

DETC2007-34688

FINDING ALL UNDERCUT-FREE PARTING DIRECTIONS FOR SOLIDS OF REVOLUTION

Xiaorui Chen

Sara McMains*

Department of Mechanical Engineering
University of California
Berkeley, California 94720
Email: {xrchen | mcmains} @me.berkeley.edu

ABSTRACT

For molding and casting processes, geometries that have undercut-free parting directions (UFPDs) are preferred for manufacturing. However, existing approaches either cannot identify all UFPDs or cannot run at interactive speeds (the best exhaustive algorithm, unimplemented, runs at $O(n^4)$ time theoretically). Our proposed feature-based approach avoids testing the whole Gaussian sphere of potential directions by first efficiently identifying all UFPDs for individual features such as extruded and revolved features, thus significantly reducing test space and running time. In this paper, we describe a fast algorithm to find all UFPDs for solids of revolution. The algorithm is based on analyzing the constructing 2D generator profiles, building on our previous results for 2-moldability analysis of polygons. The running time is $O(n)$, where n is the geometric complexity of the 2D generator profile. For parts containing multiple solids of revolution, the set of possible UFPDs can be significantly reduced based upon an analysis of each such feature, efficiently identifying many as non-2-moldable or reducing the search space for exhaustive algorithms that find all UFPDs.

1 INTRODUCTION

In molding or casting manufacturing processes, the molten material is shaped in a hollow mold. After the material solidifies,

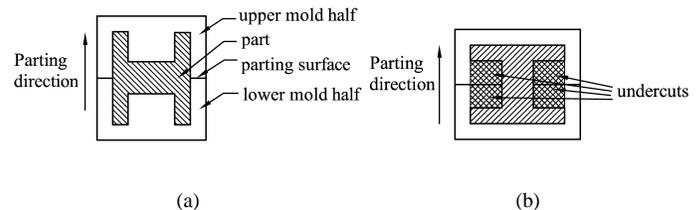


Figure 1. (a) Mold for a simple 2D part; (b) an orientation of the same part with undercuts.

the part is ejected out of the mold. Simple reusable molds consist of two rigid halves that move in opposite directions during the mold closing and opening operations, to permit the part to be extracted. The direction of motion of the mold halves is called the *parting direction*. The two mold halves meet at the *parting surface* (see Fig. 1(a)), which may be planar or non-planar. The part geometry is said to be *2-moldable (monotone)* in a direction \vec{d} if the mold halves forming it can be translated to infinity along \vec{d} and $-\vec{d}$, respectively, without collision with the interior of the part. The part shown in Fig. 1(a) is 2-moldable in the vertical direction; the same part with a different orientation shown in Fig. 1(b) is not 2-moldable in the vertical direction. Surfaces that prevent the mold from releasing the part are called *undercuts*.

In the presence of undercuts, the mold may require extra movable sections, which are inserted into the mold before the

*Address all correspondence to this author.

molten material is shaped, to form particular features that cannot be formed using only two mold halves. These undercuts not only increase the mold cost but also shorten the mold life. Therefore, all else being equal, geometries with undercut-free parting directions (UFPDs) are preferred.

Since approximately 80% of injection molding manufacturing cost typically becomes fixed in the design stage, according to our interviewees in the injection-molded part design industry, and the flexibility to make design changes rapidly diminishes later in the design cycle, it is vital that designers are provided early feedback on their design. Our research concentrates on finding all UFPDs at interactive speeds during the early design stage to help designers maintain a geometry with at least one UFPD.

2 RELATED WORK

Whether a given part geometry allows a UFPD has been studied by many researchers. Previous research can be divided into two approaches: heuristic or exhaustive.

2.1 Heuristic Approaches

Heuristic algorithms only tested a limited number of potential parting directions for 2-moldability, such as the three principle axes [1, 2], the bounding box axes [3], or the normal directions [4, 5]. If a UFPD is in the test set, the approach will find it; otherwise, the approach will not find it.

Other researchers used concavity features or graph-based algorithms. The concavity feature algorithms are based on first performing a regularized subtraction of the object from its convex hull, then identifying potential undercuts, which are called “pockets.” Each “pocket” is a set of connected surfaces that belong to the original object but not to the convex hull [6–12]. The optimal parting directions are chosen by different criteria such as the number of undercuts and draft angles. This approach, however, cannot identify directions as UFPDs when portions of a single “pocket” can only be formed by different halves of the mold, which unfortunately is not uncommon in industry. Graph-based algorithms recognized potential undercut features using boundary representation graphs [13–17]. This approach works for detecting depression or protrusion undercuts. However, it breaks down for complex features, where depressions and protrusions interact with each other.

2.2 Exhaustive Approaches

Exhaustive approaches to finding if *any* UFPDs exist for a given geometry have been presented in both two and three dimensions. Rappaport and Rosenbloom gave an $O(n)$ time algorithm to determine if a 2D polygon with n vertices is 2-moldable in arbitrary (not necessarily opposite) removal directions, and an $O(n \log n)$ time algorithm for opposite removal directions [18].

In our previous work, we developed an $O(n)$ time algorithm to find all UFPDs for a 2D polygon bounded by straight line and/or curved edges [19].

Bose and Bremner presented algorithms to determine the existence of a UFPD for a genus-zero polyhedron [20]. Their algorithms only find UFPDs with planar parting surfaces; UFPDs with non-planar parting surfaces are ignored. Ahn et al. presented an exhaustive algorithm to find all the combinatorially distinct UFPDs for a 3D (faceted) polyhedron in time $O(n^5 \log n)$ and a more efficient but more complicated algorithm that runs in time $O(n^4)$. They divided the whole Gaussian sphere into connected regions formed by an arrangement of great circles and great circular arcs. Within each region, the polyhedron has the same 2-moldability for all directions; hence only one direction needs to be tested in each region, accomplished by testing the vertices of the arrangement. In their paper, they also proved that the $O(n^4)$ time complexity is optimal in the worst case by presenting an example with $\Omega(n^4)$ combinatorially distinct parting directions. However, in practice, due to their algorithm’s complexity, their implementation instead reverted to testing heuristically chosen directions based on input edge orientations and additional randomly chosen test directions. Elber et al. gave an exact solution for a model bounded by NURBS surfaces, but it is restricted to a completely smooth boundary that is C^3 everywhere [21]. Khardekar et al. developed a programmable-graphics-hardware accelerated algorithm to test the combinatorially distinct UFPDs for a triangulated polyhedron. Their implementation can graphically display the undercut for a particular parting direction in linear time with respect to the number of triangular faces in the solid model. To find a UFPD, they tested the face normal directions and the intersections of spherical convex hulls [22], each of which bounds an inaccessible region on the Gaussian sphere corresponding to directions in which a pair of faces may occlude each other [23]. Although finding a single UFPD is greatly accelerated using their algorithm, either finding *all* UFPDs or definitely stating that *no* UFPDs exist takes $O(n^5)$ time.

