Determining the Castability of Simple Polyhedra*

Prosenjit Bose
McGill University

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. Such polyhedra can be manufactured easily using two cast parts. Assuming that the cast parts are removed by a single translation each, 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. Furthermore, 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 new application area of computational geometry is 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 injection molding [5, 6, 12], NC machining [13], automated welding [16] and layer deposition methods (such as stereolithography) [3].

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 a liquid metal, after which the cast parts are removed. The removal of the cast parts without breaking them imposes certain restrictions on the shape of the object to be constructed. For *sand casting* (see e.g. [11, 22]), 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 face along which the part is cut is referred to as the base. The first cast part is made by placing the base of the part on a flat surface, and then adding sand around it. The part is then rotated such that the base is facing up. The other part is placed such that the bases coincide and the second cast part is built by adding sand around this part. An opening into the cavity is maintained during the construction of the second cast part. This completes the construction of the cast of the prototype object. To built a metal rendition of the prototype object with this cast, liquid metal is poured into the opening until it fills the cavity. After the metal solidifies, the cast parts are removed from the object. To be able to construct the cast with this process, it is necessary that the prototype can be removed without breaking the cast. Also, for manufacturing methods related to sand casting such as other metal casting methods [11, 22] and injection molding and blow molding methods for plastics [19, 23], it must be possible to remove the constructed object from the cast without breaking it in order to re-use the cast. 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 [11, 19, 22, 23]. 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.

An object is *castable* if it can be manufactured by casting. Geometric and algorithmic issues of casting of planar objects has been studied by Rosenbloom and Rappaport [20]. This paper addresses casting of polyhedral objects. In geometric terms, castability can be defined as follows (for a polyhedron P, ∂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 in some orientation, and $h^- \cap \partial P$ is a weak terrain in some*

*This research is supported by the NSERC and the ESPRIT Basic Research Action 7141 (project ALCOM II: *Algorithms and Complexity*).

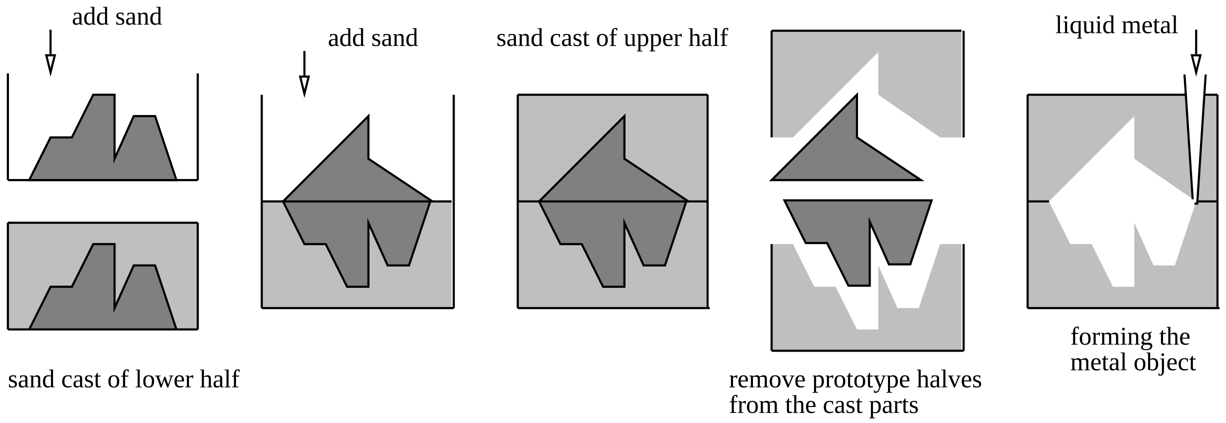

Figure 1: Construction of an object by sand casting, using two halves of the object as prototypes.

orientation. The plane h is called the casting plane. *(A weak terrain may contain edges and facets parallel to the orientation in which it is a terrain.)*

To manufacture a polyhedron P that is castable, first determine a casting plane h for P. Then the cast parts C_1 and C_2 are made from the prototype halves $h^+ \cap \partial P$ and $h^- \cap \partial P$. 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. Although our algorithms deal with the sand casting process, they can be applied in related processes as well. We consider three versions of the castability problem. They differ in the way the cast may be removed from the polyhedron P. Figure 2 shows the three versions for planar polygons.

