Casting an Object with a Core*

Hee-Kap Ahn[†] Sang Won Bae[†] Siu-Wing Cheng[‡] Kyung-Yong Chwa[†]

Abstract

This paper addresses geometric problems that concern manufacturing an object using a cast with a core. In casting, molten material is poured into the cavity of the cast and allowed to solidify. The cast has two main parts to be removed in opposite parting directions. To manufacture more complicated objects, the cast may also have a core to be removed in a direction skewed to the parting directions. In this paper, given an object and the parting and core directions, we give necessary and sufficient conditions to verify whether a cast can be constructed for these directions. In the case of polyhedral objects, we develop a discrete algorithm to perform the test in $O(n^3 \log n)$ time, where n is the object size. If the test result is positive, a cast with complexity $O(n^3)$ can be constructed within the same time bound. We also present an example to show that a cast may have $\Theta(n^3)$ complexity in the worst case. Thus, the complexity of our cast is worst-case optimal.

1 Introduction

Casting or injection molding [12, 7, 14] is ubiquitous in the manufacturing industry for producing consumer products. A cast can be viewed as a box with a cavity inside. Molten material (such as iron, glass or polymer) is poured into the cavity and allowed to solidify. Some cast consists of two parts and the hardened object is taken out by removing the two parts in opposite *parting directions*. See Figure 1 for an example. Many common objects need a core in additional to the two main parts in order to be manufactured. For example, consider a coffeemug in Figure 2(a). The handle of the mug can only be produced using the two main parts. However, these two main parts cannot produce the cavity. Figure 2(b) shows how the coffeemug can be manufactured by incorporating a core into the cast. The class of castable objects may be enlarged through the use of cores [6, 12, 14, 15], and cores are used widely

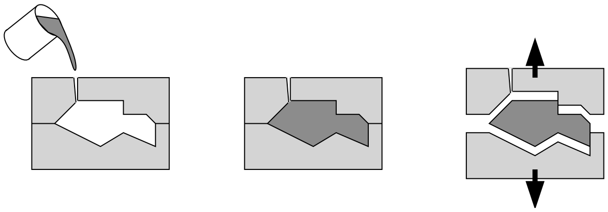


Figure 1: The casting process with a 2-part cast.

*Work by Ahn, Bae and Chwa was supported by the Brain Korea 21 Project, The School of Information Technology, KAIST, in 2005. Work by Cheng was supported by HKUST DAG04/05.EG21.

[†] Division of Computer Science, KAIST, Korea. Email: {heekap,swbae,kychwa}@tclab.kaist.ac.kr.

[‡] Department of Computer Science, HKUST, Hong Kong. Email: scheng@cs.ust.hk.

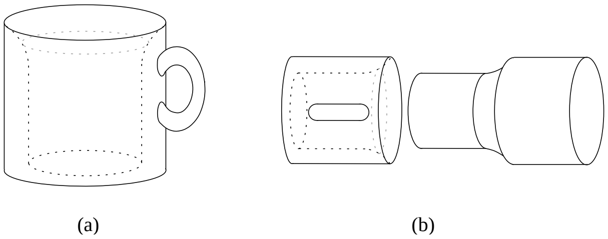


Figure 2: (a) A coffeemug is unattainable using a 2-part cast. (b) By incorporating a core to the cast, the cavity of the coffeemug can be manufactured.

in prevailing modes of production. In practice, a cast seldom has more than one core; otherwise, the retraction mechanism for the main parts and the cores would be too complicated. Figure 3 illustrates the removal of the main parts and the core in the manufacturing of an object. We require that the main parts and the core should be removed without being blocked by the cast or the object. This ensures that the given object can be mass produced by re-using the same cast. This paper is concerned with the verification of the geometric feasibility, *castability*, of the cast given the parting and core directions.

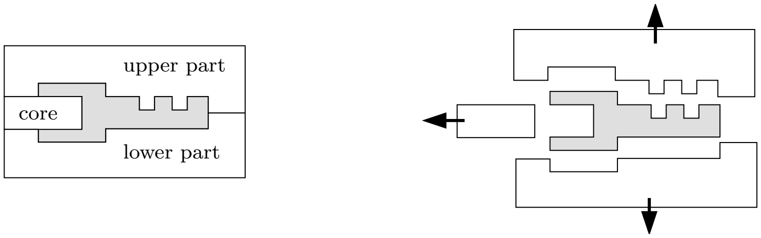


Figure 3: The casting process with a core.

There has been a fair amount of work on the castability problem [1, 3, 4, 5, 9, 10, 11] for the case that there is no core. Chen, Chou and Woo [6] described a heuristic to compute a parting direction to minimize the number of cores needed. However, the parting direction returned need not be feasible. Based on the approach of Chen, Chou and Woo, Hui presented exponential time algorithms to construct a cast [8]. However, there is no guarantee that a feasible cast will be found if there is one. Ahn et al. [2] proposed a hull operator, reflex-free hull, to define cavities in polyhedron. The motivation is that the cavities limit the search space for parting and core directions.

In this paper, we give the first exact characterization of the castability of an object given the parting and core directions. For a polyhedron of size n, we develop an $O(n^3 \log n)$-time algorithm for performing this test. The cast can be constructed within the same time bound. This paper presents, to the best of our knowledge, the first polynomial time algorithm for the problem.