In summary, heuristic approaches cannot always find a direction that is indeed a UFPD, and among the existing exhaustive approaches for general 3D polyhedra, the best theoretical time complexity is $O(n^4)$, which is not interactive for anything beyond relatively simple geometries. To take advantage of the completeness of the exhaustive algorithms and make them practically useful, speeding up the running time is thus the focus of our research. In the next section, we will give an overview of our proposed feature-based approach to finding all UFPDs.

3 FEATURE-BASED APPROACH TO FINDING ALL UFPDS

Currently most CAD systems use feature-based design, which maintains a design history for later editing and reconstruct-

tion. The overall geometry is obtained by performing Boolean operations on a tree of features, commonly including extrusion, revolution, sweep and loft features. These features are usually constructed by first drawing 2D contours, which we call 2D generator profiles, and then performing 3D operations on the 2D contours.

We observe that for unions of features, a direction is a UFPD for the overall geometry only if it is a UFPD for every individual feature. Based on this observation, we have proposed a new feature-based approach to finding all UFPDs by first finding a set of UFPDs for each feature and then testing only the directions in the intersection of all such sets using an exhaustive algorithm. Using the proposed approach, the set of possible UFPDs for a part containing individual features can be reduced for each feature added to the design tree, efficiently identifying many parts that have no UFPDs and reducing the search space for exhaustive algorithms that find all UFPDs. Moreover, the feature-based analysis can be implemented incrementally. Potential UFPDs for each individual feature are only computed once. Whenever a new feature is added, the previously tested potential UFPDs by the exhaustive algorithm is re-classified only against the new feature before running the exhaustive algorithm again, saving re-computation time when the overall geometry is updated. We have presented our algorithm finding all UFPDs for extrusions in [24]. In this paper, we show how to find all UFPDs for solids of revolution.

4 ASSUMPTIONS AND BACKGROUND

4.1 Assumptions

In this paper, we study solids of revolution that are formed by rotating 2D generator profiles 360° around their coplanar axes. We assume that the 2D generator profiles are polygons composed of only straight line segments (so the cross sections for the resulting solids of revolution are bounded by straight lines and/or hyperbolic curves). The boundary of the 2D generator profile does not cross but may touch the axis of revolution, which is the case in CAD systems such as SolidWorks. Without loss of generality, we assume that both the 2D generator profile and the axis of revolution lie on the x - z plane and that the axis of revolution coincides with the z axis. When the 2D generator profile has holes, the result is trivial since the solid of revolution contains voids; no directions are UFPDs. Thus we only consider the case where the 2D generator profile is a simple polygon that has no holes. We also assume that the 2D generator profile is non-self-intersecting.

For a polygon, possibly curved if it is the boundary of a cross section for the solid of revolution, we use the right-hand rule convention that the edges of the polygon are oriented in such a way that the interior of the polygon lies on the left when moving along the directed edges — that is, the boundary of the polygon is oriented counterclockwise. The edge normals are unit vectors

pointing towards the exterior of the polygon.

To represent all directions in 2D Euclidean space, we use a Gaussian circle; each direction in 2D can be represented by a point on the Gaussian circle by normalizing the direction to a unit vector and placing its tail at the origin. Similarly, we use a Gaussian sphere to represent all possible directions in 3D. The $+z$ direction is mapped to the north pole and the $-z$ direction is mapped to the south pole.

4.2 UFPDs for 2D Curved Polygons

This section summarizes the 2-moldability analysis for 2D curved polygons presented in our previous work [19], which is used in our algorithm for finding all UFPDs for solids of revolution. For our 2D algorithm, we introduced a data structure called the *normal graph* that captures the edge normals and their connectivity, given a possibly curved polygon. By traveling around the polygon counterclockwise, starting from any edge, the edge normals are mapped onto the Gaussian circle. They are called *normal points*. Each straight edge corresponds to one unique normal point. Each simple (i.e., G^1 -continuous but without any inflection points) curve corresponds to two normal points, denoting normals at the start point and end point. Normal points may coincide with each other. Two sequentially mapped normal points (not necessarily adjacent on the normal graph) representing different edges are connected by an arc less than 180° around the Gaussian circle. The two connected normal points may represent two adjacent straight edges, one straight edge and the start point of the (next) curved edge, the end point of the curved edge and the (next) straight edge, the end point of the curved edge and the start point of the (next) curved edge, or the last mapped normal point and the first mapped normal point. Two normal points representing the start point and the end point of the same simple curve are also connected by an arc, which is oriented counterclockwise if the curvature of the curve is positive and clockwise if the curvature is negative (the curvature is positive if its center lies on the left when moving along the curve, negative if its center lies on the right). The arcs connecting the normal points are called *normal arcs*. Normal arcs together with normal points form the *normal graph*. An example normal graph is shown in Fig. 2(b).