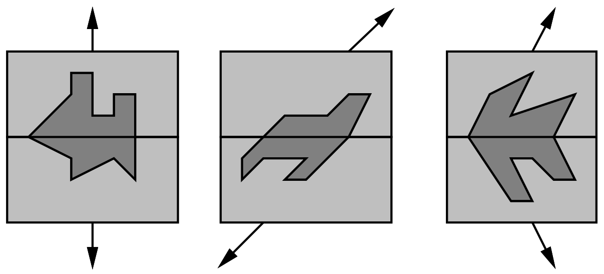

Figure 2: Three versions of the castability problem.

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).

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

In manufacturing, developing machines that perform orthogonal and opposite cast removal is much simpler than machines that perform arbitrary cast removal. In fact, opposite cast removal seems to be the most popular technique used [8, 19]. Furthermore, if orthogonal or opposite cast removal is possible, it can be determined more efficiently. We summarize the complexity of the different algorithms we developed for the casting problem. 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 ϵ).

lin. space	orthogonal	opposite	arbitrary
convex	$O(n \log^2 n)$	$O(n \log^2 n)$	$O(n^2 \log n)$
simple	$O(n^2)$	$O(n^2)$	$O(n^2 \log n)$
best res.	**orthogonal**	**opposite**	**arbitrary**
convex			$O(n^{3/2+\epsilon})$
simple	$O(n^{3/2+\epsilon})$	$O(n^{3/2+\epsilon})$	$O(n^{3/2+\epsilon})$

Our algorithms depend on geometric properties of casting, discussed in Section 2. The maximum number of distinct cast planes is analysed in Section 3, and algorithms are given in Sections 4 and 5. For detailed proofs and descriptions of the algorithms, we refer the reader to the technical report [4].

2 Preliminaries

For a polyhedron P, denote by V, E and F the set of 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 edge e or facet f is denoted by $cl(e)$ or $cl(f)$. 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.

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.

Let P be a polyhedron and let h be a plane that intersects P. If h is a casting plane and the upper cast part can be removed in direction $\vec{u}$, then the outward normal of any facet f of P that lies in h^+ makes an angle at most $\pi/2$ with $\vec{u}$. This also holds when f lies partially in h^+. Also, if all facets of P that lie completely or partially in h^+ make an angle at most $\pi/2$ with $\vec{u}$, then $h^+ \cap \partial P$ is a weak terrain. 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 case of orthogonal cast removal, the only perturbation allowed is translation.

Observation 1 *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.*

2.1 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 $E_h^{\subset}$, $E_h^{\times}$, 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 supporting f but not containing the interior of P. Denote by $\Psi_0(f)$ the same half-space, but translated such that the bounding plane contains the origin. Let

$$\xi^+(h) = \bigcap_{f \in F_h^+ \cup F_h^{\times}} \Psi_0(f) \quad \text{and} \quad \xi^-(h) = \bigcap_{f \in F_h^- \cup F_h^{\times}} \Psi_0(f) .$$

The intersection of a set of half-spaces is *non-trivial* if it contains more than a single point. Denote by $\mathit{refl}(b)$ the reflection of an object b through the origin. We make the following observations.

Lemma 1 *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.*

Lemma 2 *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.*

Lemma 3 *Let h be a plane and let ℓ be a line perpendicular to h and through the origin. h is a casting plane for polyhedron P for orthogonal cast removal if and only if $\ell \cap \xi^+(h) \cap \mathit{refl}(\xi^-(h))$ is non-trivial.*

With the above lemmas, we can test for a given plane h efficiently whether it is a casting plane for P. 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 [17]. In fact, since all planes bounding the half-spaces contain the origin, the linear programming problem is basically 2-dimensional.

Lemma 4 *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.*

2.2 Geometric properties of casting

In this section, we uncover some important geometric properties of casting planes that will allow us to bound the number of distinct casting planes and hence develop efficient algorithms.

Lemma 5 *If a simple polyhedron P is castable in non-opposite directions with casting plane h, then h contains an edge of P.*

Lemma 6 *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.*

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.

The following lemma forms the crucial link between simple polyhedra and convex polyhedra in terms of castability with removal directions restricted to orthogonal and opposite.

Lemma 7 *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.*

Lemma 8 *If the casting plane h properly intersects a facet f of convex polyheron P, and also two vertices u, v in the closure of f, then for opposite cast removal, vertices u, v must be antipodal in $cl(f)$.*

Corollary 1 *Let h be a casting plane for a convex polyhedron P which intersects a facet f properly, and assume opposite cast removal. If h intersects a vertex v and properly intersects an edge e in the closure of f, then v is antipodal to both endpoints of e. If h properly intersects two edges in the closure of f, then they are parallel.*

Lemma 8 and Corollary 1 are the two geometric facts that allow us to solve opposite cast removal more efficiently than arbitrary cast removal.