Section 2 describes the basic definitions and notations. In Section 3, we prove the exact characterization. We develop our castability verification and cast construction algorithm in Section 4. We also present a lower bound construction that shows that the cast produced has optimal complexity in the worst-case. We conclude in Section 6

2 Preliminaries

Let A be a subset of $\mathbb{R}^3$. We say that A is *open* if for any point $p \in A$, A contains some ball centered at p with positive radius. We say that A is *closed* if its complement is open. For all points $p \in \mathbb{R}^3$, p is a *boundary point* of A, if any ball centered at p with positive radius intersects both A and its complement. The *boundary of* A, denoted by $\mathrm{bd}(A)$, is the set of boundary points. The *interior* of A, $\mathrm{int}(A)$, is $A \setminus \mathrm{bd}(A)$. Note that $\mathrm{int}(A)$ must be open.

We assume that the outer shape of the cast equals a box denoted by $\mathcal{B}$. The cavity of $\mathcal{B}$ has the shape of the object $\mathcal{Q}$ to be manufactured. We assume that $\mathcal{Q}$ is an open set so that the cast $\mathcal{B} \setminus \mathcal{Q}$ is a closed set. The box $\mathcal{B}$ is large enough so that $\mathcal{Q}$ is contained strictly in its interior. We use $\vec{d}_m$ and $-\vec{d}_m$ to denote the given opposite parting directions, and $\vec{d}_c$ to denote the given core direction.

We call the main part to be removed in direction $\vec{d}_m$ the *red cast part* and denote it by $\mathcal{C}_r$. We call the other main part the *blue cast part* and denote it by $\mathcal{C}_b$. We denote the core by $\mathcal{C}_c$. We require each cast part and the core to be a connected subset of $\mathcal{B}$ such that $\mathcal{B} \setminus \mathcal{Q} = \mathcal{C}_r \cup \mathcal{C}_b \cup \mathcal{C}_c$ and these three pieces only overlap along their boundaries.

Given the object $\mathcal{Q}$ and the directions $\vec{d}_m$ and $\vec{d}_c$, our problem is to decide if $\mathcal{Q}$ is *castable*. That is, whether $\mathcal{B}$ can be partitioned into $\mathcal{C}_r$, $\mathcal{C}_b$ and $\mathcal{C}_c$ so that they can be translated to infinity in their respective directions without colliding with $\mathcal{Q}$ and the other pieces. We assume that the order of removing the parts and the core is immaterial. In other words, if $\mathcal{Q}$ is castable, the parts and the core can be removed in any order without colliding with $\mathcal{Q}$ or the other pieces.

We will need some visibility and monotonicity concepts. A point $p \in \mathbb{R}^3$ is illuminated by light from direction $\vec{d}$ if a ray emitting from p towards $\vec{d}$ does not intersect $\mathcal{Q}$. We denote by $\mathcal{V}_m$ the subset of points of $\mathcal{B} \setminus \mathcal{Q}$ that are not illuminated by light from directions $\vec{d}_m$ or $-\vec{d}_m$. An object is *monotone* in direction $\vec{d}$ if for any line ℓ parallel to $\vec{d}$, the intersection between ℓ and the interior of the object is connected.

3 The characterization of castability

In this section, we develop the exact characterization of the castability of $\mathcal{Q}$ given the parting and core directions $\vec{d}_m$ and $\vec{d}_c$. Recall that we assume $\mathcal{Q}$ to be open but we do not require $\mathcal{Q}$ to be polyhedral. We first need some notations.

We use $\mathcal{V}_m^c$ to denote the points in $\mathbb{R}^3 \setminus \mathcal{Q}$ encountered when we sweep $\mathcal{V}_m$ to infinity in direction $\vec{d}_c$. Note that $\mathcal{V}_m^c$ includes $\mathcal{V}_m$ itself. Consider the illumination of $\mathcal{Q} \cup \mathcal{V}_m^c$ by lights from directions $\vec{d}_m$ and $-\vec{d}_m$. We use $\mathcal{V}_o$ to denote those points in $\mathcal{B} \setminus (\mathcal{Q} \cup \mathcal{V}_m^c)$ that are not illuminated. Then $\mathcal{V}_o^c$ denotes the points in $\mathbb{R}^3 \setminus (\mathcal{Q} \cup \mathcal{V}_m^c)$ encountered when we sweep $\mathcal{V}_o$ to infinity in direction $\vec{d}_c$.

We first prove two technical results.

Lemma 1 *Given $\vec{d}_m$ and $\vec{d}_c$, if $\mathcal{Q}$ is castable, then $\mathcal{V}_m^c \cap \mathcal{B} \subseteq \mathcal{C}_c$ and $\mathcal{V}_o^c \cap \mathcal{B} \subseteq \mathcal{C}_c$.*

Proof. Let p be a point in $\mathcal{V}_m$. By the definition of $\mathcal{V}_m$, if we move p in direction $\vec{d}_m$ or $-\vec{d}_m$ to infinity, p will hit $\mathcal{Q}$. So p cannot be a point in $\mathcal{C}_r$ or $\mathcal{C}_b$. Thus, $\mathcal{V}_m \subseteq \mathcal{C}_c$. Since $\mathcal{Q}$ is castable, $\mathcal{C}_c$ can be translated to infinity in direction $\vec{d}_c$ without colliding with $\mathcal{Q}$ and the red and blue cast parts. This implies that $\mathcal{V}_m^c \cap \mathcal{B} \subseteq \mathcal{C}_c$. By the definition of $\mathcal{V}_o$, if we move a point $q \in \mathcal{V}_o$ in direction $\vec{d}_m$ or $-\vec{d}_m$, q will hit $\mathcal{Q} \cup \mathcal{V}_m^c$. So q must be a point in $\mathcal{C}_c$, implying that $\mathcal{V}_o \subseteq \mathcal{C}_c$. Thus, the same reasoning shows that $\mathcal{V}_o^c \cap \mathcal{B} \subseteq \mathcal{C}_c$. ◻