Each normal arc connects two normal points and each normal point is spanned by two arcs. If the two arcs spanning the same normal point have opposite orientations (one counterclockwise and the other clockwise), the normal point is called a *turning point*. The normal graph can be simplified to a *summary normal graph* in order to optimize the speed when checking the 2-moldability of the polygon. On the Gaussian circle, there are two kinds of points. For some points, the ray starting from the center of the Gaussian circle, passing through the point, and pointing to infinity intersects the normal graph only once, shown by a thin black line on the summary normal graph in Fig. 2(c); for other

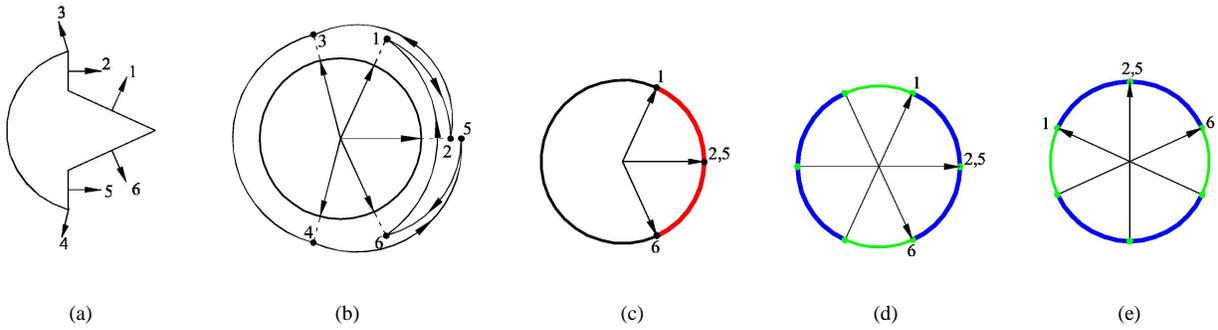


Figure 2. (a) A curved polygon; (b) normal graph (normal points and normal arcs expanded out from Gaussian circle for visualization purposes); (c) summary normal graph; (d) Gaussian circle is divided into regions; (e) Result in (d) is rotated 90° CCW (UFPDs in thin green, non-UFPDs in thick blue).

points, the ray intersects the normal graph more than once, shown by a thick red line in Fig. 2(c). Intersections at turning points do not count. The classification of all points on the Gaussian circle is shown on the summary normal graph (Fig. 2(c)).

In [19], we proved that if and only if a line L that passes through the center of the Gaussian circle intersects the normal graph exactly twice, the direction perpendicular to L is a UFPD. Thus in the figure, the two intersections of L with the summary normal graph, if it corresponds to a UFPD, will both have to be with the thin black portion corresponding to a single intersection with the original normal graph (see Fig. 2(c)). Lines where either intersection is with the portion shown in thick red means the direction perpendicular to L is a non-UFPD. Therefore, we can divide the Gaussian circle into regions within which L intersects the summary normal graph at exactly two thin black points (such regions are denoted by a thin green line in Fig. 2(d) and may be a single point), and regions within which L intersects the summary normal graph at one or more thick red points (such regions are denoted by a thick blue line in Fig. 2(d)). The regions are bounded by turning points or their reflections through the center of the circle. The result is then rotated around the Gaussian circle 90° to obtain the corresponding UFPDs. Thin green regions correspond to UFPDs and thick blue regions correspond to non-UFPDs (Fig. 2(e)).

5 FINDING ALL UFPDS FOR SOLIDS OF REVOLUTION

For a general polyhedron, a direction \vec{d} is a UFPD if and only if every line parallel to \vec{d} intersects the boundary of the polyhedron at most twice, where an intersection may be either a point or a line segment [12, 26].

Previous research for finding all UFPDs for a solid of revolution assumed that all UFPDs can be obtained by taking each edge of concavity features on its 2D generator profile and finding the intersection of associated hemi-circles on the Gaussian

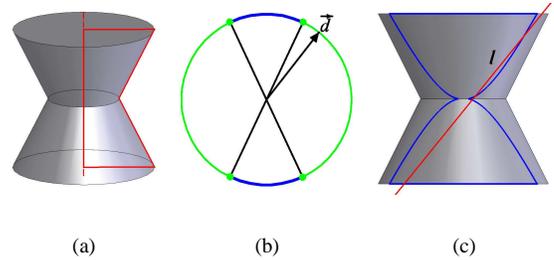


Figure 3. (a) A solid of revolution with 2D generator profile shown; (b) UFPDs parallel to the plane of the 2D generator profile, in thin green, assumed by [25]; (c) A line l parallel to \vec{d} intersects the boundary of a cross section of the solid four times, which makes \vec{d} a non-UFPD.

circle, where each edge defines a hemi-circle that is bounded by a line parallel to the edge through the circle center and that lies on the side of the line where the edge normal points. This intersection defines UFPDs for the 2D generator profile; it is then rotated around the pole axis on the Gaussian sphere [25]. An example is shown in Fig. 3(a), for which all proposed UFPDs parallel to the plane of the 2D generator profile are shown in Fig. 3(b) by thin green lines, with \vec{d} being an example UFPD. However, checking a hyperbolic conic section C parallel to the 2D generator profile shows that there is a line l parallel to \vec{d} that intersects the boundary of C , and hence the boundary of the solid, more than two times. Therefore, \vec{d} is not actually a UFPD. Next we will describe how we can correctly find all UFPDs for any solid of revolution.

In [24], we proved that a direction \vec{d} is a UFPD for a given polyhedron if and only if \vec{d} is a UFPD for all the cross sections in a family parallel to \vec{d} , where a family of cross sections are defined as an infinite number of cross sections that are parallel to each other. The particular orientation of a family of cross sections parallel to \vec{d} can be selected arbitrarily. However, since it is

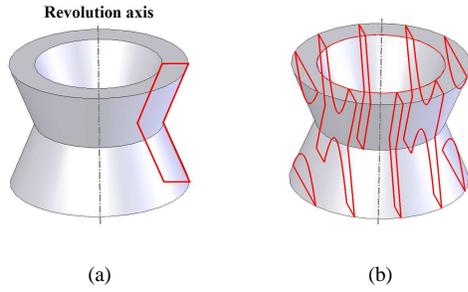


Figure 4. (a) A solid of revolution with 2D generator profile shown; (b) A family of cross sections parallel to the axis of revolution and a test direction that lies in the plane perpendicular to the 2D generator profile (only five cross sections are shown).

enough to test only one family of cross sections parallel to \vec{d} , for solids of revolution, we use vertical cross sections that are parallel to the axis of revolution (z axis) and the test direction (see Fig. 4).

