of the planes in set $\mathcal{H}^{\subset}(P)$ (resp. $\mathcal{H}^{\times}(P)$) is valid. The distinction between planes that intersect an edge properly and those that do not is important from both a computational and geometric point of view. Considering opposite cast removal for a simple polyhedron $P$, if a plane $h$ is contained in $H^{\subset}(P)$ then by Theorem 1, only the convex hull of $P$ needs to be considered because $h$ is also contained in $H^{\subset}(CH(P))$. In the case of opposite cast removal, we provide a linear bound on $H^{\subset}(P)$ and $H^{\times}(P)$ for convex polyhedra. The proofs are constructive; thus, from the proofs we can construct sets $\mathcal{H}^{\times}(P)$ and $\mathcal{H}^{\subset}(P)$ each of linear size. For a simple polyhedron $P$, Theorem 1 shows that $H^{\subset}(P) \subseteq H^{\subset}(CH(P))$ which implies a linear bound on the set $H^{\subset}(P)$. Again from the proof we can construct a set $\mathcal{H}^{\subset}(P)$ of linear size. Although for simple polyhedra we do not provide a linear bound on $H^{\times}(P)$, we are able to construct a set $\Delta^{\times}(P)$ of linear size with the property that if $P$ is castable with some plane $h \in H^{\times}(P)$ and directions $\vec{d}$ and $-\vec{d}$, then $\vec{d} \in \Delta^{\times}(P)$. For arbitrary cast removal, we do not need to construct sets of candidate casting planes or directions. Since Lemma 1 and Lemma 2 show that a valid casting plane must contain an edge of the polyhedron and an edge of its convex hull, we use a different approach, which is detailed in Section 5.

For opposite cast removal, we have the following properties of intersections of a casting plane $h$ and a polyhedron $P$:

(P1) The facets $F^{\times}$ that intersect $h$ properly have their outward normals such that when translated to the origin, they are contained in a plane (since $\bigcap\{\Psi_0(f) \mid f \in F^{\times}\}$ must contain a line through $o$).

(P2) All edges that intersect $h$ properly are parallel (otherwise the incident facets span more than a plane).

(P3) Any two vertices in the closure of a facet $f$ and in $h$ are antipodal in $cl(f)$. Any vertex and edge in the closure of $f$ and intersecting $h$ are antipodal in $cl(f)$. (See Lemma 9.)

The set of distinct casting planes $H(P)$ is divided into two groups: the set $H^{\subset}(P)$ that intersect no edge of the polyhedron properly and the set $H^{\times}(P)$ those that intersect at least one edge of the polyhedron properly. We first concentrate on those that intersect at least one edge properly and then address those that intersect no edge properly.

Let $P$ be a simple polyhedron with $n$ vertices. Since orthogonal cast removal is simply a special case of opposite cast removal, all bounds for opposite cast removal apply to orthogonal cast removal. Therefore, we restrict our discussion to opposite cast removal. We begin by exploring some of the implications of property P2.

Let $h$ be a casting plane in $H^{\times}(P)$ with cast removal directions $\vec{d}$ and $-\vec{d}$. By property P2, all edges of $P$ intersected by $h$ must be parallel. Let $e$ be one such edge. The cast removal directions $\vec{d}$ and $-\vec{d}$ must be parallel to the line containing $e$ otherwise by Observation 3 one of the facets adjacent to $e$ violates the validity of casting plane $h$. Let $E_1, E_2, \ldots, E_k$ be a partitioning of $E$ into maximal sets of parallel edges. Given a set of parallel edges $E'$ we say that a direction is parallel to $E'$ if it is parallel to each edge in $E'$. Given any plane $h$ in $H^{\times}(P)$, the cast removal directions $\vec{d}$ and $-\vec{d}$ for $h$ must be parallel to $E_i$ for some $i$. Since a simple polyhedron has at most $O(n)$ edges, we conclude with the following:

**Lemma 13** *Let $P$ be a simple polyhedron with $n$ vertices. Let $\Delta^{\times}(P) = \{\vec{d} \mid \vec{d}$ is parallel to one of $E_i,\ i = 1, \ldots, k\}$ and $D^{\times}(P) = \{\vec{d} \mid \vec{d}$ and $-\vec{d}$ are valid cast removal directions for $h \in H^{\times}(P)\}$.*

*(a)* $D^{\times}(P) \subseteq \Delta^{\times}(P)$,

*(b)* $|\Delta^{\times}(P)| \in O(n)$, *and*

*(c)* $\Delta^{\times}(P)$ *can be computed in* $O(n \log n)$ *time.*

In the case of opposite cast removal of convex polyhedra, we prove that $H^{\times}(P)$ has linear size. The proof is constructive and provides an $O(n \log n)$ time method for constructing the candidates $\mathcal{H}^{\times}(P)$ for convex polyhedra. The following lemma is crucial in the development of more efficient algorithms for determining the castability of convex polyhedra.

**Lemma 14** *Given a convex polyhedron $P$, the number of distinct opposite casting planes that intersect some edge of $P$ properly is at most linear in the number of vertices of $P$.*

**Proof:** Let $E'$ be a maximal subset of parallel edges of $P$, of which at least one edge is properly intersected by some opposite casting plane. By convexity of $P$, such a casting plane must intersect the closure of all edges of $E'$, because no such closure of an edge can be strictly above or below the casting plane. The cast removal directions are parallel to $E'$, and by the classification defined in the proof of Lemma 5, for any casting plane $h$ with these cast removal directions, the vertices of $P$ are partitioned into $V^+ \subset cl(h^+)$, $V^- \subset cl(h^-)$, and $V^{\subset} \subset h$. In particular, the endpoints $G$ of the edges in $E_i$ are classified into sets $G^+ \subset cl(h^+)$ and $G^- \subset cl(h^-)$.

Consider an arbitrary casting plane $h$ properly intersecting the edges of $E'$. Since $P$ is convex, if $v \in h$ but $v \notin G$, then $v \in V^{\subset}$. This can be seen as follows. If $v$ is adjacent to an edge of $P$ contained in $h$, then $v$ is classified as "above/below" in the language of Lemma 5, hence must be contained in $V^{\subset}$. Otherwise $v$ is adjacent to two facets $F_l$ and $F_r$ parallel to $E'$. Since none of the edges adjacent to $v$ are parallel to $E'$ (otherwise they would be *in* $E'$), there must be facets joining $F_l$ and $F_r$ in cyclic order about $v$. By the convexity of $P$, these "joining" facets force $v$ to be in $V^{\subset}$.

Any valid casting plane properly intersecting $E'$ must separate $G^+$ from $G^-$. Furthermore, we may assume that any casting plane $h$ with cast removal directions parallel to $E'$ contains at least three members of $G \cup V^{\subset}$. If $V^{\subset}$ contains 3 or more vertices then this is trivial, so we consider the case where $V^{\subset}$ contains two or fewer vertices. From the preceding argument we see that if $V^{\subset}$ contains less than three vertices, then so does any opposite casting plane $h$. Since the only edges intersected properly by $h$ are in $E'$, we can perturb $h$ so that it intersects the required number of vertices of $G$ without intersecting any new facets.

