Mold Accessibility via Gauss Map Analysis[*]

Gershon Elber[†]
Computer Science Department, Technion, Haifa 32000, Israel

Xianming Chen[‡], Elaine Cohen
Computer Science Department, University of Utah, SLC 84112, Utah

Abstract

In manufacturing processes like injection molding or die casting, a 2-piece mold is required to be separable, that is, be able to have both pieces of the mold remove in opposite directions while interfering neither with the mold nor with each other.

The fundamental problem is to find a viewing (i.e. separating) direction, from which a valid partition line (i.e. the contact curves of the two mold pieces) exists. While previous research work on this problem exists for polyhedral models, verifying and finding such a partition line for general freeform shapes, represented by NURBS surfaces, is still an open question.

This paper shows that such a valid partition exists for a compact surface of genus g, if and only if there is a viewing direction from which the silhouette consists of exactly $g+1$ non-singular disjoint loops. Hence, the 2-piece mold separability problem is essentially reduced to the topological analysis of silhouettes. In addition we deal with removing almost vertical surface regions from the mold so that the form can more easily be extracted from the mold.

It follows that the aspect graph, which gives all topologically distinct silhouettes, allows one to determine the existence of a valid partition as well as to find such a partition when it exists. In this paper, we present an aspect graph computation technique for compact free-form objects represented as NURBS surfaces. All the vision event curves (parabolic curves, flecnodal curves, and bitangency curves) relevant to mold separability are computed by symbolic techniques based on the NURBS representation, combined with numerical processing. An image dilation technique is then used for robust aspect graph cell decomposition on the sphere of viewing directions. Thus, an exact solution to the 2-piece mold separability problem is given for such models.

Keywords*: Mold separability, silhouette generator, silhouette, aspect graph, vision event, visibility*

1 Introduction

Design-for-manufacturing necessitates that a CAD system be enhanced with the capability to verify that the designed model can be manufactured. In this paper, we consider the problem of determining the existence of and finding (if one exists) a valid 2-piece mold for a designed solid model, whose boundary is represented as NURBS

Most papers on 2-piece mold separability problem (e.g. [16, 5, 1]) work on polyhedral models. Typically, they select some heuristic directions from which to verify separability. Thus, the algorithms are not complete in the sense that they only verify but cannot find a valid 2-piece mold even though one might exist. In [1], a complete algorithm is presented, but for implementation it falls back to verifying some heuristically selected directions because of the high complexity of the complete algorithm.

In this paper, we use models bounded by NURBS surfaces that have $C^{(3)}$ continuity, present and implement a complete algorithm for the mold separability problem. $C^{(3)}$ is required only because relevant aspect graph computation requires third derivatives.

A 2-piece mold, denoted $2\mathcal{PM}$ hereafter, is used in manufacturing processes such as injection molding or die casting [4, 13]. In general,

Definition 1 *An object that is manufacturable with the aid of an n-piece mold is denoted by*

[*] This work was supported in part by ARO (DAAD19-01-1-0013), NIH (571495), NSF (EIA0121533), and NSF (CCR0310705). All opinions, findings, conclusions or recommendations expressed in this document are those of the author and do not necessarily reflect the views of the sponsoring agencies.

[†] The research was supported in part by the Fund for Promotion of Research at the Technion, IIT, Haifa, Israel.

[‡] Readers may contact the authors through Xianming Chen, Department of Computer Science, University of Utah, Salt Lake City, Utah 84112, (801)585-6161, xchen@cs.utah.edu.

$n\mathcal{PM}$. The curves along which the object is partitioned are called partition line*, and denoted by $\mathcal{PL}$.*

Specifically in this paper, we consider $2\mathcal{PM}$, and require that there be an *opposing* separating direction V for the two pieces of the mold.

Obviously the existence of a $2\mathcal{PM}$ for a solid model is completely determined by its boundary surface. The boundary surface must be closed and bounded, i.e. compact.

If the the separating direction V and the the boundary surface S are to be emphasized, we will use the notation $2\mathcal{PM}_S^V$.