Furthermore, the 2-moldability of a solid of revolution is circularly symmetric around the axis of revolution. If a direction is a UFPD, all directions having the same latitude on the Gaussian sphere are UFPDs and vice versa for non-UFPDs. Therefore, to test the 2-moldability of solids of revolution, we only need to test all the directions for 2-moldability that are parallel to one vertical plane. Without loss of generality, we choose the x - z plane and test the family of cross sections that are parallel to the x - z plane for 2-moldability, since a direction parallel to the x - z plane is a UFPD if and only if it is a UFPD for all cross sections in the family. After all UFPDs parallel to the x - z plane are obtained, the result is revolved around the pole axis on the Gaussian sphere to form all the UFPDs for the solid of revolution (for an example, see Fig. 5). The rest of the paper will focus on how to find all UFPDs that are parallel to the x - z plane. Unless otherwise specified, the test direction and the cross sections considered are parallel to the x - z plane for the remainder of the paper.

Note that the boundary of a cross section parallel to but not in the x - z plane for solids of revolution is composed of hyperbolic segments. To analyze the cross section, we first show some properties of hyperbolic conic sections.

5.1 Hyperbolic Conic Sections

We assume that the axis of a right circular cone is coincident with the z -axis. When intersected by a vertical plane parallel to the x - z plane, the cone produces a hyperbolic curve (see Fig. 6(a)). If the cone is trimmed at the bottom by a horizontal plane parallel to the x - y plane, the hyperbolic curve may also be trimmed (see Fig. 6(b)). The asymptotes of all hyperbolic conic sections that are parallel to the x - z plane are projected to the sil-

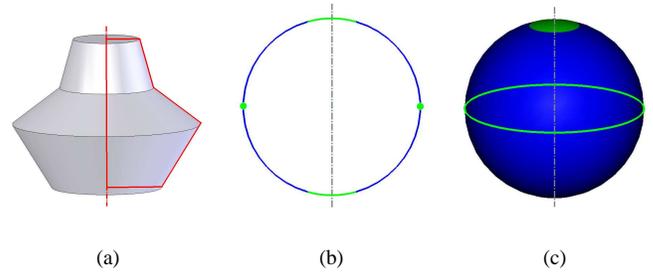


Figure 5. (a) A solid of revolution with its 2D generator profile outlined; (b) All directions parallel to the x - z plane with UFPDs in light green and non-UFPDs in dark blue; (c) Rotation of the result obtained in (b) on the Gaussian sphere.

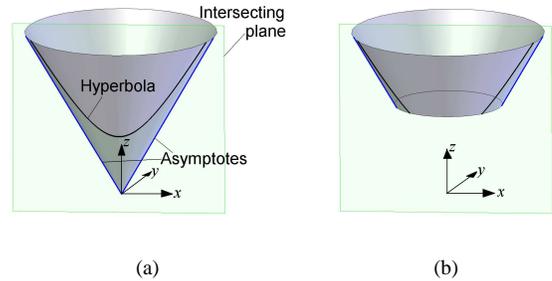


Figure 6. (a) conic section is a hyperbola; (b) hyperbola is trimmed by a horizontal plane.

houette lines of the cone on the x - z plane.

For a (possibly trimmed) cone, the following theorem holds. The full proof is deferred to the full version of the paper due to space restrictions.

Theorem 1. *Given a direction \vec{d} that is parallel to the x - z plane, if the smaller angle between the z axis and \vec{d} is larger than the angle between the z axis and the asymptotes of the hyperbolic conic sections, we can always find a line parallel to \vec{d} such that it intersects a hyperbolic conic section (and hence the conic surface) twice; otherwise, a line parallel to \vec{d} intersects any hyperbolic conic section at most once.*

5.2 2-Moldability of Solids of Revolution

In this section, we show that the 2-moldability of solids of revolution is constrained by the edges of the 2D generator profiles, which lie on the x - z plane.

Given a 2D generator profile for the solid of revolution, denote $E_{in} = \{e_{in}\}$ the set of edges on the 2D generator profile that have normal directions pointing inward to the axis of revolution, e_{in} not on the axis of revolution. For each e_{in} , denote θ_{in} the

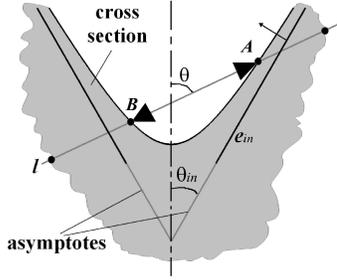


Figure 7. Line l intersects the solid of revolution at least four times.

smaller angle between e_{in} and the z axis. We have the following lemma.

Lemma 2. *If a direction \vec{d} , parallel to the x - z plane, is a UFPD for the solid of revolution, denoting θ the smaller angle between \vec{d} and the z axis, we have $\theta \leq \min(\theta_{in})$.*

Proof. For any edge $e_{in} \in E_{in}$, its revolution around the z axis is a (possibly trimmed) cone. Without loss of generality, suppose the normal direction of e_{in} points upward. We will prove that any direction \vec{d} with $\theta > \theta_{in}$ is not a UFPD for the solid of revolution.

From Theorem 1, for any direction \vec{d} with $\theta > \theta_{in}$, there always exists a line l parallel to \vec{d} intersecting twice with a hyperbolic conic section of the cone formed by e_{in} (Fig. 7). The line l goes upward into the interior of the solid at one intersection A and exits the solid along the same direction. It also goes downward into the interior of the solid at another intersection B and exits the solid along the same direction. With at least two entrances and two exits, l intersects the solid at least four times. Therefore, \vec{d} is not a UFPD for the solid of revolution.

In other words, if \vec{d} is a UFPD for the solid of revolution, we must have $\theta \leq \theta_{in}$ for all e_{in} , that is, $\theta \leq \min(\theta_{in})$.

Since all UFPDs must have $\theta \leq \theta_{in}$ for any e_{in} having a normal direction pointing inward to the axis of revolution, we say that UFPDs are *bounded by the revolution of e_{in}* . For edges $E_{out} = \{e_{out}\}$ with normal directions pointing away from the axis of revolution, a similar statement holds:

Lemma 3. *Given a UFPD \vec{d} for the solid of revolution and an edge e_{out} on the 2D generator profile, if one or both endpoints of e_{out} are concave, \vec{d} is either in a region of the Gaussian sphere bounded by the revolution of e_{out} or is a horizontal direction.*

Proof. To prove this lemma, we have three cases: only the end point of directed edge e_{out} is concave, only the start point of e_{out} is concave, or both the start point and the end point of e_{out} are concave. Since the proofs for these three cases are similar, we only provide the proof for the first case here to save space. Without loss of generality, we assume that the normal direction of e_{out} points upward.