It follows that any valid casting plane that properly intersects the edges of $E'$ must separate $G^{+} \cup V^{\subset}$ from $G^{-} \cup V^{\subset}$ and contain at least 3 members of $G \cup V^{\subset}$. By Lemma 4 there are $O(|E'|)$ such planes. Since every edge of $P$ contributes to only one subset $E'$ of parallel edges, the lemma follows by Euler's formula. ■

Since Lemma 4 also tells us how to compute the $O(n)$ planes of the proof above in $O(n \log n)$ time, we have the following corollary.

**Corollary 1** *Given a convex polyhedron $P$ with $n$ vertices, in $O(n \log n)$ time a set of candidate casting planes $\mathcal{H}^{\times}(P)$ of $O(n)$ size can be constructed such that if there is an opposite casting plane that intersects at least one edge of $P$ properly, then $\mathcal{H}^{\times}(P)$ will contain an opposite casting plane.*

We now turn our attention to casting planes that intersect no edge properly. Because of Theorem 1, we only need to consider convex polyhedra. The following lemma is the basis of an inductive argument to prove a linear bound on the number of distinct casting planes that intersect no edge of a convex polyhedron properly.

**Lemma 15** *Given a convex polyhedron $P$, there exists a vertex $v$ with constant degree such that $v$ participates in a constant number of antipodal pairs on the incident facets.*

**Proof:** Let $\tilde{V}, \tilde{E}, \tilde{F}$ be the number of vertices, edges and facets of $P$. The summed degree of all vertices $D = 2\tilde{E} \leq 6\tilde{V} - 12$. Every vertex has at least degree 3, thus there must be at least $\tilde{V}/2 + 1$ vertices of degree at most 8.

Given a convex polygon with $t$ vertices, there are at most $3t/2$ antipodal pairs [26]. Let $d(f)$ represent the degree of facet $f$. The total number of antipodal pairs, summed over all facets is at most

$$\sum_{f \in F} 3d(f)/2 \leq 3\tilde{E} \leq 9\tilde{V} - 18.$$

This implies that the total vertex contribution in antipodal pairs, $A$, satisfies $A \leq 18\tilde{V} - 36$. Observe that every vertex of $P$ participates in at least 3 antipodal pairs: at least one in each incident facet. If all $\tilde{V}/2 + 1$ vertices of degree at most 8 are in at least 33 antipodal pairs on the incident facets, then $A \geq 33(\tilde{V}/2 + 1) + 3(\tilde{V}/2 - 1) = 18\tilde{V} + 30$, a contradiction. Hence, there exists a vertex which is in at most 32 antipodal pairs and with degree at most 8. ■

Let $h$ be a candidate casting plane of $P$, and let $Q = h \cap P$. If $Q$ contains three consecutive vertices $u, v, w$ that are also vertices of $P$, then each of $u$ and $w$ is either an endpoint of an edge incident to $v$, or a vertex antipodal to $v$ on the closure of a facet $f$ incident to $v$. We say that the plane through $u, v, w$ is *generated by* $v$. It follows that the set of candidate casting planes generated by $v$ has size $\binom{d+a}{2}$, where $d$ is the degree of $v$ and $a$ is the number of vertices antipodal to $v$ in the closures of the facets incident to $v$. Every casting plane $h$ that does not intersect any edge properly contains at least three vertices that are consecutive in $h \cap P$, and therefore, every such casting plane is generated by some vertex of $P$.

**Theorem 3** *Given a convex polyhedron $P$ with $n$ vertices, the maximum number of distinct opposite casting planes for $P$ is $O(n)$.*

**Proof:** First, assume that the casting plane $h$ intersects some edge $e$ of $P$ properly. By Lemma 14, there are $O(n)$ distinct casting planes of this type.

Next, we show that the number of casting planes that do not intersect any edge properly is linear. For such a casting plane $h$, all vertices of the intersection polygon $Q = h \cap P$ are also vertices of $P$.

The proof is by induction. Let $v$ be a vertex of $P$ of degree at most 8 and which participates in at most 32 antipodal pairs (see Lemma 15). The number of casting planes containing $v$ which do not intersect any edges properly is bounded from above by the number of planes generated by $v$, and hence, is constant. We remove $v$ from $P$ and continue the count on the convex hull of the remaining vertices. We have counted all distinct casting planes that contain $v$. Since any casting plane of $P$ that does not contain $v$ and does not intersect any edge incident to $v$ properly is also a casting plane of $CH(\text{vertices of } P - v)$, the theorem follows by induction. ∎

There is another interesting combinatorial bound on the complexity of the intersection of all opposite casting planes with a convex polyhedron. Referring to the proof of Lemma 14, we notice that two distinct opposite casting planes $h_1$ and $h_2$ that intersect an edge of $E'$ properly are *similar*, because they define the same cast removal directions, and they intersect the same closure of edges and facets. In other words, if $h_1$ and $h_2$ each intersect edges properly that are parallel, there cannot be two vertices $u, v$ strictly to the one side of $h_1$ and strictly to different sides of $h_2$. We use the term *weakly equivalent* for two such planes. Two planes are *strongly distinct* if they are not weakly equivalent. There are $O(n)$ strongly distinct casting planes for any convex polyhedron $P$ with $n$ vertices. We analyze the combinatorial complexity of $h \cap P$, summed over all strongly distinct casting planes $h$. This quantity is well-defined for opposite cast removal, since two weakly equivalent casting planes have an equal-size intersection with $P$. We prove a bound of $O(n \log n)$ on the summed complexity. Note that when the sum is over all distinct casting planes (not *strongly* distinct), the summed complexity can be $\Theta(n^2)$ if $P$ has a set of $\Omega(n)$ parallel edges. The bound makes use of a hierarchical decomposition of $P$ that closely resembles the hierarchy of Dobkin and Kirkpatrick [12]. It is the basis of the $O(n \log^2 n)$ time algorithm for casting of convex polyhedra with opposite cast removal.

**Lemma 16** *Given a convex polyhedron $P$ with $n$ vertices, there exists a subset $V'$ of the vertices $V$ of size $\Omega(n)$, such that each $v \in V'$ has degree at most 8 and is antipodal to at most 52 vertices in facets incident to $v$.*

**Proof:** Similar to Lemma 15, one can prove that there are at least $\tilde{V}/5$ vertices of degree at most 8 and in at most 52 antipodal pairs ; otherwise,

$$\begin{aligned} A &\geq 53(\tilde{V}/2 - \tilde{V}/5 + 2) + 3(\tilde{V}/2 + \tilde{V}/5 - 2) \\ &= 18\tilde{V} + 100, \end{aligned}$$

a contradiction. ■

The following hierarchical decomposition of $P$ generates a set $\mathcal{H}^{\subset}(P)$ of planes that contains all the candidate casting planes that do not intersect an edge properly. The correctness follows from the proof of Theorem 3.

**Algorithm 1:** *Compute all generated planes*

1. Set $i = 1$.
2. Compute the antipodal pairs of the facets of $P$.
3. Select a subset $V_i$ of $V$ as in Lemma 16. For every vertex $v \in V_i$, generate all planes through $u, v, w$. For every vertex $v \in V_i$, the number of generated planes is at most $\binom{52+8}{2} = 1770$, thus $O(n)$ for the whole subset.
4. Recompute the convex hull of the vertices of $P$ minus the vertices of $V_i$.
5. Repeat at step 2 with $i = i + 1$ unless $P$ has no vertices left.