Note we assume a connected boundary surface for any solid model (in fact, the connectedness is usually implicit in the mathematical definition of a surface), and hence we do not consider void objects (such as a hollowed sphere), whose inner and outer boundary surfaces are not connected. This assumption does not make our discussion any less general, since void objects obviously cannot have a valid $2\mathcal{PM}$.

As silhouette generators and silhouettes are our starting point to solve the $2\mathcal{PM}$ problem, we give their definition here.

Definition 2 *Given a viewing direction V, and a surface S,*

1. *The* silhouette generator $\mathcal{G}_S^V$ *is the locus of points on S, whose surface normals are perpendicular to V.*
2. *The corresponding* silhouette $\mathcal{S}_S^V$ *is the projection of the silhouette generator along the direction V onto an arbitrary image plane.*

The rest of the paper is organized as follow. In Section 2, we give the necessary background of vision events and aspect graphs. In Section 3, the relations between mold accessibility and, silhouettes and visibility, are investigated. In Section 4, a complete set of algorithms for computing the $2\mathcal{PM}$ separability for NURBS surfaces is given. The implementation uses a combination of symbolic and numeric computation, as well as a technique from image processing. Some examples are given in Section 5. Also for practical manufacturing, Section 6 computes the partition line itself given the view direction, shows how to prescribing a relief angle, and further shows how to redesign the *difficult to remove* regions . Finally, the paper concludes in Section 7.

2 Background

In this section we briefly review relevant basic terms and results about vision events and aspect graphs, which are essential to the two theorems and the algorithms of this paper. Refer to Koenderink [17] for more detail. For readers familiar with differential geometry and singularity theory, this section can be skipped safely.

1. Let S be the surface, p be any point on S, and n_p the surface unit normal at p. The *Gauss Mapping* is defined as $\mathcal{N} : S \longrightarrow \mathscr{S}^2, \mathcal{N}(p) = n_p$,

   Here, $\mathscr{S}^2$ denotes the manifold of all unit surface normals, and is called the Gauss Sphere. Later, $\mathscr{S}^2$ will also be used to represent the manifold of unit orthogonal viewing directions, and consequently is also called the viewing sphere.

   Every viewing direction $V \in \mathscr{S}^2$ prescribes a great circle C on $\mathscr{S}^2$ orthogonal to V. Then,

   $$\mathcal{G}_S^V = \{p \in S \mid \mathcal{N}(p) \in C\}.$$

2. A *parabolic curve* is defined as the locus of points on S with zero Gaussian curvature, i.e., one of the principal curvature being zero. The zero principal curvature direction is necessarily an *asymptotic direction*, since the second fundamental form is 0 (i.e. self-conjugate), or, intuitively the tangent line at that direction has order 3 contact with the surface. The asymptotic directions on a parabolic curve generate a developable called *axis cylinder developable*.

3. Parabolic curves separate the elliptic from the hyperbolic regions of the surface. There are 0, 1, and 2 asymptotic directions on each point of elliptic surface regions, parabolic curves, and hyperbolic surface regions, respectively. At a hyperbolic surface region, there are two asymptotic direction(vector) fields, the integral of which are two families of *asymptotic curves*. The locus of all the geodesic inflection points on these two families of asymptotic curves gives two *flecnodal curves*. At these geodesic inflection points, there are order-4 contact asymptotic directions, which generate two non-developable ruled surfaces(with either of the flecnodal curve as the base curve), called *flecnodal scrolls*.

4. A *bi-tangent* ray is a line tangent to the surface at two different points. If the line also lies in the plane that is tangent to the surface points, it is then a *limiting bi-tangent ray*, the locus of which is a *bi-tangent developable*, contacting the surface at a *bi-tangency curve*.

5. For all of these three ruled surfaces, translating their generating lines to the origin gives the corresponding *generating cones*.

6. For orthographic projection, the manifold of viewing directions can be represented as the unit sphere surface $\mathscr{S}^2$. $\mathscr{S}^2$ can then be partitioned (by the various generating cones) into regions, with the property that the silhouettes of all the viewing directions in any one region have the same topology. When the viewing direction crosses the boundary of a topologically equivalent region on $\mathscr{S}^2$ into anther one, the topology of the silhouette suddenly changes, i.e., a *vision event* occurs. A graph whose nodes are the representative silhouettes of all the partitioned regions on $\mathscr{S}^2$, and whose edges are the vision events between them, is called an *aspect graph*.