With illustration in Fig. 8(a), we analyze the edge e_1 , whose end point is concave. Call the polygon containing e_1 , on the boundary of the central cross section, P . (This cross section consists of the generator profile and its reflection, so it may contain one or two disconnected polygons.) The local normal graph at e_1 for P is illustrated in Fig. 8(b), with normal arcs connecting n_0 , n_1 and n_2 , the normals of edges e_0 , e_1 and e_2 , respectively.

When the intersecting plane moves a little further away from the x - z plane, the boundary of the cross section contains a pair of symmetric hyperbolic segments corresponding to each edge on the 2D generator profile. Denote the curved polygon on this boundary, containing a set of connected hyperbolic segments, as P' . The local normal graph for P' has normal arcs connecting n_{0e} , n_{1s} , n_{1e} and n_{2s} (see Fig. 8(b)), where n_{is} and n_{ie} denote, respectively, the normals at the start point and end point of a hyperbolic segment on the cone formed by e_i . Projected onto the x - z plane, one of the asymptotes for the hyperbolic segment coincides with e_i . It can be shown from the equations of the hyperbolic segment and the line e_1 that n_{1s} and n_{1e} will always be closer to the $+z$ direction on the normal graph than n_1 , with n_{1e} even closer than n_{1s} . Since the end point of e_1 is concave, n_{1e} is closer to the $+z$ direction on the normal graph than n_{2s} . Therefore, the normal point n_{1e} is a turning point. On the summary normal graph, the arc between n_{1e} and n_{1s} or n_{2s} , whichever is closer to the z axis, will correspond to directions with more than one intersections between the ray starting from the center of the Gaussian circle and the normal graph. This arc, after rotating 90° counterclockwise, corresponds to a set of non-UFPDs (recall the example shown in Fig. 2). Note that the position of n_{1e} changes continuously from n_1 (at the central plane) to infinitely close to the $+z$ direction (when the hyperbolic conic section on the cone formed by e_2 shrinks until it disappears) as the intersecting plane moves away from the x - z plane. The whole arc on the summary normal graph between n_1 and the $+z$ direction therefore corresponds to non-UFPDs after rotation, not including the $+z$ direction (see Fig. 8(c)). Since all cross sections are symmetric relative to the z axis, the reflection of this arc about the z axis also corresponds to non-UFPDs, after rotation. The remaining part of the summary normal graph stays unclassified. In a summary, the UFPDs are bounded by the revolution of e_1 .

The proof is similar for the other two cases.

In this section, we have shown how to pre-identify a set of non-UFPDs using Lemma 2 and Lemma 3. The directions not in this set are *candidate UFPDs*. In the next section, we further classify these candidate UFPDs (as UFPDs or non-UFPDs) by analyzing three different cases.

5.3 Types of Solids of Revolution

We can classify solids of revolution into three categories (see Fig. 9), based on the properties of the closest element(s) (vertex or edge) on the 2D generator profile to the axis of revolution. The

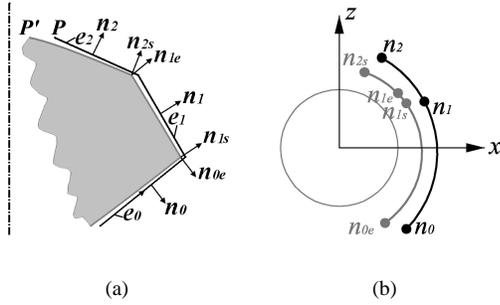


Figure 10. (a) Both endpoints of edge e_1 of the generator profile P are convex, a non-central cross section with boundary P' is shaded; (b) local normal graphs at e_1 (dark for P , light for P').

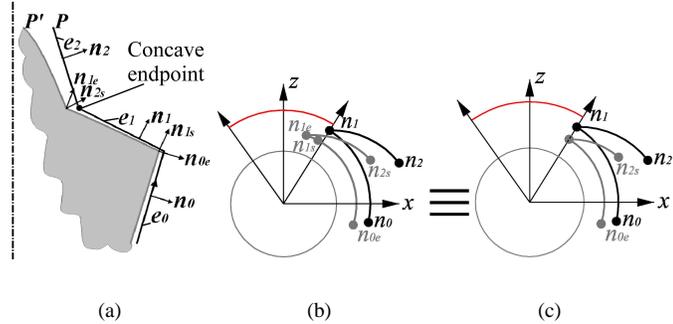


Figure 11. (a) Start point is convex and end point is concave for edge e_1 ; (b) Original local normal graphs at e_1 (dark for P , light for P'); (c) equivalent local normal graphs of (b).

When each edge e_i on P corresponds to a straight or hyperbolic segment on P' (that is, the number of edges on P equals the number of edges on P'), according to the convexity of the endpoints of e_i , we have four cases.

1. Both the start point and the end point of e_i are convex.
2. The start point of e_i is convex and the end point is concave.
3. The start point of e_i is concave and the end point is convex.
4. Both the start point and the end point of e_i are concave.

For the first case when both endpoints of e_i are convex, the local normal graphs at e_1 for both P and P' are shown in Fig. 10(b). It is obvious that the UFPDs for P and P' are equivalent since their summary normal graphs are identical locally around the normal point n_1 .

For the second case (Fig. 11(a)), the local normal graphs at e_1 for P and P' differ only in the pre-identified non-UFPD region (Fig. 11(b)), bounded by the symmetric arc around the z axis, which is derived according to Section 5.2. Since directions corresponding to this region are already classified as non-UFPDs, \vec{d}

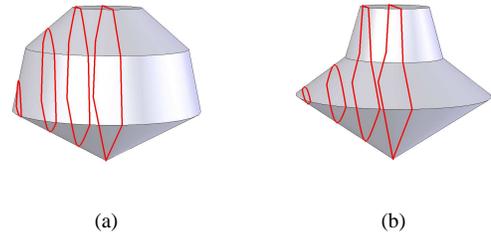


Figure 12. The boundary of the cross section has fewer edges as it moves away from the axis of revolution.