Lemma 2 *Given $\vec{d}_m$ and $\vec{d}_c$, if $\mathcal{V}_m^c \cup \mathcal{V}_o^c$ is monotone in $\vec{d}_c$, $\mathcal{Q} \cup \mathcal{V}_m^c \cup \mathcal{V}_o^c$ is monotone in $\vec{d}_m$.*

Proof. Assume to the contrary that there exists a line ℓ parallel to $\vec{d}_m$ that intersects the interior of $\mathcal{Q} \cup \mathcal{V}_m^c \cup \mathcal{V}_o^c$ in at least two disjoint intervals. Let p be a point on ℓ that lies in between two consecutive intervals of $\ell \cap (\mathcal{Q} \cup \mathcal{V}_m^c \cup \mathcal{V}_o^c)$.

Let L be a line segment on ℓ through p such that L can be broken into three pieces s, s' and s'' so that $s, s'' \subset \mathcal{Q} \cup \mathcal{V}_m^c \cup \mathcal{V}_o^c$ and s' is outside $\mathcal{Q} \cup \mathcal{V}_m^c \cup \mathcal{V}_o^c$. It is impossible that both s and s'' lie inside $\mathcal{Q} \cup \mathcal{V}_m^c$; otherwise, p cannot be illuminated when $\mathcal{Q}$ is illuminated by light from directions $\vec{d}_m$ and $-\vec{d}_m$ (i.e., $p \in \mathcal{V}_m$), a contradiction. So s or s'', say s, intersects $\mathcal{V}_o^c$ only. Without loss of generality, assume that s is the end of L in direction $\vec{d}_m$.

Let q be a point on s. Let ξ be the ray emitting from q in direction $-\vec{d}_c$. Since $\mathcal{V}_m^c \cup \mathcal{V}_o^c$ is monotone in direction $\vec{d}_c$ by assumption, ξ must hit $\mathcal{V}_m \cup \mathcal{V}_o$. On the other hand, ξ cannot hit $\mathcal{V}_m$ because q would be in $\mathcal{V}_m^c$ instead. So ξ hits $\mathcal{V}_o$ and let q' be some point in $\xi \cap \mathcal{V}_o$. By the definition of $\mathcal{V}_o$, the ray emitting from q' in direction $-\vec{d}_m$ hits $\mathcal{V}_m^c$. (We choose the direction $-\vec{d}_m$ because the point q lies on the end s of L in direction $\vec{d}_m$.) Let L' be the maximal line segment on this ray such that q' is one endpoint of L' and L' lies outside $\mathcal{V}_m^c$. Notice that L' is not illuminated when $\mathcal{Q} \cup \mathcal{V}_m^c$ is illuminated by light from directions $\vec{d}_m$ and $-\vec{d}_m$. It follows that $L' \subseteq \mathcal{V}_o$. Translate L' in direction $\vec{d}_c$ until it overlaps with L and call this translated line segment L''. It follows that $L'' \subseteq \mathcal{V}_o^c$.

Observe that L'' and L share one common endpoint q, so we have $p \in L''$ or $p \notin L''$. If $p \in L''$, then $p \in \mathcal{V}_o^c$, contradicting our assumption that $p \notin \mathcal{Q} \cup \mathcal{V}_m^c \cup \mathcal{V}_o^c$. If $p \notin L''$, then L'' is contained in the line segment pq. Note that pq intersects $\mathcal{V}_o^c$ and pq does not intersect $\mathcal{Q} \cup \mathcal{V}_m^c$. Consider the endpoint v of L'' in the interior of pq. This endpoint v is the translated endpoint of L' whose neighborhood intersects $\mathcal{V}_m^c$. Thus, an arbitrarily small neighborhood of v must intersect $\mathcal{Q} \cup \mathcal{V}_m^c$. But then pq intersects $\mathcal{Q} \cup \mathcal{V}_m^c$, a contradiction. ◻

We are ready to prove the characterization.

Theorem 3 *Given $\vec{d}_m$ and $\vec{d}_c$, $\mathcal{Q}$ is castable if and only if $\mathcal{V}_m^c \cup \mathcal{V}_o^c$ is monotone in $\vec{d}_c$.*

Proof. If $\mathcal{Q}$ is castable, then for any point $p \in \mathcal{C}_c$, moving p to infinity in direction $\vec{d}_c$ will not hit $\mathcal{Q}$, $\mathcal{C}_r$, or $\mathcal{C}_b$. By Lemma 1, $(\mathcal{V}_m^c \cup \mathcal{V}_o^c) \cap \mathcal{B}$ is contained in $\mathcal{C}_c$. Therefore, by considering the movement of all points in $(\mathcal{V}_m^c \cup \mathcal{V}_o^c) \cap \mathcal{B}$ in direction $\vec{d}_c$, we conclude that $\mathcal{V}_m^c \cup \mathcal{V}_o^c$ is monotone in $\vec{d}_c$. This proves the necessity of the condition.