The number of generated planes is linear since each vertex generates a constant number. Antipodal pairs computations take $O(n)$ time and convex hull computations take $O(n \log n)$ time, see e.g. [14, 26]. The total time taken by Algorithm 1 is given by the recurrence $T(n) \leq T((1-\alpha)n) + O(n \log n)$ where $\alpha \geq 1/5$ is the constant in the $\Omega(n)$ of Lemma 16. This recurrence solves to $T(n) = O(n \log n)$. Therefore, we conclude with the following.

**Lemma 17** *Given a simple polyhedron $P$ with $n$ vertices, we can construct a set of $O(n)$ planes $\mathcal{H}^{\subset}(P)$ in $O(n \log n)$ time such that if there is an opposite casting plane that intersects no edge of $P$ properly, then $\mathcal{H}^{\subset}(P)$ will contain an opposite casting plane.*

**Theorem 4** *Given a convex polyhedron $P$ with $n$ vertices, the total complexity of $h \cap P$, summed over all strongly distinct opposite casting planes $h$ for $P$, is $O(n \log n)$.*

**Proof:** In the following proof, we make a distinction between planes that are *generated*, and other planes that can be casting planes. Planes of the second type intersect some edge properly.

Consider a hierarchical decomposition of the vertices of $P$ into sets $V_1, \ldots, V_m$ as described above. Observe that $m \in O(\log n)$.

Let $h$ be any plane, and let $v_1, \ldots, v_k$ be the sequence of vertices in $h \cap P$. We first show that every consecutive subsequence $v_i, \ldots, v_{i+2m-1}$ of vertices that also are vertices of $P$ (no proper intersections of edges of $P$ with $h$) contains a vertex that generates $h$. To this end, observe that $v_j$ generates $h$ if and only if $v_j$ is in a vertex set $V_s$ with lower or equivalent index as its neighbors, thus if $v_{j-1} \in V_r$ and $v_{j+1} \in V_t$, then $r \geq s$ and $t \geq s$. Since there are only $m$ vertex sets, any consecutive sequence of $2m$ vertices contains at least one that that generates the plane $h$.

Consider the vertices $v_i$ that are proper intersections of $h$ and an edge of $P$. Any edge $e$ gives rise to at most one strongly distinct casting plane, and therefore, the total number of these vertices in $h \cap P$, summed over all strongly distinct casting planes, is linear.

Summarizing, the sequences of $h \cap P$ summed over all strongly distinct casting planes contain $O(n)$ vertices that generate a casting plane, $O(n)$ vertices that are proper intersections of edges with a casting plane, and at most $2m - 1$ vertices in between. It follows that the total complexity of the intersections is $O(nm) = O(n \log n)$. ■

From Theorem 4 and its proof we immediately obtain a combinatorial bound on intersections of planes and convex polyhedra. The following corollary is not used for our results on casting but is interesting in its own right.

**Corollary 2** *Given a convex polyhedron $P$ with $n$ vertices, the number of planes that intersect the interior of $P$ but do not intersect any facets of $P$ is $O(n)$, and the number of edges of $P$ contained in these planes, summed over all planes, is $O(n \log n)$.*

# 4 Algorithms for opposite cast removal

In this section and the next, algorithms are presented for the computation of casting planes, and hence, determining whether a given polyhedron is castable. This section focuses on opposite cast removal.

## 4.1 A simple algorithm for simple polyhedra

Given a simple polyhedron $P$ with $n$ vertices, we compute $O(n)$ candidate casting planes and directions as follows. We first compute the $O(n)$ candidate casting directions $\Delta^{\times}(P)$ for casting planes that intersect some edge properly. This can be done in $O(n \log n)$ time (see Lemma 13). Second, we compute the candidate casting planes $\mathcal{H}^{\subset}(P)$ that intersect no edge properly in $O(n \log n)$ time by Algorithm 1 (see Lemma 17). Each candidate plane and direction can be tested in $O(n)$ time by Lemmas 3 and 5. We conclude:

**Theorem 5** *Given a simple polyhedron $P$ with $n$ vertices, one can decide in $O(n^2)$ time and linear space whether $P$ is castable when the cast parts must be removed in orthogonal or opposite directions.*

## 4.2 Walking around convex polyhedra

For convex polyhedra, the above result can be improved by considering only those facets intersecting a candidate plane or adjacent to an edge contained in the candidate plane. By Lemma 7, a candidate casting plane for a convex polyhedron is valid if and only if the linear program formed from these facets (in the same manner as the previous algorithm) is feasible. We also know, by Theorem 4, that the total number of facets that we check, for all $O(n)$ candidate casting planes, is only $O(n \log n)$. This will lead to an $O(n \log^2 n)$ time algorithm for a convex polyhedron with $n$ vertices. The candidate casting planes for a polyhedron $P$ can trivially be split into two classes, those that intersect an edge of $P$ properly and those that intersect no edge of $P$ properly. Each candidate in the second class is *generated* (in the terminology of Theorem 4) by some vertex of $P$. We thus break the problem into two parts. First, we check whether or not there is a valid casting plane that intersects some edge of $P$ properly. If there is none, we check whether or not there is a valid casting plane generated by some vertex of $P$.

### Algorithm 2: Testing for a casting plane that intersects an edge properly

We first describe the algorithm to test if there is a valid casting plane that intersects a particular edge properly.

Let $P$ be a convex polyhedron and $h$ a plane. For any face (vertex edge or facet) $f$ of $P$ intersected by $h$, let $next_h(f)$ and $prev_h(f)$ denote the next and previous face of $P$ encountered by a walk around the boundary of $P \cap h$ in an arbitrary but fixed direction.

**Lemma 18** *Let $P$ be a convex polyhedron, $h$ a plane intersecting the interior of $P$ and $\vec{d}$ be a direction not parallel to $h$. If, for every face $f$ of $P$ intersected by $h$, $next_h(f)$ and $prev_h(f)$ split the facets adjacent to $f$ into those compatible with $\vec{d}$ and those incompatible with $\vec{d}$, then $h$ is a casting plane for $P$ with casting directions $\vec{d}$ and $-\vec{d}$.*

**Proof:** Let $h^+$ ($h^-$) be the halfspace induced by $h$ containing a vector with orientation $\vec{d}$ ($-\vec{d}$) from $h$. We consider each face $f$ of $P$ that corresponds to an edge of $P \cap h$. There are two cases.

If $f$ is a facet of $P$ intersected by $h$ then $f$ is compatible with both $\vec{d}$ and $-\vec{d}$ by virtue of splitting the facets adjacent to $prev_h(f)$ into those compatible with $\vec{d}$ and those incompatible with $\vec{d}$.