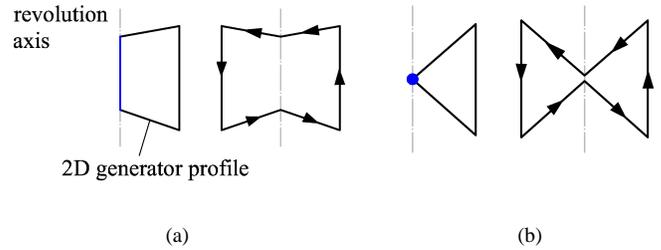


Figure 13. (a) Central cross section when the closest element is a single edge; (b) central cross section when the closest element is a single vertex (for purposes of visualization, the two coinciding vertices on the axis of revolution are pulled apart slightly).

is not in this region. Thus this portion of the normal graph can be ignored and simplified by moving the turning point n_{1e} onto the boundary of this region. The equivalent local normal graph at e_1 for P' , shown in Fig. 11(c), is thus the same as the local normal graph for P . Therefore, given that \vec{d} is a UFPD for P and a candidate UFPD for the solid, \vec{d} is also a UFPD for P' .

Case 3 and case 4 can be explained using the same logic.

When the number of edges of P' is smaller than that of P , the normal graph can be analyzed similarly. But there are fewer normal arcs on the normal graph for P' than for P since the hyperbolic segments corresponding to some edges on the 2D generator profile will disappear as the intersecting plane moves continuously away from the x - z plane (see Fig. 12). However, the same logic still applies. Therefore, we can use the boundary of the central cross section alone to further classify the candidate UFPDs.

The boundary polygon of the central cross section can be constructed directly from the 2D generator profile by taking the boundary of the union of the interiors of the 2D generator profile and its reflection, connectivity taken as shown in Fig. 13(b) when the closest element is a single vertex.

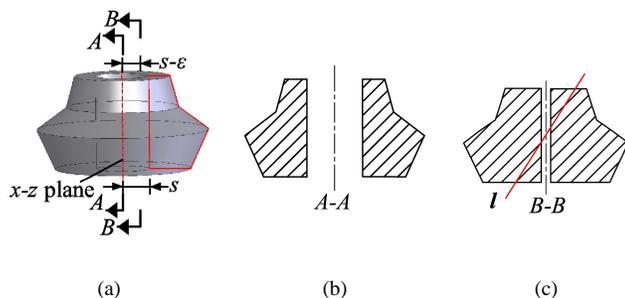


Figure 14. (a) An example of type II solid of revolution; (b) central cross section; (c) a cross section which is $s - \epsilon$ away from the x - z plane (line l is parallel to the test direction and intersects the cross section more than two times).

5.3.2 Type II. For type II revolutions, the closest element on the 2D generator profile to the axis of revolution is a single edge or vertex that does not lie on the axis (see Fig. 9(b)).

When the closest element is an edge e , the only possible UFPDs are the $+z$ and $-z$ directions. This can be easily proved. Suppose the distance between e and the axis of revolution is s . Consider the vertical intersecting plane that is $s - \epsilon$ away from the x - z plane, $\epsilon > 0$. When ϵ is small enough, for any given direction \vec{d} (except the $+z$ and $-z$ directions), there always exists a line l parallel to \vec{d} intersecting the boundary of the cross section more than two times (see Fig. 14).

We now analyze the only remaining candidate UFPDs, i.e., the $+z$ and $-z$ directions, and show how they can be tested by converting the geometry to a type I revolution with equivalent UFPDs.

Suppose V is a point on e , not necessarily one of its endpoints. Call its projection onto the axis of revolution O (see Fig. 15(a)). As we prove below, inserting two oppositely oriented edges VO and OV into the 2D generator profile does not affect the 2-moldability of the solid of revolution in the $+z$ and $-z$ directions. The additional two edges are revolved to form two disks, with normals pointing in the $+z$ and $-z$ directions, respectively. Since e is the closest element on the 2D generator profile to the axis of revolution, these two disks are neither occluded by nor occlude *any* part of the original solid of revolution, when viewed from the $+z$ and $-z$ directions, respectively. Hence they do not add new undercuts or change existing undercuts. That is, the 2-moldability of the solid of revolution in the $+z$ and $-z$ directions is not changed. Directions other than the $+z$ and $-z$ directions remain non-UFPDs.

If we weld the two new edges at their open vertices, the 2D generator profile becomes that for a type I revolution. As is proved in the above section, the boundary of the central cross section can be easily constructed and uniquely determines the 2-moldability of the solid of revolution (see Fig. 15(a)). Note

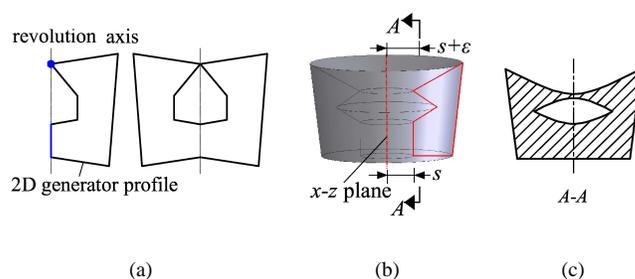


Figure 16. Type III solids of revolution. (a) Closest elements lie on the axis of revolution; (b) closest elements does not lie on the axis of revolution; (c) a cross section, which is $s + \epsilon$ away from the x - z plane, has a hole.

that directions other than the $+z$ and $-z$ directions are obviously non-UFPDs for the constructed central cross section.

When the closest element is a vertex V , suppose the projection of V on the axis is O (see Fig. 15(b)). Similarly, adding two edges VO and OV (see Fig. 15(b)) to the 2D generator profile does not change the 2-moldability of the solid of revolution. That is, up-facets (respectively, down-facets) remain up-facets (respectively, down-facets), no new undercuts are added, and no existing undercuts are affected, relative to a given candidate UFPD (the pre-identified non-UFPDs remain non-UFPDs). Thus we again convert the geometry to a type I revolution with equivalent UFPDs.

5.3.3 Type III. For type III solids of revolution, there are multiple closest elements (edges or vertices) on the 2D generator profile to the axis of revolution.

When the closest elements lie on the axis, no directions are UFPDs for the solid of revolution since one or more holes are present on the central cross section (see Fig. 16(a) for an example). No directions are UFPDs for the central cross section and hence for the solid of revolution.