We prove the sufficiency by showing the construction of a cast for $\mathcal{Q}$. Ahn et al. [4] proved that an object is castable using a 2-part cast (without any core) in parting direction $\vec{d}$ if and only if the object is monotone in direction $\vec{d}$. Thus, Lemma 2 implies that $\mathcal{Q} \cap (\mathcal{V}_m^c \cup \mathcal{V}_o^c)$ is castable using a 2-part cast in direction $\vec{d}_m$. We use the construction by Ahn et. al [4] to build $\mathcal{C}_r$ and $\mathcal{C}_b$ from $\mathcal{B} \setminus (\mathcal{Q} \cup \mathcal{C}_c)$ with the necessary modification for handling the core. The details are as follows. Without loss of generality, we assume that $\vec{d}_m$ is the upward vertical direction, $\vec{d}_c$ makes angle of at most $\pi/2$ with $\vec{d}_m$, and the horizontal projection of $\vec{d}_c$ aligns with the positive x-axis.

Recall that the cast is made from a rectangular axis-parallel box $\mathcal{B}$. We make $\mathcal{B}$ sufficiently large and position $\mathcal{Q}$ inside $\mathcal{B}$ so that $\mathcal{V}_m^c \cup \mathcal{V}_o^c$ intersects the interior of one vertical side facet of $\mathcal{B}$ only. Let S be that side facet of $\mathcal{B}$. Thicken S slightly to form a slab S^+. Let T be the top horizontal facet of $\mathcal{B}$. Thicken T slightly to form one slab T^+.

We move $\mathcal{Q} \cup \mathcal{V}_m^c \cup \mathcal{V}_o^c$ upward to infinity to form one swept volume. Then we subtract $\mathcal{Q} \cup \mathcal{V}_m^c \cup \mathcal{V}_o^c$ from this swept volume to form a shape $\mathcal{X}$. We can almost make $\mathcal{X}$ the red cast part, but it is possible that $\mathcal{X}$ is disconnected. So we add T^+ to connect the components of $\mathcal{X}$ to form one red cast part $\mathcal{C}_r$. Similarly, we can almost make $(\mathcal{V}_m^c \cup \mathcal{V}_o^c) \cap \mathcal{B}$ the core, but it may be disconnected. So we add $S^+ \setminus \mathcal{C}_r$ to connect the components in $(\mathcal{V}_m^c \cup \mathcal{V}_o^c) \cap \mathcal{B}$ to form the core $\mathcal{C}_c$. Lastly, we construct the blue cast part $\mathcal{C}_b$ as $\mathcal{B} \setminus (\mathcal{Q} \cup \mathcal{C}_r \cup \mathcal{C}_c)$.

The fact that the cast parts and core can be removed in any order basically follows from the construction. It suffices to argue that each cast part or core can be removed first. Since $\mathcal{V}_m^c \cup \mathcal{V}_o^c$ is monotone in direction $\vec{d_c}$, the core $\mathcal{C}_c$ can be removed first without colliding with $\mathcal{Q}$ or the other cast parts. Consider $\mathcal{C}_r$. As $\mathcal{Q} \cup \mathcal{V}_m^c \cup \mathcal{V}_o^c$ is monotone in direction $\vec{d_m}$, the removal of $\mathcal{C}_r$ cannot collide with $\mathcal{Q}$ or $\mathcal{C}_c$. Clearly, the removal of $\mathcal{C}_r$ cannot collide with $\mathcal{C}_b$ by construction. The argument that $\mathcal{C}_b$ can be removed first is similar. ◻

If we are given a CAD system that is equipped with visibility computation, volume sweeping, and monotonicity checking operation, the characterization in Theorem 3 can be used directly to check the castability of any object. The proof also yields the construction of the cast.

4 An algorithm for polyhedra

In this section, we apply Theorem 3 to check the castability of a polyhedron. The goal is to obtain a discrete algorithm whose running time depends on the combinatorial complexity of the polyhedron. To be consistent with the previous section, our object is the interior of the polyhedron and we denote it by $\mathcal{P}$. The combinatorial complexity n of $\mathcal{P}$ is the sum of the numbers of vertices, edges, and facets in $\mathrm{bd}(\mathcal{P})$. We present an $O(n^3 \log n)$-time algorithm for testing the castability of $\mathcal{P}$ given $\vec{d_m}$ and $\vec{d_c}$. During the verification, we compute $\mathcal{V}_m^c \cup \mathcal{V}_o^c$, from which the cast $\mathcal{C}$ can be easily obtained as mentioned in the proof of Theorem 3.

Throughout this section, we assume that $\vec{d_m}$ is the upward vertical direction. We also make two assumptions about non-degeneracy. First, no facet in $\mathrm{bd}(\mathcal{P})$ is vertical. Second, the vertical projections of two polyhedron edges are either disjoint or they cross each other. These non-degenerate assumptions simplify the presentation and they can be removed by a more detailed analysis. We call a facet of $\mathcal{P}$ an *up-facet* if its outward normal points upward, and a *down-facet* if its outward normal points downward.

Let $\mathcal{H}$ be a horizontal plane below $\mathcal{P}$. We project all facets of $\mathcal{P}$ onto $\mathcal{H}$. The projections may self-intersect and we insert vertices at the crossings. The resulting subdivision has $O(n^2)$ size and we denote it by $\mathcal{M}$. For each cell of $\mathcal{M}$, we keep the set of polyhedron facets that cover it. We can compute $\mathcal{M}$ in $O(n^2 \log n)$ time using a plane-sweep algorithm and the association of polyhedron facets to cells can be done in $O(n^3 \log n)$ time.