Otherwise, $f$ is an edge $e$ of $P$ contained in $h$. Since $e$ splits the facets adjacent to $prev_h(e)$ into those compatible with $\vec{d}$ and those incompatible with $\vec{d}$, one facet adjacent to $e$ is compatible only with $\vec{d}$ (call it $f^+$) and one is compatible only with $-\vec{d}$ (call it $f^-$). From the convexity of $P$, if $f^+$ is in $h^-$, then so is $f^-$; but this a contradiction, since then $h$ would be a supporting plane of $P$. It follows that those facets contained in $h^+$ adjacent to an edge contained in the casting plane are compatible with $\vec{d}$ (and a symmetric condition for $h^-$ and $-\vec{d}$). By Lemma 7, $h$ is a casting plane for $P$, with casting directions $\vec{d}$ and $-\vec{d}$. ∎

Taken together, Lemma 9 and Lemma 18 suggest an algorithm to test if there is a casting plane intersecting a particular edge $e$ properly. By Observation 3, our two casting directions $\vec{d}$ and $-\vec{d}$ are parallel to $e$. We can inductively apply Lemma 9 to walk around $\partial P$ and collect all of the vertices and edges of $P$ that must be intersected by any opposite casting plane properly intersecting $e$. Let $F_i$ denote the $i$-th facet encountered by this walk, and let $q_i$ ($q_{i+1}$) denote the face of $F_i$ at which the walk enters (leaves) $F_i$. Initially, set $q_0$ to be $e$ and $F_0$ to be one of facets adjacent to $e$. At step $i$, we need to first find $q_i$ and $F_i$ and then find the $\vec{d}$-antipodal face $q_{i+1}$ to $q_i$ in $F_i$. The $\vec{d}$-antipodal face $q_{i+1}$ can be found by binary search on the normals of facets adjacent to $F_i$. To find the "next" facet $F_i$, we again apply Observation 3 to note that the outward normal of $F_i$ must be perpendicular to $\vec{d}$; thus we can find $F_i$ by binary search on the normals of facets adjacent to $q_{i-1}$ (in the case where $q_{i-1}$ is an edge, the binary search is trivial). In the case where there is no such facet, there must be an edge $e_i$ separating the facets adjacent to $q_{i+1}$ into those compatible with $\vec{d}$ and those incompatible with $\vec{d}$, and this edge must be contained in any valid casting plane properly intersecting $e$; in this case we set $q_i$ and $q_{i+1}$ to be the endpoints of $e_i$. We can then test by linear programming whether there is is a plane $h$ that intersects each $q_i$. If there is, we know that $h$ is a casting plane for $P$ by Lemma 18.

We preprocess the polyhedron for this algorithm as follows:

1. With Algorithm 1, compute a hierarchical decomposition of $P$ into $O(\log n)$ vertex sets $V_1, \ldots, V_m$, as in Theorem 4. Store with each vertex $v$ all $O(1)$ planes generated by $v$.

2. For every facet $f$, store the outward normals of the facets that are incident to an edge in the closure of $f$ in a sorted list.

3. For every vertex, store the outward normals of its incident facets in a sorted list (these are linearly ordered since they are incident to the same vertex).

These steps can be done in $O(n \log n)$ time.

There may of course be *generated* planes that intersect some edge of $P$ properly. In order to bound the overlap of effort between our two algorithms, we mark any generated triple encountered while running the first algorithm so that we need not consider them in the second. Algorithm 2 needs to be repeated once for each edge of our input polyhedron; in order to avoid testing the planes intersecting a given edge properly more than once, we mark any edge encountered during a walk as treated. Since a plane that intersects an edge $e$ properly must intersect each edge parallel to $e$, if we encounter a previously marked edge, we can stop the current walk and go on to the next unmarked edge. Each properly intersected edge is therefore considered at most twice over all invocations of Algorithm 2: once during a walk, and once as an initial face $q_0$.

**Algorithm 3: Test for generated casting planes**

We now need to test those candidate casting planes that are generated. For Algorithm 3, we carry out the additional preprocessing steps:

1. For every vertex $v$ of $P$, store the edges adjacent to $v$ in clockwise order, so that it is possible to determine by binary search for any query plane $h$ containing $v$, the facets or edges incident to $v$ that $h$ intersects.

2. For every facet $f$ of $P$, store the vertices in the closure of $f$ in clockwise order, so that it is possible to determine by binary search for any query plane $h$ which edges or vertices in the boundary of $f$ intersect $h$.

Each of these preprocessing steps can be carried out in $O(n \log n)$ time, so the total preprocessing time is $O(n \log n)$.

Conceptually, this algorithm is simpler than the previous one: for each generated triple, we trace the intersection of the plane $h$ spanned by that triple around the outside of the polyhedron, collecting all of the facets intersecting $h$ properly or adjacent to an edge contained in $h$. At each step, given a current facet $f$, and the point at which the walk entered $cl(f)$, we can find by binary search the point at which our candidate plane must exit $cl(f)$. If the exit point is on the interior of an edge $e$, then we abandon the current walk since if there were a valid plane properly intersecting $e$, we would have discovered it with Algorithm 2. If the exit point is a vertex $v$, we find by binary search on the normals of facets adjacent to $v$ what the next facet or edge intersected by $h$ must be. Similarly to above, if we encounter a triple of vertices that is already marked, we know that the current plane has already been tested, and we can abandon that particular candidate. Each time we walk across an edge $e$, we add each facet adjacent to $e$ to the appropriate one of $F^+(E_h)$ or $F^-(E_h)$ (in the notation of Section 2.3). Each time we walk across a facet, we add that facet to $F_h^\times$. If a walk returns to its starting edge, we have collected all of the facets necessary to test our candidate plane by the linear program described in Lemma 7.

**Theorem 6** *Given a convex polyhedron $P$ with $n$ vertices, one can decide in $O(n \log^2 n)$ time and linear space whether $P$ is castable when the cast parts must be removed in orthogonal or opposite directions.*

**Proof:** The above algorithms attain the claimed time bound. This can be seen as follows. The total preprocessing time is $O(n \log n)$. Let us count the total number of steps walking around the polyhedron in Algorithm 2. Since each edge is intersected properly by at most two walks, we charge the step that intersects an edge properly to that edge. Consider the steps between two proper edge intersections. We charge those before the first generated triple encountered to the previous edge properly intersected, and those after the first generated triple to the most recently encountered generated triple. From the proof of Theorem 4, we know that there are $O(n)$ triples and that every $2m \in O(\log n)$ consecutive vertices contains at least one generated triple. It follows that $O(n \log n)$ steps are charged to generated triples and edges. Since each walking step takes $O(\log n)$ time, Algorithm 2 takes $O(n \log^2 n)$ time to generate linear programs.

For Algorithm 3, the time bound follows in a similar way; each step that discovers a vertex is charged to the most recently encountered generated triple. Since each walk in Algorithm 3 intersects at most one edge properly, we charge the step that discovers this edge to the triple that started the walk. It follows that the second algorithm also takes $O(n \log n)$ steps and $O(n \log^2 n)$ time to generate linear programs.

By Theorem 4, the total complexity of all linear programs generated by both algorithms is $O(n \log n)$, hence the total time to test all candidate planes is $O(n \log n)$. ∎

## 4.3 A data structuring approach

The second, $O(n \log^2 n)$ time solution described above only applies to convex polyhedra. By using data structures, we will improve upon the quadratic time results of Theorem 5 for simple polyhedra as well. Unfortunately, the storage requirements increase with the data structuring method. The idea is to test every candidate casting plane and every candidate casting direction by querying with it in a data structure (instead of applying linear programming). The query determines whether the plane or direction really is a valid casting plane or casting direction for the polyhedron $P$. The preprocessing of the data structures should be less than quadratic and the query time should be less than linear in order to beat the quadratic time bound. The previous subsection showed how to find the $O(n)$ candidate casting planes $\mathcal{H}^{\subset}(P)$ and the $O(n)$ candidate casting directions $\Delta^{\times}(P)$ in $O(n \log n)$ time, so we only describe the data structures and the query algorithm. We first describe the data structure for testing a candidate casting plane; the data structure for testing a candidate casting direction follows later.