When these multiple closest elements do not lie on the axis of revolution, suppose the distance of the closest elements to the axis of revolution is s . Consider the intersecting plane that is $s + \epsilon$ away from the x - z plane, $\epsilon > 0$. When ϵ is not too large, the cross section has a hole in the middle (see Fig. 16(c)). Therefore, no directions are UFPDs for this cross section and hence for the solid of revolution.

In summary, type III solids of revolution are not 2-moldable in any direction.

5.4 Summary and Algorithm

From the above discussion, we can conclude that the 2-moldability analysis on a solid of revolution can be obtained by

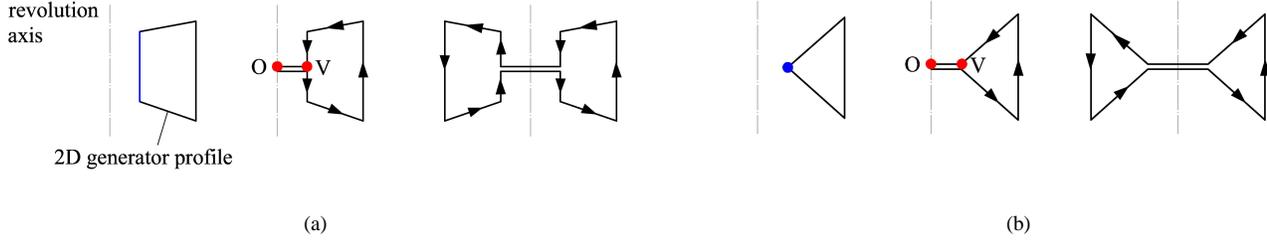


Figure 15. Type II solid of revolution. (a) Modified central cross section when the closest element is a single edge; (b) modified central cross section when the closest element is a single vertex. For purposes of visualization, coinciding edges are pulled apart slightly.

```

Input :  $P_g$ , 2D generator profile on the  $x$ - $z$  plane;
           $z$ , axis of revolution.
Output : Set of UFPDs  $\Phi$  parallel to the  $x$ - $z$  plane.

Traveling around  $P_g$  to find all the closest elements  $E_c$ .
if  $\text{sizeof}(E_c) > 1$  then
    Solid of revolution is type III.
    Return  $\Phi = \text{null}$ .
else
    Solid of revolution is type I or II.
     $\Phi_1 = 90^\circ$ .
    foreach edge  $e_i$  on  $P_g$ , if normal of  $e_i$  points toward
     $z$  or at least one endpoint of  $e_i$  is concave and
    normal of  $e_i$  points away from  $z$  do
         $\theta =$  angle between  $e_i$  and  $z$  axis.
         $\Phi_1 = \min(\Phi_1, \theta)$ .
    end
    Construct polygon  $P$  according to Fig. 13 (type I)
    or Fig. 15 (type II).
    Find all UFPDs  $\Phi_2$  for  $P$  using 2D algorithm.
    if no edge on  $P_g$  has a normal pointing toward  $z$ 
    then
        Return  $\Phi = ([90^\circ - \Phi_1, 90^\circ + \Phi_1] \cup [270^\circ - \Phi_1, 270^\circ + \Phi_1]) \cap \Phi_2$ .
    else
        Return  $\Phi = ([90^\circ - \Phi_1, 90^\circ + \Phi_1] \cup [270^\circ - \Phi_1, 270^\circ + \Phi_1] \cup [0^\circ, 0^\circ] \cup [180^\circ, 180^\circ]) \cap \Phi_2$ .
    end
end

```

Algorithm 1: FindingUFPDsForRevolution()

performing analysis on its 2D generator profile only. The algorithm is summarized in the Algorithm 1 box.

In our previous work, we proved that finding all UFPDs for a polygon P takes $O(n)$ time [19]. Extracting all the closest elements and finding the minimum Φ_1 also takes $O(n)$ time. Therefore, the overall algorithm for finding all UFPDs for a solid of

revolution has a time complexity of $O(n)$.

6 CONCLUSIONS

Finding *all* UFPDs at interactive speeds gives designers maximum flexibility choosing a parting direction early in the design process, when redesign cost is the lowest. Existing approaches either cannot find all UFPDs (heuristic approaches) or cannot run at interactive speeds for anything beyond relatively simple part geometries (exhaustive approaches). We have proposed a new feature-based approach to reduce the time to find all UFPDs, taking advantage of feature-based CAD systems. Since UFPDs are constrained by each individual feature composing the geometry, only when a direction is a UFPD for all features, is it a potential UFPD for the overall geometry. Thus only these directions need to be tested using an exhaustive algorithm; otherwise, the direction is directly classified as a non-UFPD for the overall geometry.

In this paper, we showed how to find all UFPDs for a solid of revolution via analyzing its 2D generator profile in $O(n)$ time. These UFPDs were previously found correctly only by exhaustive algorithms for arbitrary parts that took at least $O(n^4)$ time. Since revolved features often greatly constrain the potential UFPDs for the overall geometry, the test space on the Gaussian sphere becomes significantly smaller than the entire Gaussian sphere that previously had to be tested. If no directions are UFPDs for all revolved features, the part can be immediately identified as non-2-moldable without further testing; in this case, designers can either go back to redesign the part geometry or choose an optimal parting direction for a mold with more than two pieces based on other criteria such as the number of undercuts or undercut volume.

Some people argue that designers usually use directions that are aligned with or normal to revolution axis directions as parting directions when revolved features exist. While this is a fact in current practice, it is because these are the directions whose 2-moldability a human designer can most easily evaluate. Our algorithm automatically provides designers with more alternative

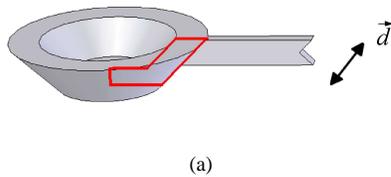


Figure 17. An example containing a revolved feature, with the 2D generator profile highlighted. The only UFPD for the example geometry is \vec{d} , which is neither the revolution axis direction nor one of its normal directions.

UFPDs to choose from. With candidate parting directions predefined as revolution axis directions or their normal directions, either the design innovation and alternatives are limited or the manufacturing cost rises because of the undercuts. Fig. 17 shows an example containing a revolved feature. The geometry would cause undercuts if the parting direction is defined as the revolution axis direction or one of its normal directions. Our algorithm finds the UFPD, which enables the part to be manufactured at a lower cost.