In [4], we also show that for a convex polyhedron P, 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, leading to the algorithm described in Section 4.

3 The number of distinct casting planes

Given a polyhedron P, 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. By Observation 1, a trivial upper bound on the number of distinct casting planes for a polyhedron with n vertices is $O(n^3)$.

This section gives a linear upper bound on the maximum number of distinct casting planes for convex polyhedra in case of orthogonal and opposite cast removal as well as a quadratic upper bound for arbitrary cast removal.

3.1 Orthogonal and opposite cast removal

We restrict our attention to opposite cast removal, since orthogonal cast removal is (close to) a special case of opposite cast removal, and there are only minor differences between the two cases. For an explicit rendition of the differences, we refer the reader to [4]. For opposite cast removal, we have the following properties of intersections of a casting plane h and a polyhedron P: (i) The facets of $F^{\times}$ that intersect h properly have their outward normals such that when translated to the origin, they span a plane or part of it (since $\bigcap\{\Psi_0(f) \mid f \in F^{\times}\}$ contains a line through o). (ii) All edges that intersect h properly are parallel (otherwise the incident facets span more than a plane). (iii) 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 8).

Lemma 9 *Given a convex polyhedron P, the number of distinct casting planes that intersect some edge of P properly is at most linear in the number of vertices of P, assuming opposite cast removal.*

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 three vertices that are consecutive in $h \cap P$, and therefore, every such casting plane is generated by some vertex of P. It is straightforward to prove that every convex polyhedron has a vertex of constant degree which participates in a constant number of antipodal pairs (on the incident facets). Using this fact inductively, we obtain:

Theorem 1 *Given a convex polyhedron P with n vertices, the maximum number of distinct casting planes for P is $O(n)$, assuming opposite removal of the cast parts.*

There is another interesting combinatorial bound on the complexity of the intersection of all distinct casting planes with a convex polyhedron. Two distinct casting planes h_1 and h_2 that properly intersect the same edge 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 properly intersect edges 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 *weakly distinct* if they are not weakly equivalent. There are $O(n)$ weakly distinct casting planes for any convex polyhedron P with n vertices. We analyse the combinatorial complexity of $h \cap P$, summed over all weakly 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 (although they may intersect different facets, edges and vertices). 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 *weakly* 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 [9]. It is the basis of the $O(n \log^2 n)$ time algorithm for casting of convex polyhedra with opposite cast removal.

Lemma 10 *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 ≤ 8 and is antipodal to ≤ 12 vertices in facets incident to v.*

The following hierarchical decomposition of P generates a set of planes that contains all the candidate casting planes that do not intersect an edge properly:

1. Mark all edges of P as "original". Compute all antipodal pairs of all facets of P and mark them as "original". Set $i = 1$.
2. Select a subset V_i of V as in Lemma 10. For every vertex $v \in V_i$, generate all planes through u, v, w for which $\overline{uv}$ and $\overline{vw}$ are original edges, or (u, v) and (v, w) are original antipodal pairs, or one is an original edge and the other an original antipodal pair. For every vertex $v \in V_i$, the number of generated planes is at most $\binom{12+8}{2} = 190$, thus $O(n)$ for the whole subset.
3. Recompute the convex hull of the vertices of P minus the vertices of V_i.
4. Recompute the antipodal pairs of the facets. (Note: Edges or antipodal pairs of $CH(P \setminus V_i)$ are not original unless they were deemed original in step 1.)
5. Repeat at step 2 with $i = i + 1$ unless P has no vertices left.

As in [9], the number of subsets V_i is $O(\log n)$. It follows—as we show in [4]—that every $\Omega(\log n)$ consecutive vertices of P contained in a casting plane h contains a vertex that generates h. This leads to:

Theorem 2 *Given a convex polyhedron P with n vertices, the maximum total complexity of $h \cap P$, summed over all weakly distinct casting planes h for P, is $O(n \log n)$, assuming opposite cast removal.*

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

3.2 Arbitrary cast removal

We have shown that the number of casting planes that also allow opposite cast removal is linear. For the other casting planes, we know from Lemma 5 that they contain an edge of P. Since we may also assume that they contain a third vertex, we conclude:

Theorem 3 *Given a convex polyhedron P with n vertices, the maximum number of distinct casting planes for P is $O(n^2)$, assuming arbitrary removal of the cast parts.*

4 Algorithms for orthogonal and opposite cast removal

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

4.1 A simple algorithm for simple polyhedra

We compute $O(n)$ candidate casting planes as follows. By Lemma 7, we need only consider the casting planes of the convex hull of P. We first compute the candidate casting planes that intersect some edge properly, and then the ones that are generated. We only consider opposite cast removal; the case of orthogonal cast removal only requires some straightforward changes.