It turns out that the data structure for testing a candidate casting plane is exactly the same for the three versions of removing the cast; only the query algorithms are different. Let $P$ be a polyhedron. For any vertex $v \in V$, define $F(v)$ as the set of facets incident to $v$, and for any subset $V' \subseteq V$, define $F(V')$ as the set of facets incident to at least one vertex of $V'$. We make the following observation:

**Observation 6** *For any plane $h$, we have $F_h^+ \cup F_h^\times = F(V_h^+)$.*

We store $P$ in a 2-level data structure $T$. The primary tree is a 3-dimensional partition tree that stores the set $V$ of vertices of $P$, see Matoušek [20, 21] and Agarwal and Sharir [2] for example. They show that for any constant $\epsilon > 0$, a structure of size and preprocessing time $O(n^{3/2+\epsilon})$ exists, such that for any query plane $h$, the subset $V_h^+ \subseteq V$ of vertices above $h$ can be retrieved in $O(n^{1/2+\epsilon})$ canonical nodes in $O(n^{1/2+\epsilon})$ time. For any node $\delta$ of $T$, corresponding to a canonical subset $V_\delta$, the secondary structure at $\delta$ stores $V_\delta$ as follows. Recall from Subsection 2.2 that for a facet $f$, $\Psi(f)$ is the closed half-space which supports $f$ and locally does not contain $P$, and $\Psi_0(f)$ is $\Psi(f)$ translated such that the bounding plane contains the origin. For a set of facets $F$, define $\Psi_0(F)$ as the intersection of $\Psi_0(f)$, for $f \in F$. Let $B_\delta$ be the cone $\Psi_0(F(V_\delta))$ with apex at the origin. The secondary structure is simply an array or balanced binary tree that stores the facets of $B_\delta$ in cyclic order around the apex. The secondary structure allows for queries with a half-line starting at the origin, to determine whether the half-line is contained in $B_\delta$. This query is in fact a 2-dimensional query to determine whether a point lies in a convex polygon.

Suppose that we wish to determine whether $h$ is a valid casting plane for orthogonal cast removal. Then we search with $h$ in $T$ and determine the canonical nodes of $T$ with respect to $h$, in particular, the set $\Delta^+ = \{\delta_1, \ldots, \delta_t\}$ of nodes of which the associated sets $V_{\delta_1}, \ldots, V_{\delta_t}$ are a partition of $V_h^+$. Let $\ell_h^+$ be the upward half-line normal to $h$ starting at the origin. For each of the nodes $\delta_i$, we query with $\ell_h^+$ in the secondary structure to determine by binary search whether $\ell_h^+ \in B_{\delta_i}$. If the answer is positive for all nodes $\delta_1, \ldots, \delta_t$, then $P \cap cl(h^+)$ is a terrain with respect to the direction normal to $h$ (and parallel to $\ell_h^+$). The query is repeated for $h^-$, to determine whether $P \cap cl(h^-)$ is a terrain with respect to the direction normal to $h$. If this is also the case, then $h$ is a casting plane of $P$ for orthogonal cast removal. Since there are $O(n^{1/2+\epsilon})$ cones (canonical nodes), the query time is $O(n^{1/2+\epsilon} \log n)$.

Next we consider cast removal in opposite directions. The query is a variation to the previous solution. We determine both sets $\Delta^+$ and $\Delta^-$ of canonical nodes for the queries with $h^+$ and $h^-$, respectively. Let $\Delta^+ = \{\delta_1, \ldots, \delta_t\}$ and $\Delta^- = \{\delta'_1, \ldots, \delta'_s\}$. We wish to determine whether the common intersection of the $t$ cones $B_{\delta_1}, \ldots, B_{\delta_t}$, intersected with the reflection through the origin of the common intersection of the $s$ cones $B_{\delta'_1}, \ldots, B_{\delta'_s}$, is non-trivial, see Observation 5. One can decide whether the common intersection is non-trivial using the algorithm of Reichling [29]. He shows how to find an extremal point in the common intersection of $k$ convex $n$-gons in $O(k \log^2 n)$ time. In our case, we 'reflect' all operations on the second set of cones. (Alternatively, we could store both the normal and the reflected cones explicitly at every node and choose the appropriate set, but this is not necessary.) The query time is $O((s+t)\log^2 n) = O(n^{1/2+\epsilon} \log^2 n)$ time.

Thirdly, we consider the query in the same structure to solve arbitrary cast removal. We remark that for arbitrary cast removal, we can determine using $O(n \log n)$ queries whether a casting plane exists, even though we do not have a subquadratic bound on the number of casting planes in this case. This will be shown in the next section.

Let $h$ be the plane of which we wish to determine whether it is a casting plane. We determine the set $\Delta^+$ as before; let $B_{\delta_1}, \ldots, B_{\delta_t}$ be the cones that are stored at the set $\Delta^+$ canonical nodes. Now we have to determine whether the common intersection of these cones is non-trivial. Any half-line starting at the origin and in the common intersection of the cones represents a direction with respect to which $P \cap cl(h^+)$ is a terrain. A half-line in the common intersection of the $t$ cones can be determined in $O(t \log^2 n) = O(n^{1/2+\epsilon} \log^2 n)$ time using Reichling's algorithm. If no such half-line exists, then $P \cap cl(h^+)$ is not a terrain with respect to any direction. If $P \cap cl(h^+)$ is a terrain for some direction, we repeat the query to test whether $P \cap cl(h^-)$ is a terrain.

**Lemma 19** *For any constant $\epsilon > 0$ and any simple polyhedron $P$ with $n$ vertices, a data structure of size and preprocessing time $O(n^{3/2+\epsilon})$ exists, such that for any query plane $h$, one can determine in $O(n^{1/2+\epsilon})$ time whether $h$ is a casting plane for $P$ in any of the versions for removing the cast.*

**Remark:** In fact, we have also shown that for any query half-space $h^+$, we can determine within the same bounds whether $P \cap h^+$ is a terrain in some direction. Furthermore, by choosing a different partition tree for the primary tree, we can trade query time for storage space, see e.g. Chazelle, Sharir and Welzl [8], Matoušek [21] and Agarwal and Sharir [2]. For any $n \leq N \leq n^3$, a data structure of size and preprocessing time $O(N^{1+\epsilon})$ exists with query time $O(n^{1+\epsilon}/N^{1/3})$. The theorem above states the version we need for the casting problem.

**Remark:** de Berg [11] noted that the result for orthogonal cast removal can be improved to $O(n^{4/3+\epsilon})$ time and space. Conceptually, the improved data structure reverses the tests done in the main and the secondary structure to determine castability, when compared to the previously described solution. Use a two-level data structure of which the main tree is a 2-dimensional partition tree on the planes bounding $\Psi_0(F)$. Since these planes all pass through the origin, a 2-dimensional partition tree is indeed sufficient. It allows one to select all planes below and all planes above a given query ray (normal to the query plane) starting at the origin in $O(n^{1/3+\epsilon})$ canonical subsets [2, 8, 21]. For any canonical subset of a node $\delta$, the vertices in the closures of the facets $f$ for which the half-space $\Psi_0(f)$ appears in that canonical subset are further preprocessed into a secondary data structure with $\delta$ for half-space emptyness queries, see for example Clarkson and Shor [9]. The whole data structure requires $O(n^{4/3+\epsilon})$ space and preprocessing time. This improvement only applies to orthogonal cast removal, because the cast removal direction must be known in advance.