After computing $\mathcal{M}$, we test whether $\mathcal{V}_m^c \cup \mathcal{V}_o^c$ is monotone in $\vec{d_c}$ (see Theorem 3). We partition $\mathcal{H}$ into 2D slabs by taking vertical planes parallel to $\vec{d_c}$ through all vertices of $\mathcal{M}$. Since there are $O(n^2)$ vertices in $\mathcal{M}$ and a vertical plane parallel to $\vec{d_c}$ intersects $O(n)$ edges of $\mathcal{P}$, there are $O(n^3)$ intersections in total. So the overlay of $\mathcal{M}$ and the slabs can be computed in $O(n^3 \log n)$ time using a plane-sweep algorithm.

Consider a slab Σ on $\mathcal{H}$. From the construction, Σ contains no vertex in its interior and is partitioned into $O(n)$ regions by the edges of $\mathcal{M}$. Let $\vec{d}$ be the projection of $\vec{d_c}$ on $\mathcal{H}$. The regions in Σ are linearly ordered in direction $\vec{d}$ and we label them by Δ_0, Δ_1, ... in this order. Notice that Δ_0 is unbounded in direction $-\vec{d}$ and the last region is unbounded in direction $\vec{d}$. We use ζ_i to denote the boundary edge between Δ_{i-1} and Δ_i. For each region Δ_i, we keep the set of polyhedron facets that cover it. We cannot do this straightforwardly. Otherwise, since there are $O(n^3)$ regions over all slabs and we may keep $O(n)$ polyhedron facets per region, the total time and space needed to do this would be $O(n^4)$. The key observation is that the total number of changes in the sets of polyhedron facets over all regions in Σ is $O(n)$. Therefore, we can use a persistent search tree [13] to store the sets of polyhedron facets for all regions in Σ. This takes $O(n \log n)$ time and $O(n)$ space to build per slab. Hence, it takes a total of $O(n^3 \log n)$ time and $O(n^3)$ space.

We employ an inductive strategy for testing the monotonicity of $\mathcal{V}_m^c \cup \mathcal{V}_0^c$ in $\vec{d_c}$ within the unbounded 3D slab $\Sigma \times [\infty, -\infty]$ for each 2D slab Σ on $\mathcal{H}$. Repeating this test for all 2D slabs on $\mathcal{H}$ gives the final answer. We scan the regions in Σ in the order $\Delta_0, \Delta_1, \ldots$. During the scanning, we incrementally grow a volume $\mathcal{V}^c$. The volume $\mathcal{V}^c$ is initially empty and $\mathcal{V}^c$ will be equal to $(\mathcal{V}_m^c \cup \mathcal{V}_o^c) \cap (\Sigma \times [\infty, -\infty])$ in the end.

We first discuss the data structures that we need to maintain during the scanning. Consider the event that we cross the boundary ζ_i and that the portion of $\mathcal{V}_m^c \cup \mathcal{V}_o^c$ encountered so far is monotone in $\vec{d_c}$. Take the vertical strip through ζ_i. We translate this strip slightly into Δ_{i-1} (resp. Δ_i) and denote the perturbed strip by H_i^- (resp. H_i^+). Let I_i^- denote the intersection $H_i^- \cap (\mathcal{V}_m^c \cup \mathcal{V}_o^c)$ and let I_i^+ denote the intersection $H_i^+ \cap (\mathcal{V}_m^c \cup \mathcal{V}_o^c)$. Both I_i^- and I_i^+ consist of $O(n)$ trapezoids. We call the upper and lower sides of each trapezoid its *ceiling* and *floor*, respectively. The ceiling of each trapezoid τ lies on a boundary facet of $\mathcal{V}_m^c \cup \mathcal{V}_o^c$. We call this boundary facet the *ceiling-facet* of τ. This ceiling-facet may lie within a down-facet in $\mathrm{bd}(\mathcal{P})$ or it may be parallel to $\vec{d_c}$ and disjoint from $\mathrm{bd}(\mathcal{P})$. The latter kind of facets are generated by the sweeping towards $\vec{d_c}$. Therefore, it suffices to store a polyhedron facet or a plane parallel to $\vec{d_c}$ to represent the ceiling-facet. We denote this representation by $f_u(\tau)$. Similarly, the floor of τ lies on a boundary facet of $\mathcal{V}_m^c \cup \mathcal{V}_o^c$. This boundary facet may lie within a up-facet of $\mathcal{P}$ or it may be parallel to $\vec{d_c}$ and disjoint from $\mathrm{bd}(\mathcal{P})$. We call it the *floor-facet* of τ and denote its representation by $f_\ell(\tau)$.

We are ready to describe the updating strategy when we reach a new region Δ_i. We first discuss the monotonicity test. Later, we discuss how to grow $\mathcal{V}^c$ if the test is passed. Note that there is a change in the polyhedron facets that cover Δ_{i-1} and Δ_i. There are several cases.

1. For any trapezoid $\tau \in I_i^-$, neither $f_u(\tau)$ nor $f_\ell(\tau)$ is about to vanish above ζ_i. Then some polyhedron edge e must project vertically onto ζ_i. Also, the vertical projections of the two incident polyhedron facets of e cover Δ_i but not Δ_{i-1}. Consider the projection e^- of e in direction $-\vec{d_c}$ onto H_i^-. This projection e^- cannot lie between two trapezoids in I_i^- because the space between two trapezoids is the polyhedron interior. So there are only two cases:

   (a) The projection e^- cuts across the interior of a trapezoid $\tau \in I_i^-$. In this case, we abort and report that $\mathcal{P}$ is not castable. The reason is that one polyhedron facet incident to e must block the sweeping of τ towards $\vec{d_c}$. It follows that $\mathcal{V}_m^c \cup \mathcal{V}_o^c$ is not monotone in $\vec{d_c}$ and so $\mathcal{P}$ is not castable by Theorem 3.