Let $E_1, \ldots, E_k$ be a partitioning of E into subsets of parallel edges. Let $V_i^{\subset}$ be the set of vertices that must be contained in the casting plane for the cast removal direction parallel to the edges of E_i. For each E_i, we compute the convex hull G_i^+ of the lower endpoints of E_i and the convex hull G_i^- of the upper endpoints of E_i. The convex hull of $|E_i|$ points in 3-dimensional space can be computed in $O(|E_i| \log |E_i|)$ time, see e.g. [10, 18]. Then we compute all planes that contain the vertices of $V_i^{\subset}$ and separate the interiors of G_i^+ and G_i^-, and contain at least three vertices of $V_i^{\subset}$, G_i^+ and G_i^-. This gives $O(|E_i|)$ candidate casting planes. Summed over all subsets $E_1, \ldots, E_k$, we obtain $O(n)$ candidate casting planes in $O(n \log n)$ time.

Second, we compute the other candidate casting planes, which are the generated planes of the hierarchical decomposition of Subsection 3.1. Antipodal pairs computations take $O(n)$ time and convex hull computations take $O(n \log n)$ time, see e.g. [10, 18]. The total time taken by the process is given by the recurrence $T(n) \leq T((1-\alpha)n) + O(n \log n)$, where α is the constant in the $\Omega(n)$ of Lemma 10. This recurrence solves to $T(n) = O(n \log n)$. We conclude:

Lemma 11 *Given a polyhedron P with n vertices, one can compute in $O(n \log n)$ time a set Q of $O(n)$ planes such that any casting plane h that contains at least three vertices of P is contained in Q, assuming opposite cast removal.*

Theorem 4 *Given a 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 result can be improved. Recall that determining whether a plane h is a casting plane for P can be done by only considering the facets intersected by h and the facets incident to the edges that are contained in h (this only holds for convex polyhedra), see Section 2.2. A linear program on this set of facets tells us whether h is a casting plane. We also know, by Theorem 2, that the total number of facets that we check, for all $O(n)$ candidate casting planes, is only $O(n \log n)$. To test each candidate casting plane, either generated or a member of an equivalence class of weakly equivalent planes, our algorithm walks around the intersection of the plane with the polyhedron at a cost of $O(\log n)$ per step. This leads to an $O(n \log^2 n)$ time algorithm for a convex polyhedron P with n vertices.

Theorem 5 *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.*

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 4 for simple polyhedra as well. Unfortunately, the storage requirements increase with the data structuring method. The idea is to test every candidate casting plane by querying with it in a data structure (instead of applying linear programming). The query determines whether h really is a valid casting plane for the polyhedron P. The preprocessing of the data structure should be less than quadratic and the query time should be less than linear in order to beat the quadratic time bound. It turns out that the data structure 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 2 *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 [14, 15] 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 δ of T, corresponding to a canonical subset V_δ, the secondary structure at δ stores V_δ as follows. Recall from Subsection 2.1 that for a facet f, $\Psi(f)$ is one of the closed half-spaces of which the bounding plane supports f, and $\Psi_0(f)$ is $\Psi(f)$ translated such that the bounding plane contains the origin. Let $F(V_\delta)$ be the set of facets incident to at least one vertex of V_δ. Let B_δ be the cone $\bigcap\{\Psi_0(f) \mid f \in F(V_\delta)\}$ with apex at the origin. The secondary structure is simply an array or balanced binary tree that stores the facets of B_δ 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_δ. 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 ℓ_h^+ be the upward half-line normal to h starting at the origin. For each of the nodes δ_i, we query with ℓ_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 $P \cap h^+$ is a terrain with respect to the direction normal to h (and parallel to ℓ_h^+). The query is repeated for h^-, to determine whether $P \cap h^-$ 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 $t = O(n^{1/2+\epsilon})$, 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 Δ^+ and Δ^- 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 in the origin of the common intersection of the s cones

$B_{\delta'_1}, \ldots, B_{\delta'_s}$, is non-trivial. One can decide whether the common intersection is non-trivial using the algorithm of Reichling [21]. 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 the reflected cones explicitly at every node and work with these, 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 casting plane. We determine the set Δ^+ as before; let $B_{\delta_1}, \ldots, B_{\delta_t}$ be the cones that are stored at the set Δ^+ 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 $P \cap h^+$ 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 $P \cap h^+$ is not a terrain with respect to any direction. If $P \cap h^+$ is a terrain for some direction, we repeat the query to test whether $P \cap h^-$ is a terrain.