Next we turn to testing candidate casting directions. The ideas behind the data structure for these queries are similar to the ones for testing planes. It turns out that we only need consider the case of orthogonal and opposite cast removal. For arbitrary cast removal—to be solved in the next section—we won't need to test a pair of candidate casting directions. Only the case of opposite cast removal is explained below.

Let $P$ be a simple polyhedron. Following the proof of Lemma 5 we must test for a given pair of opposite candidate casting directions $\vec{d}$ and $-\vec{d}$ whether a plane exists that separates the facets of $P$ in a compatible way. Recall from Subsection 2.1 that a facet can be represented by a point on the sphere of directions, such that $\vec{d}$ and $-\vec{d}$ determine two hemispheres that specify which facets should translate in direction $\vec{d}$ and which in direction $-\vec{d}$. Recall that if a facet $f$ is compatible only with $\vec{d}$, then all vertices of $P$ incident to $f$ must lie above or in the (unknown) casting plane.

We store $P$ in a 2-level data structure $T'$ as follows. The primary tree is a 2-dimensional partition tree of close to linear size that stores the set of points on the sphere of directions that correspond to the facets of $P$. For any great circle on the sphere of directions, the points in each of the hemispheres can be retrieved in $O(n^{1/2+\epsilon})$ canonical nodes in $O(n^{1/2+\epsilon})$ time. For any node $\delta$ of $T'$, corresponding to the canonical subset $F_\delta$, the secondary structure stores all vertices $V(F_\delta)$ incident to the facets in $F_\delta$. We dualize these vertices to a set of positive closed half-spaces and store the common intersection, which is a convex polytope. We also store the common intersection of the negative closed half-spaces. Both polytopes are stored using the drum representation of Dobkin and Kirkpatrick [13].

For a candidate query direction pair $\vec{d}$ and $-\vec{d}$, we take the great circle on the sphere of directions that separates the directions compatible with $\vec{d}$ and with $-\vec{d}$. We use this great circle to search in the primary structure, the 2-dimensional partition tree, to obtain the canonical nodes of the query. Let $\Delta^+$ and $\Delta^-$ be the sets of canonical nodes representing groups of facets that should translate in directions $\vec{d}$ and $-\vec{d}$, respectively. For the former set we take the associated polytopes formed by the positive closed half-spaces and for the latter set we take the polytopes formed by the negative closed half-spaces. Then we compute the common intersection of all these $O(n^{1/2+\epsilon})$ polytopes. If it is empty, the candidate casting directions $\vec{d}$ and $-\vec{d}$ are rejected. If there is a point in the common intersection, it dualizes back to a casting plane for which the directions $\vec{d}$ and $-\vec{d}$ are valid casting directions.

An algorithm of Reichling solves the common intersection of convex polytopes in 3 dimensions using the drum representation in $O(k \log k \log^3 n)$ time, for $k$ polytopes with $n$ facets in total [30].

**Lemma 20** *For any constant $\epsilon > 0$ and any simple polyhedron $P$ with $n$ vertices, a data structure of size and preprocessing time $O(n^{1+\epsilon})$ exists, such that for any pair of query directions $\vec{d}$ and $-\vec{d}$, one can determine in $O(n^{1/2+\epsilon})$ time whether a casting plane $h$ exists for $P$ when the cast parts must be removed in directions $\vec{d}$ and $-\vec{d}$.*

We can now conclude this section with the improved results we have obtained on determining the castability for opposite cast removal.

**Theorem 7** *For any constant $\epsilon > 0$ and any simple polyhedron $P$ with $n$ vertices, one can determine in $O(n^{3/2+\epsilon})$ time and space whether $P$ is castable when the cast parts must be removed orthogonal to the casting plane, or in opposite directions.*

**Proof:** Generate the $O(n)$ candidate casting planes and $O(n)$ candidate casting directions. Construct the data structures of Lemmas 19 and 20, and query with the candidate casting planes and directions. The result follows. ■

# 5 Algorithms for arbitrary cast removal

In this section we study the most general casting problem of this paper: determine whether a simple polyhedron $P$ is castable when the cast parts may be removed in arbitrary directions. Using results of the previous sections, we can obtain an $O(n^{9/4+\epsilon})$ time algorithm: we devise a data structure as in the previous section to test all candidate $O(n^2)$ casting planes. We will do better in this section. Using Lemmas 1 and 2 and one more observation on arbitrary cast removal, we first obtain a simple $O(n^2 \log n)$ time and linear space algorithm, and then a more complicated $O(n^{3/2+\epsilon})$ time and space algorithm.

Let $P$ be a polyhedron. We first test whether $P$ admits opposite cast removal using the simple $O(n^2)$ time algorithm of Theorem 5. If so, we are done. Otherwise, if $P$ is convex, then, by Lemma 1, we only have to consider casting planes that contain some edge of $P$. If $P$ is non-convex, then, by Lemma 2, we only have to consider casting planes that contain an edge of the convex hull of $P$. Recall from Section 2.2 that each plane $h$ defines two linear programs $\xi^+(h)$ and $\xi^-(h)$ relative to any polyhedron $P$. Also recall that a linear program is trivial if its set of feasible solutions contains only one point.

**Observation 7** *Let $P$ be a polyhedron and $h$ be a plane that contains an edge $e$ of the convex hull of $P$. Assume without loss of generality that $e$ is horizontal and that a vertical plane exists which supports $e$ and has $P - cl(e)$ completely to the one side. If the linear program $\xi^+(h)$ is trivial then no plane $\mu$ containing $e$ for which $P \cap h^+ \subset \mu^+$ is a casting plane for $P$. Similarly, if $\xi^-(h)$ is trivial then no plane $\mu$ containing $e$ for which $P \cap h^- \subset \mu^-$ is a casting plane for $P$.*

The observation follows because the linear program $\xi^+(\mu)$ contains all constraints of the linear program $\xi^+(h)$, and possibly more, if $P \cap h^+ \subset \mu^+$. The above observation sets up a binary search for a casting plane that contains some edge $e$ of the convex hull of $P$. First, compute the convex hull of $P$. For each edge $e$ of the convex hull, rotate $P$ such that $e$ is as in the observation. Consider the $n-2$ vertices that are not endpoints of $e$, and sort them by the order in which a vertical plane supporting $e$ encounters them if the plane starts rotating about $e$ (The plane $h$ can rotate in two directions about $e$; it is not important which direction is chosen, as long as this choice is made consistently). Assume without loss of generality that the order is $v_1, \ldots, v_{n-2}$. We test whether the plane $h$ supporting $e$ and also containing $v_{n/2-1}$ is a casting plane by determining whether $\xi^+(h)$ is non-trivial and $\xi^-(h)$ is non-trivial. By the above observation, we can stop considering $e$ if both are trivial. If both are non-trivial, we can also stop and $h$ is a casting plane. Otherwise, if only $\xi^+(h)$ is non-trivial, we continue the binary search on $v_{n/2}, \ldots, v_{n-2}$, and if only $\xi^-(h)$ is non-trivial, we continue the binary search on $v_1, \ldots, v_{n/2-2}$. After at most $\lceil \log_2(n-2) \rceil$ steps, we have determined whether there exists a casting plane that contains $e$. This leads to:

**Theorem 8** *Given a simple polyhedron $P$ with $n$ vertices, one can determine in $O(n^2 \log n)$ time and linear space whether a casting plane for $P$ exists, when the cast parts can be removed in arbitrary directions.*

**Proof:** To decide whether opposite cast removal is possible we first apply Theorem 5 and use $O(n^2)$ time. The computation of the convex hull of $P$ requires $O(n \log n)$ time. There are $O(n)$ edges about which a plane is rotated. The sorting of the vertices $v_1, \ldots, v_{n-2}$ takes $O(n \log n)$ time, and each step of the binary search takes linear time by Lemma 3. Hence, the above procedure takes $O(n^2 \log n)$ time. ■

As in Subsection 4.3, we can preprocess $P$ into a data structure such that any casting plane can be tested in $O(n^{1/2+\epsilon})$ time. Since $O(n \log n)$ casting planes are tested by the above procedure, the test part can be improved to $O(n^{3/2+\epsilon})$ time. However, how can we obtain the order of $v_1, \ldots, v_{n-2}$ without sorting? Again, the solution lies in data structuring. Notice first that the order of $v_1, \ldots, v_{n-2}$ is not needed explicitly. In the first step, the vertex that is the median $v_{n/2-1}$ must be determined, and in the following steps a median in one of the two halves.

The data structure that is needed preprocesses $P$ for the following query problem: Given a query edge $e$ such that there exists a plane $h$ containing $e$ that has the interior of $P$ completely to the one side, and an integer $k$, find the $k$-th vertex of $P$ that is encountered by $h$ when it rotates about $e$ (see Figure 4). Dualization yields a more familiar query problem: preprocess a set of $n$ planes (dual to the vertices of $P$), such that for any given query ray, the $k$-th intersection point with the planes can be determined. The query ray is contained in the line dual to the line supporting $e$, and the starting point of the ray is any point dual to a plane containing $e$ that has the interior of $P$ completely to the one side.

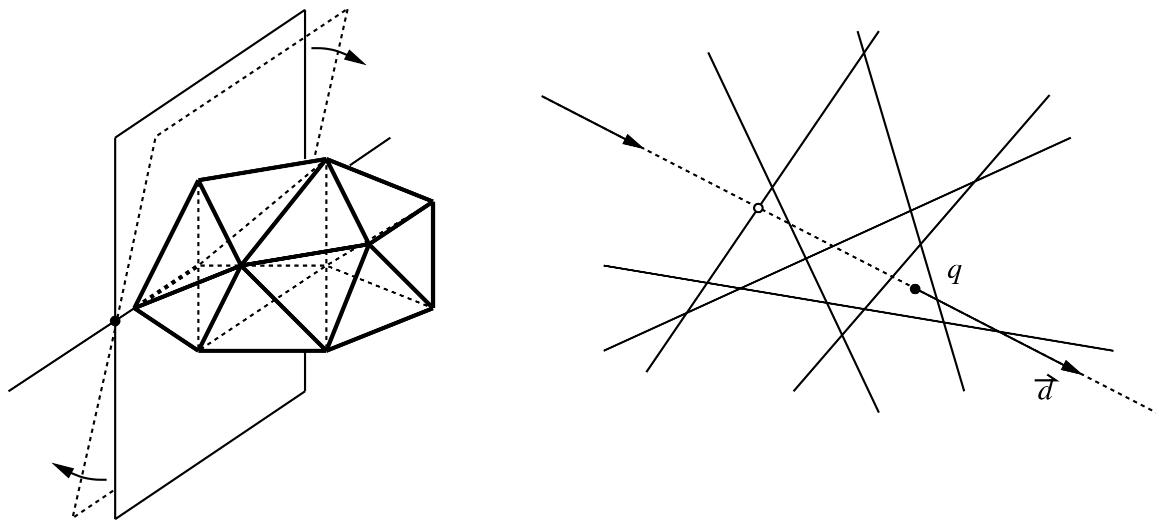


Figure 4: Left, rotating a plane about an edge through $P$. Right, the dual problem (in 2 dimensions), where the third plane intersecting the ray $q + \lambda \cdot \vec{d}$ is found after wrap-around.

Let $\mathcal{D}(V) = \{\mathcal{D}(v) \mid v \in V\}$ be the set of planes dual to the set $V$ of vertices of $P$, preprocess them into a data structure for line segment intersection counting, as given by Agarwal and Matoušek [1]. They show that for any $\epsilon > 0$, a structure of size and preprocessing time $O(n^{3/2+\epsilon})$ exists, such that segment intersection counting queries can be answered in $O(n^{1/2+\epsilon})$ time. Furthermore, they show how the same structure can be used to find the $k$-th intersection point of a query ray with $\mathcal{D}(V)$ in $O(n^{1/2+\epsilon})$ time. Let the query ray be parameterized by $q + \lambda \cdot \vec{d}$, $\lambda \geq 0$, where $q$ is a point and $\vec{d}$ is a vector in 3-dimensional space. In our application, if the $k$-th intersection point does not exist, and the last intersection point is the $j$-th, then the query should be continued in 'wrap-around' mode: find the $(k - j)$-th plane for the query ray along the same line and in the same direction, but with $q$ translated in direction $-\vec{d}$ to infinity, see Figure 4. When this happens, the plane rotating about $e$ in primal space rotates past a vertical orientation. These adaptations to the query algorithm can easily be made within the same asymptotic time bounds. Hence, we conclude:

**Theorem 9** *For any simple polyhedron $P$ with $n$ vertices and any constant $\epsilon > 0$, one can determine in $O(n^{3/2+\epsilon})$ time and space whether $P$ is castable, when the cast parts can be* *removed in arbitrary directions.*

# 6 Conclusions and open problems

This paper studied the geometric version of the problem of determining whether a simple polyhedron can be manufactured using casting. It was assumed that there are two cast parts, and each has to be removed with a single translation. We presented simple algorithms that use $O(n^2)$ or $O(n^2 \log n)$ time and linear space which are based on linear programming. Furthermore, we showed that theoretically, better results can be obtained using partition trees and their variants. This leads to an $O(n^{3/2+\epsilon})$ time and space algorithm. Using the partition trees of Matoušek [21], the bound is in fact $O(n^{3/2} \operatorname{polylog}(n))$. Finally, as a by-product, we obtained a combinatorial bound on the number of planes intersecting a polyhedron in edges only, and on the summed number of edges in these planes.

Manufacturing applications have not been studied much in computational geometry. There are quite a large number of open problems to be solved in this area. We first list some open problems related to this paper, and then list a few others in cast design that deserve attention.

1. What is the maximum number of distinct casting planes in case of arbitrary cast removal? This paper shows $O(n^2)$, whereas the only lower bound we have is linear.
2. For a convex polyhedron $P$, what is the maximum summed complexity of the intersection of all distinct casting planes with $P$? This paper shows $O(n \log n)$ in case of opposite cast removal, but the trivial lower bound is linear.
3. Give simple algorithms for casting that improve our simple $O(n \log^2 n)$, $O(n^2)$ and $O(n^2 \log n)$ time algorithms. Find algorithms that improve upon our $O(n^{3/2+\epsilon})$ time algorithms.
4. Suppose that the casts may be removed with any motion. Give algorithms to determine whether a polyhedron is castable in this case.
5. Suppose that we wish to determine castability of an object with non-linear boundaries. Give (simple) algorithms that solve this problem.
6. Suppose that more cast parts are allowed. Determine for a polyhedron how many cast parts are necessary.
7. For some casting processes, it is not necessary that the cast parts must be separated by a plane. In these cases, every convex polyhedron is castable. However, no algorithms are known for cast removal of simple polyhedra using other separating surfaces.