   (b) The projection e^- lies above all trapezoids. The case that e^- lies below all trapezoids can be handled symmetrically. Let f be the down-facet incident to e. If we project e vertically downward, the projection either lies on some up-facet f', or a boundary facet of $\mathcal{V}_m^c \cup \mathcal{V}_o^c$ that is parallel to $\vec{d_c}$, or lies at infinity. The last case happens when I_i^- is empty and Δ_{i-1} is empty (e.g., when we cross the boundary ζ_1 between Δ_1 and Δ_0) and there is nothing to be done for this case. We discuss the other two cases. Let e' denote this vertical downward projection of e.

    i. If e' lies on a up-facet f', then e and e' define a new trapezoid τ that lies above all trapezoids in I_i^- and that $f_u(\tau) = f$ and $f_\ell(\tau) = f'$. I_i^+ contains all trapezoids in I_i^- as well as τ. However, if the outward normal of f makes an obtuse angle with $\vec{d_c}$, then f blocks the sweeping of τ towards $\vec{d_c}$ and we should abort and conclude as before that $\mathcal{P}$ is not castable.

    ii. If e' lies on a boundary facet of $\mathcal{V}_m^c \cup \mathcal{V}_o^c$ that is parallel to $\vec{d_c}$, then e' actually lies on $f_u(\tau)$ where τ is the topmost trapezoid in I_i^-. Thus, we should grow τ upward and set $f_u(\tau) = f$. I_i^+ contains this updated trapezoid τ and the other trapezoids in I_i^-. There is no change in the monotonicity status.

2. For some trapezoid $\tau \in I_i^-$, $f_u(\tau)$ or $f_\ell(\tau)$ is about to vanish above ζ_i. In this case, a polyhedron edge e bounds $f_u(\tau)$ or $f_\ell(\tau)$ and e projects vertically onto ζ_i. There are two cases:

    (a) The polyhedron facets incident to e lie locally on different sides of the vertical plane through ζ_i. Let f be the incident facet of e that lies locally in direction $\vec{d_c}$ from e. In this case, the vanishing $f_u(\tau)$ or $f_\ell(\tau)$ should be replaced by f. However, if the outward normal of f makes an obtuse angle with $\vec{d_c}$, we should abort and conclude as before that $\mathcal{P}$ is not castable.

    (b) Otherwise, both incident facets of e lie locally in direction $-\vec{d_c}$ from e. There is no change in monotonicity status, but we need to perform update as follows. Let f be the vanishing $f_u(\tau)$ or $f_\ell(\tau)$ of τ. There are two cases:

        i. There are trapezoids in I_i^- that lie above and below f. Clearly, τ is one of them. Let τ' be the other trapezoid. Then $f_u(\tau')$ or $f_\ell(\tau')$ is about to vanish above ζ_i too. In this case, we should merge τ and τ' into one trapezoid. The ceiling-facet and floor-facet of this merged trapezoid are the non-vanishing ceiling-facet and floor-facet of τ and τ'. I_i^+ contains this merged trapezoid and the trapezoids in I_i^- other than τ and τ'.

        ii. All trapezoids in I_i^- lie locally on one side of f. Assume that τ is the topmost trapezoid in I_i^-. The other case can be handled symmetrically. Then $f = f_u(\tau)$. It means that we are about to sweep the shadow volume below f and bounded by τ into the space above Δ_i. Thus, we should set $f_u(\tau)$ to be the plane that passes through e and is parallel to $\vec{d_c}$. I_i^+ contains this updated trapezoid τ and the other trapezoids in I_i^-.

If we represent each trapezoid in I_i^- combinatorially by its ceiling-facet and floor-facet, the above description tells us how to update I_i^- combinatorially to produce I_i^+. Notice that I_i^+ will be treated as I_{i+1}^- when we are about to cross the boundary ζ_{i+1} in the future. By storing the trapezoids in I_i^- in a balanced binary search tree, the update at ζ_i can be performed in $O(\log n)$ time. Since there are $O(n)$ regions in Σ, scanning Σ takes $O(n \log n)$ time. Summing over all 2D slabs on $\mathcal{H}$ gives a total running time of $O(n^3 \log n)$.

What about growing $\mathcal{V}^c$ into the space above Δ_i? After the update, for each trapezoid $\tau \in I_i^+$, $f_u(\tau)$ and $f_\ell(\tau)$ cut $\Delta_i \times [\infty, -\infty]$ into two unbounded solid and one bounded solid B_τ. Conceptually, we can grow $\mathcal{V}^c$ by attaching B_τ for each trapezoid $\tau \in I_i^+$, but this is too consuming. Observe that if I_i^+ merely inherits a trapezoid τ from I_i^-, there is no hurry to sweep τ into the space above Δ_i. Instead, we wait until ζ_j for the smallest $j > i$ such that I_j^+ does not inherit τ from I_{j-1}^-. Then $f_u(\tau)$ and $f_\ell(\tau)$ cut $R \times [\infty, -\infty]$ into two unbounded solids and one bounded solid S_τ, where R is the area within Σ bounded by ζ_i and ζ_j. We attach S_τ to grow $\mathcal{V}^c$. By adopting this strategy, we spend $O(1)$ time to grow

$\mathcal{V}^c$ when we cross a region boundary. Hence, we spend a total of $O(n^3)$ time to construct $\mathcal{V}_m^c \cup \mathcal{V}_o^c$. Once $\mathcal{V}_m^c \cup \mathcal{V}_o^c$ is available, we can construct the cast in $O(n^3)$ time as explained in the proof of Theorem 3.