Theorem 6 *For any constant $\epsilon > 0$ and any simple polyhedron P with n vertices, a data structure of size and prepreocessing 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 [7], Matoušek [15] 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.

Corollary 3 *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.*

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 5 and 6 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 4. If so, we are done. Otherwise, if P is convex, then by Lemma 5 we only have to consider casting planes that contain some edge of P. If P is non-convex, then, by Lemma 6, we only have to consider casting planes that contain an edge of the convex hull of P.

Observation 3 *Let P be a polyhedron and h a plane that contains an edge e of the convex hull of P. Assume w.l.o.g. that e is horizontal and that a vertical plane exists which supports e and has $P - cl(e)$ completely to the one side.*

- *If $P \cap h^+$ is a terrain and $P \cap h^-$ is not a terrain, then no plane μ containing e for which $P \cap h^- \subset \mu^-$ is a casting plane.*
- *If $P \cap h^+$ is not a terrain and $P \cap h^-$ is a terrain, then no plane μ containing e for which $P \cap h^+ \subset \mu^+$ is a casting plane.*
- *If $P \cap h^+$ and $P \cap h^-$ are both not a terrain, then no plane containing e is a casting plane.*

The above observation sets up a binary search for a casting plane that contains some edge e of the convex hull of P. A plane h rotating about e encounters the $n-2$ other vertices of P in some order. One binary search step on this order tests whether $P \cap h^+$ and $P \cap h^-$ are terrains using linear programming. Therefore, we have the following.

Theorem 7 *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.*

As shown in Subsection 4.3, we can preprocess P into a data structure such that any plane can be tested in

$O(n^{1/2+\epsilon})$ time. Since $O(n \log n)$ 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 solves 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. 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.

Let $\mathcal{D}(V) = \{\mathcal{D}(v_i) \mid v_i \in P\}$ be the set of planes dual to the 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 $q + \lambda \cdot \vec{u}$, $\lambda \geq 0$, where q is a point and $\vec{u}$ 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 $-\vec{u}$ to infinity. 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 8 *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 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. The n^ϵ-factors can be replaced by polylogarithmic factors if we use the partition trees of Matoušek [15]. 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 weakly 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. (De Berg [?] observed that orthogonal cast removal for simple polyhedra can be done in $O(n^{4/3+\epsilon})$ time by using a different data structure to test candidate casting planes.)

4. Suppose that we wish to determine castability of an object with non-linear boundaries. Give (simple) algorithms that deal with this problem.

5. For some casting processes, 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.

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

References

[1] Agarwal, P. K., and J. Matoušek, Ray shooting and parametric search. *Proc. 24th ACM Symp. Theory of Computing* (1992), pp. 517–526.

[2] Agarwal, P. K., and M. Sharir, Applications of a new space partitioning technique. *Proc. 2nd WADS* (1991), Lect. Notes in Comp. Science 519, Springer-Verlag, pp. 379–391.

[3] Asberg, B., G. Blanco, P. Bose, J. Garcia, M. Overmars, G. Toussaint, G. Wilfong, and B. Zhu, Feasibility of design in stereolithography. To appear in: *Proc. 13th Symp. on FST & TCS*, 1993.

[4] Bose, P., D. Bremner, and M. van Kreveld, Determining the Castability of Simple Polyhedra. Tech. Rep. SOCS 93.12, School of Comp. Science, McGill University, 1993.

[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. SOCS 92.16, School of Comp. Science, McGill University, 1992.

[7] Chazelle, B., M. Sharir, and E. Welzl, Quasi-optimal upper bounds for simplex range searching and new zone theorems. *Proc. 6th ACM Symp. on Comp. Geometry* (1990), pp. 23–33.

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

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

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

[11] Elliott, R., *Cast iron technology.* Butterworths, London, 1988.

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

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

[14] Matoušek, J., Efficient partition trees. *Proc. 7th ACM Symp. on Comp. Geometry* (1990), pp. 1–9.

[15] Matoušek, J., Range searching with efficient hierarchical cuttings. *Proc. 8th ACM Symp. on Comp. Geometry* (1991), pp. 276–285.

[16] 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.

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

[18] Preparata, F.P., and M.I. Shamos, *Computational geometry – an introduction.* Springer-Verlag, 1985.

[19] 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.

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

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

[22] Walton, C.F., and T.J. Opar (Eds.), *Iron castings handbook.* Iron casting society, Inc., 1981.

[23] Whelan, A., *Injection moulding machines.* Elsevier, London, 1984.