**Acknowledgements.** This research was initiated during the workshop on 'Injection molding' at the Bellairs Research Institute, McGill University. The authors thank the organizer Godfried Toussaint and the other participants for creating a good research environment. Furthermore, we thank Pankaj Agarwal for answering several of our questions. Finally, we thank both referees for their many helpful comments.

# A Convex polyhedra that are not castable

| | | |
|---|---|---|
| -5871633 | 7315477 | 3465226 |
| 2494994 | 9546066 | -1627170 |
| 4897569 | 8706790 | 453478 |
| 6395355 | -5103364 | -5749360 |
| -6661667 | 5838315 | -4640712 |
| -5482869 | 4061384 | -7310492 |
| -1633960 | -6912910 | 7038598 |
| -6973801 | 5021136 | -5114120 |
| 2904199 | 6316631 | 7187890 |
| -1042712 | -9847838 | 1390283 |
| -3447429 | 7711357 | -5352589 |
| 3384797 | 376088 | -9402217 |

Table 3: A set of 12 points whose convex hull is not castable.

According to an exhaustive checking of all possible planes through three vertices, the convex hull of the set of points given in Table 3 is not castable. The points were generated at random, near the surface of a sphere. By Observation 2, it suffices to consider the $O(n^3)$ casting planes through triples of vertices; each of these planes was considered in turn and the corresponding linear programs constructed as in Section 2.2. All computations were done in exact rational arithmetic, using Maple. A simple Maple program to check convex polyhedra for castability by brute force is available from the second author.

# References


[1] Agarwal, P. K., and J. Matoušek, Ray shooting and parametric search. *SIAM J. Computing* **22** (1993), pp. 794–806.

[2] Agarwal, P. K., and M. Sharir, Applications of a new space partitioning technique. *Discr. & Comp. Geometry* **9** (1993), pp. 11–38.

[3] Asberg, B., G. Blanco, P. Bose, J. Garcia, M. Overmars, G. Toussaint, G. Wilfong, and B. Zhu, Feasibility of design in stereolithography. Appears in this issue. Also appears in *Proc. 13th Symp. on FST & TCS*, Bombay, India, 1993.

[4] Bose, P., *Geometric and Computational Aspects of Manufacturing Processes*, Ph.D. Thesis, School of Computer Science, McGill University, December 1994, also available as U.B.C. Technical Report 95-02, Dept. of Comp. Sci., University of British Columbia, January 1995.

[5] Bose, P., M. van Kreveld, and G. Toussaint, Filling polyhedral molds. *Proc. 3rd WADS* (1993), Lect. Notes in Comp. Science 709, Springer-Verlag, pp. 210–221.

[6] Bose, P., and G. Toussaint, Geometric and computational aspects of injection molding. *Proc. 3rd Int. Conf. on CAD and Computer Graphics* (1993), Beijing, China, pp. 237–242. Also: Tech. Rep. No. SOCS 92.16, School of Computer Science, McGill University, 1992.

[7] Bose, P., and G. Toussaint, Geometric and computational aspects of manufacturing processes - a survey, *Computers and Graphics*, 18(4), (1994), pp. 487–497.

[8] Chazelle, B., M. Sharir, and E. Welzl, Quasi-optimal upper bounds for simplex range searching and new zone theorems. *Algorithmica* **8** (1992), pp. 407–429.

[9] Clarkson, K. L., and P. W. Shor, Applications of random sampling in computational geometry, II. *Discr. & Comp. Geometry* **4** (1989), pp. 387–421.

[10] Personal Communication with R. Connolly, Manager Injection Molding, Industrial Materials Institute, National Research Council of Canada, Boucherville, Quebec, November 1993.

[11] Personal Communication with M. De Berg, Utrecht University, 1994

[12] Dobkin, D. P., and D. G. Kirkpatrick, A linear time algorithm for determining the separation of convex polyhedra. *J. Algorithms* **6** (1985), pp. 381–392.

[13] Dobkin, D.P., and D.G. Kirkpatrick, Fast detection of polyhedral intersection. *Theor. Comp. Science* **27** (1983), pp. 241–253.

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

[15] Elliott, R., *Cast Iron Technology.* Butterworths, London, 1988.

[16] Fekete, S.P., and J.S.B. Mitchell, *Geometric aspects of injection molding,* manuscript, 1993.

[17] Held, M., *On the Computational Geometry of Pocket Machining.* Lect. Notes in Comp. Science 500, Springer-Verlag, 1991.

[18] Houle, M. E. *Weak Separation of Sets.* PhD thesis, McGill University, 1989.

[19] Isayev, A.I. (Ed.), *Injection and Compression Molding Fundamentals.* Marcel Dekker, Inc., New York, 1987.

[20] Matoušek, J., Efficient partition trees. *Discr. & Comp. Geometry* **8** (1992), pp. 315–334.

[21] Matoušek, J., Range searching with efficient hierarchical cuttings. *Discr. & Comp. Geometry* **10** (1993), pp. 157–182.

[22] McAllister, M., and J. Snoeyink, Two dimensional computation of the three dimensional reachable region for a welding head. *Proc. 5th Canadian Conf. on Comp. Geom* (1993), pp. 437-442.

[23] Megiddo, N., Linear-time algorithms for linear programming in $R^3$ and related problems. *SIAM J. Comp.* **12** (1983), pp. 759–776.

[24] Megiddo, N., Linear programming in linear time when the dimension is fixed. *J. ACM* **31** (1984), pp. 114–127.

[25] O'Rourke, J., *Computational Geometry in C*, Cambridge University Press, New York, 1994.

[26] Preparata, F.P., and M.I. Shamos, *Computational Geometry – an introduction.* Springer-Verlag, New York, 1985.

[27] Pribble, W.I., Molds for reaction injection, structural foam and expandable styrene molding. In: *Plastics Mold Engineering Handbook*, J.H. DuBois and W.I. Pribble (Eds.), Van Nostrand Reinhold Company Inc., New York, 1987.

[28] Rappaport, D., and A. Rosenbloom, Moldable and castable polygons. *Proc. 4th Canadian Conf. on Comp. Geom.* (1992), pp. 322-327.

[29] Reichling, M., On the detection of a common intersection of $k$ convex objects in the plane. *Inf. Proc. Lett.* **29** (1988), pp. 25–29.

[30] Reichling, M., On the detection of a common intersection of $k$ convex polyhedra. In: *Computational Geometry and its Applications.* Lect. Notes in Comp. Science 333, Springer-Verlag, 1988, pp. 180–186.

[31] Seidel, R., Small-dimensional linear programming and convex hulls made easy. *Discr. & Comp. Geometry* **6** (1991), pp. 423–434.

[32] Walton, C.F., and T.J. Opar (Eds.), *Iron Castings Handbook.* Iron casting society, Inc., 1981.

[33] Whelan, A., *Injection Moulding Machines.* Elsevier, London, 1984.

[34] C.-K. Yap. Symbolic treatment of geometric dependancies. *J. Symbolic Computation*, **10** (1990), pp. 349–370.