Theorem 4 *Given $\vec{d_m}$ and $\vec{d_c}$, the castability of a polyhedron with size n can be determined in $O(n^3 \log n)$ time and $O(n^3)$ space. If castable, the cast can be constructed in the same time and space bounds.*

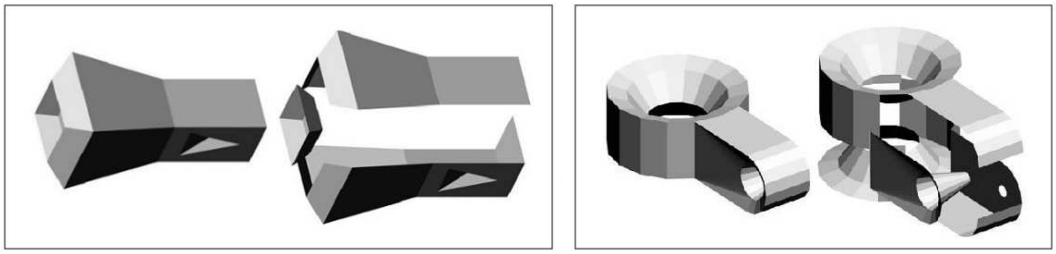


Figure 4: The boundary of each object is partitioned into three pieces in accordance with the pair of its removal directions in which the object has been verified castable.

We developed a preliminary implementation of the algorithm of Theorem 4. Figure 4 shows the output of our implementation on two polyhedra. The direction $\vec{d_m}$ is the upward vertical direction. The direction $\vec{d_c}$ is the leftward direction. In Figure 4, we show the partition of the polyhedron facets depending on which cast part they belong to. For the ease of visualization, each boundary patch is translated slightly in its corresponding removal direction.

5 Worst-case example

In this section, we present a lower bound construction showing that a castable polyhedron of size n can require a cast of $\Omega(n^3)$ size. Thus the space complexity in Theorem 4 is worst-case optimal and the time complexity of our algorithm is at most a $\log n$ factor off the worst-case optimum. Throughout this section, we assume that $\vec{d_m}$ is the upward vertical direction and $\vec{d_c}$ is the leftward direction.

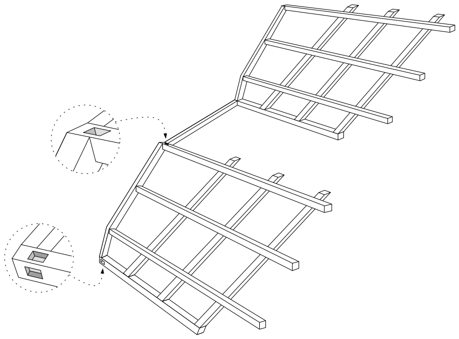
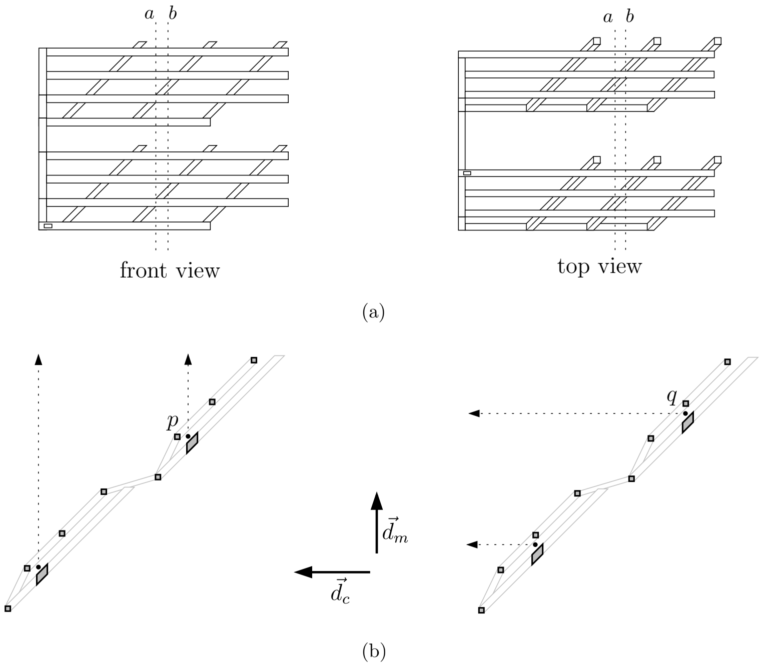


Figure 5: The lower bound example in a perspective view.

Figure 5 shows our lower bound construction. The polyhedron consists of two parts: the upper part has four horizontal legs in a staircase and three slanted legs sitting on a horizontal leg. The lower part is an almost identical copy of the upper part, except that it has three small holes as shown in the figure. The upper hole can only be covered by the red cast part to be removed vertically upward, and the other two holes only be covered each by the core and the blue cast part. Figure 6(a) shows the front view (when we look at it from the left) and the top view of the polyhedron $\mathcal{P}$. In both projections, all three horizontal legs cross the other three slanted legs in the upper part as well as in the lower part.



(a)

(b)

Figure 6: (a) A top view and a side view of the lower bound construction. (b) Two cross sections along a (left) and b (right). The only way to remove p (resp. q) is translating it in $\vec{d_m}$ (resp. $\vec{d_c}$).