Whether undercuts exist is not the only criteria when choosing an optimal parting direction for a complex geometry. Other factors, such as the complexity of the parting surface, also play an important role [1,27]. Generally UFPDs are the preferred parting directions. But designers and manufacturers may choose non-UFPDs with planar parting surfaces instead if all UFPDs require complex non-planar parting surfaces. Our future work aims to define optimal parting directions for an arbitrary geometry based on multiple criteria as well as exploring 2-moldability analysis on more complex feature types such as sweeps and lofts.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 0547675.

REFERENCES

[1] Ravi, B., and Srinivasan, M. N., 1990. "Decision criteria for computer-aided parting surface design". *Computer-Aided Design*, **22**(1), pp. 11–18.

[2] Wong, T., Tan, S. T., and Sze, W. S., 1998. "Parting line formation by slicing a 3D CAD model". *Engineering with Computers*, **14**(4), pp. 330–343.

[3] Chen, Y. H., 1997. "Determining parting direction based on minimum bounding box and fuzzy logics". *Int. J. Mach. Tools Manufact.*, **37**(9), pp. 1189–1199.

[4] Hui, K. C., and Tan, S. T., 1992. "Mould design with sweep operations - a heuristic search approach". *Computer-Aided Design*, **24**(2), February, pp. 81–91.

[5] Hui, K. C., 1997. "Geometric aspects of the mouldability of parts". *Computer-Aided Design*, **29**(3), pp. 197–208.

[6] Chen, L.-L., Chou, S.-Y., and Woo, T. C., 1993. "Parting directions for mould and die design". *Computer-Aided Design*, **25**(12), December, pp. 762–768.

[7] Woo, T. C., 1994. "Visibility maps and spherical algorithms". *Computer-Aided Design*, **26**(1), January, pp. 6–16.

[8] Chen, L.-L., and Chou, S.-Y., 1995. "Partial Visibility for Selecting a Parting Direction in Mold and Die Design". *Journal of Manufacturing Systems*, **14**(5), pp. 319–330.

[9] Weinstein, M., and Manoochehri, S., 1996. "Geometric Influence of a Molded Part on the Draw Direction Range and Parting Line Locations". *Journal of Mechanical Design*, **118**(3), March, pp. 29–39.

[10] Wuerger, D., and Gadh, R., 1997. "Virtual prototyping of die design. Part one: Theory and formulation". *Concurrent Engineering : Research and Applications*, **5**(4), December, pp. 307–315.

[11] Wuerger, D., and Gadh, R., 1997. "Virtual Prototyping of Die Design. Part Two: Algorithmic, Computational, and Practical Considerations". *Concurrent Engineering : Research and Applications*, **5**(4), December, pp. 317–326.

[12] Ha, J., Yoo, K., and Hahn, J., 2006. "Characterization of polyhedron monotonicity". *Computer-Aided Design*, **38**(1), pp. 48–54.

[13] Ganter, M. A., and Skoglund, P. A., 1991. "Feature extraction for casting core development". In 17th Design Automation Conference presented at the 1991 ASME Design Technical Conferences, American Society of Mechanical Engineers, pp. p 93–100.

[14] Fu, M. W., Fuh, J. Y. H., and Nee, A. Y. C., 1999. "Generation of optimal parting direction based on undercut features in injection molded parts". *IIE Transactions*, **31**(10), pp. 947–955.

[15] Fu, M. W., Fuh, J. Y. H., and Nee, A. Y. C., 1999. "Undercut feature recognition in an injection mould design system". *Computer-Aided Design*, **31**(12), October, pp. 777–790.

[16] Ye, X. G., Fuh, J. Y. H., and Lee, K. S., 2001. "A hybrid method for recognition of undercut features from moulded parts". *Computer-Aided Design*, **33**(14), pp. 1023–1034.

[17] Yin, Z., Ding, H., and Xiong, Y., 2001. "Virtual prototyping of mold design: geometric mouldability analysis for near-net-shape manufactured parts by feature recognition and geometric reasoning". *Computer-Aided Design*, **33**(2), pp. 137–154.

[18] Rappaport, D., and Rosenbloom, A., 1994. "Moldable and castable polygons". *Computational Geometry: Theory and Applications*, **4**(4), pp. 219–233.

[19] McMains, S., and Chen, X., 2006. "Finding undercut-free parting directions for polygons with curved edges". *ASME Journal of Computing and Information Science in Engi-*

- neering*, **6**(1), March, pp. 60–68.
- [20] Bose, P., and Bremner, D., 1997. “Determining the Castability of Simple Polyhedra”. *Algorithmica*, **17**(1-2), pp. 84–113.
 - [21] Elber, G., Chen, X., and Cohen, E., 2005. “Mold Accessibility via Gauss Map Analysis”. *Journal of Computing and Information Science in Engineering*, **5**(2), pp. 79–85.
 - [22] Dhaliwal, S., Gupta, S., Huang, J., and Priyadarshi, A., 2003. “Algorithms for computing global accessibility cones”. *Journal of Computing and Information Science in Engineering*, **3**(3), pp. 200–209.
 - [23] Khardekar, R., Burton, G., and McMains, S., 2006. “Finding Feasible Mold Parting Directions Using Graphics Hardware”. *Computer-Aided Design*, **38**(4), pp. 327–341.
 - [24] Chen, X., and McMains, S., 2006. “Finding All Undercut-Free Parting Directions for Extrusions”. In *Geometric Modeling and Processing*, LNCS 4077, Springer-Verlag, pp. 514–527.
 - [25] Kurth, G. R., and Gadh, R., 1997. “Virtual prototyping of die-design: determination of die-open directions for near-net-shape manufactured parts with extruded or rotational features”. *Computer Integrated Manufacturing System*, **10**(1), pp. 69–81.
 - [26] Ahn, H.-K., de Berg, M., Bose, P., Cheng, S.-W., Halperin, D., Matousek, J., and Schwarzkopf, O., 2002. “Separating an object from its cast”. *Computer-Aided Design*, **34**(8), pp. 547–59.
 - [27] Boothroyd, G., Dewhurst, P., and Knight, W., 2002. *Product Design for Manufacture and Assembly, Second Edition*. Marcel Dekker.