Clearly, the polyhedron is castable with respect to the given $\vec{d_m}$ and $\vec{d_c}$. We argue that the cast has $\Omega(n^3)$ size. Imagine that we take a plane h parallel to $\vec{d_m}$ and $\vec{d_c}$, and sweep $\mathcal{P}$ with h in the direction orthogonal to h. During the sweep, the cross-section $\mathcal{P} \cap h$ is 2-part castable (in 2D) with parting direction $\vec{d_m}$, except when h intersects a volume in $\mathcal{V}_m$. When this happens, $\mathcal{P} \cap h$ is castable with parting direction $\vec{d_m}$ and core direction $\vec{d_c}$. Figure 6(b) shows two cross-sections during the sweep. The left and right pictures in Figure 6(b) show the cross-section at positions a and b, respectively, shown in Figure 6(a). In the left cross section, let p be a point in $\mathcal{B} \setminus \mathcal{P}$, lying inbetween a horizontal leg and a slanted leg in direction $\vec{d_c}$ and above the slanted leg. The only way to remove p is translating it in $\vec{d_m}$, which means that the set of points in $\mathcal{B} \setminus \mathcal{P}$ hit by the ray from p in direction $\vec{d_m}$ to infinity must belong to $\mathcal{C}_r$. Now let q be a point in $\mathcal{V}_m$ in the right cross section. Analogously, the set of points in $\mathcal{B} \setminus \mathcal{P}$ hit by the ray from q in direction $\vec{d_c}$ to infinity must belong to $\mathcal{V}_m^c$ and hence $\mathcal{C}_c$. In addition, the set of points in $\mathcal{B} \setminus \mathcal{P}$ that lies inbetween the ray and $\mathcal{P}$ in direction $\vec{d_m}$ belong to $\mathcal{C}_c$ since these points belong to the set $\mathcal{V}_o$ by definition. While we sweep $\mathcal{P}$ with h, these two kinds of cross sections appear alternately.

We put $\Theta(n)$ horizontal legs and $\Theta(n)$ slanted legs in both the upper and the lower parts. In the upper part, each slanted leg must be in contact with both $\mathcal{C}_r$ and $\mathcal{C}_c$. Moreover, the contacts with $\mathcal{C}_r$ and $\mathcal{C}_c$ alternate $\Theta(n)$ times along the slanted leg. As a result, the slanted legs in the upper part have a total of $\Theta(n^2)$ contacts with $\mathcal{C}_c$. These contacts sweep in direction $\vec{d_c}$ and generate $\Theta(n^2)$ swept volumes. The merging of any two such swept volumes is forbidden by the alternate appearances of the left cross-section in Figure 6(b). Each swept volume projects vertically and produces a shadow on each horizontal leg that lies below it. Thus, the total complexity of $\mathcal{C}_c$ is $\Omega(n^3)$.

6 Conclusion

Given the parting direction and the core direction, we presented an exact characterization of the castability of an object. Based on this characterization, we developed an $O(n^3 \log n)$-time and $O(n^3)$-space algorithm for testing a polyhedron of size n. We presented a lower bound construction to show that our cast complexity is worst-case optimal.

References

[1] H.K. Ahn, S.W. Cheng, and O. Cheong. Casting with skewed ejection direction. In *Proc. 9th Annu. International Symp. on Algorithms and Computation,*, volume 1533 of *Lecture Notes in Computer Science*, pages 139–148. Springer-Verlag, 1998.

[2] H.K. Ahn, S.W. Cheng, O. Cheong, and J. Snoeyink. The reflex-free hull. *International Journal of Computational Geometry and Applications*, 14(6):453–474, 2004.

[3] H.K. Ahn, O. Cheong, and R. van Oostrum. Casting a polyhedron with directional uncertainty. *Computational Geometry: Theory and Applications*, 26(2):129–141, 2003.

[4] H.K. Ahn, M. de Berg, P. Bose, S.W. Cheng, D. Halperin, J. Matoušek, and O. Schwarzkopf. Separating an object from its cast. *Computer-Aided Design*, 34:547–559, 2002.

[5] P. Bose and G. Toussaint. Geometric and computational aspects of gravity casting. *Computer-Aided Design*, 27(6):455–464, 1995.

[6] L.L. Chen, S.Y. Chou, and T.C. Woo. Parting directions for mould and die design. *Computer-Aided Design*, 25:762–768, 1993.

[7] R. Elliot. *Cast Iron Technology*. Butterworths, London, 1988.

[8] K. Hui. Geometric aspects of mouldability of parts. *Computer Aided Design*, 29(3):197–208, 1997.

[9] K.C. Hui and S.T. Tan. Mould design with sweep operations—a heuristic search approach. *Computer-Aided Design*, 24:81–91, 1992.

[10] K. K. Kwong. *Computer-aided parting line and parting surface generation in mould design*. PhD thesis, The University of Hong Kong, Hong Kong, 1992.

[11] J. Majhi, P. Gupta, and R. Janardan. Computing a flattest, undercut-free parting line for a convex polyhedron, with application to mold design. *Computational Geometry: Theory and Applications*, 13:229–252, 1999.

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

[13] N. Sarnak and R.E. Tarjan. Planar point location using persistent search trees. *Communications of the ACM*, 29:669–679, 1986.

[14] C.F. Walton and T.J. Opar, editors. *Iron Castings Handbook*. Iron casting society, Inc., 1981.

[15] E. C. Zuppann. Castings made in sand molds. In J. G. Bralla, editor, *Handbook of Product Design for Manufacturing*, pages 5.3–5.22. McGraw-Hill, New York, 1986.