7. A point on a curve is called an *ordinary cusp* if locally the curve can be represented as $y^2 = x^3$ after an appropriate coordinate transformation. In this paper, all cusps considered are ordinary, and are simply called cusps.

8. It may happen that there is a cross on the silhouette, either from one loop or two distinct loops. For an opaque object, part of one of the four branches is occluded, and hence the cross appears as a T-shape, called a *T-junction*.

Aspect graphs play a major role in many geometric applications, most noticeably in computer vision and recognition. These graphs capture vision events of topological changes in the object appearance as the view direction is changed. A complete understanding of aspect graph requires differential geometry and singularity theory, both of which are well-established [19, 17, 2, 3]. The implementation, however, is not as satisfactory even after three decades of research and development. Earlier research has been conducted mainly on polyhedral models [8, 9, 10] or some specific kind of surfaces, such as quadratic surfaces or surfaces of revolution [18]. Petitjean [21] computed aspect graphs of smooth algebraic implicit surfaces.

In this paper, we implement a set of algorithms to compute the aspect graph of a freeform object represented as NURBS surfaces. The computed aspect graph is almost complete, except we do not compute cusp-crossing and triple-point vision events, which are irrelevant to the mold separability problem. See Section 3. Symbolic computation is used extensively, with the help of numeric computation. This differentiates our approach from previous methods. We also use a different method based on image dilation to decompose viewing sphere into topologically equivalent maximal regions. See details in Section 4.

We now give some relevant results from singularity theory on aspect graph with brief explanation.

1. A silhouette generator $\mathcal{G}_S^V$ generally consists of one or more closed *regular* space curves on S, except when V is the asymptotic direction of a *parabolic point* on S. Then $\mathcal{S}_S^V$ has singularity of either an intersection (or cross) or an isolated point. The singularities are *unstable*, i.e., they disappear under small perturbation. This is because
$$\mathcal{G}_S^V = \{p \in S \mid \mathcal{N}(p) \in C\},$$
(where C is the great circle orthogonal to V), and $\mathcal{N}$ is a locally differmorphism except at parabolic points.

2. The viewing direction V and tangent direction T to $\mathcal{G}_S^V$ are conjugate, i.e., their second fundamental form is zero, $II(V,T) = 0$ [6, pp. 61-62].

   This simple result is actually all we need to completely understand any kind of cusp singularities.

3. The only singularities that a silhouette $\mathcal{S}_S^V$ can have consists of cusps and T-junctions, both of which are stable. A cusp occurs when V is the asymptotic direction at the cusp's pre-image on S. A T-junction occurs when V is tangent to two distinct points on S

   When V is the asymptotic direction, which is self-conjugate, i.e., $II(V,V) = 0$, we are necessarily looking at the tangent direction of the silhouette generator (cf. 2), and $\mathcal{S}_S^V$ will potentially have a cusp.

   Vision events are closely related to parabolic curves, flecnodal curves and bi-tangency curves on the surface, or more precisely to their corresponding asymptotic ray manifolds or bi-tangent ray manifolds. There are two other multi-tangent ray manifolds related to vision events (see cusp-cross and triple-point events below). For mold separability problems, however, they can be safely ignored.

4. There are 6 types of vision events, 3 local and 3 multi-local (Figure 1 and Figure 2).

   (a) A *lip event* occurs when the viewing direction crosses the axis cylinder developable at a parabolic point of elliptic type [17, pp. 297-303]. In this case, 2 cusps (dis)appear.

   (b) A *beak-to-beak events* occurs when the viewing direction crosses the axis cylinder developable at a parabolic point of hyperbolic type [17, pp. 297-303]. In this case, 2 cusps (dis)appear.

   (c) A *swallow tail event* occurs when the viewing direction crosses the flecnodal scroll. Again 2 cusps (dis)appear, and in addition, a T-junction (dis)appears.

   (d) A *tangent crossing event* occurs when the viewing direction crosses the limiting bi-tangent ray manifold, In that case, 2 segments of silhouette approach, touch, and then separate again.

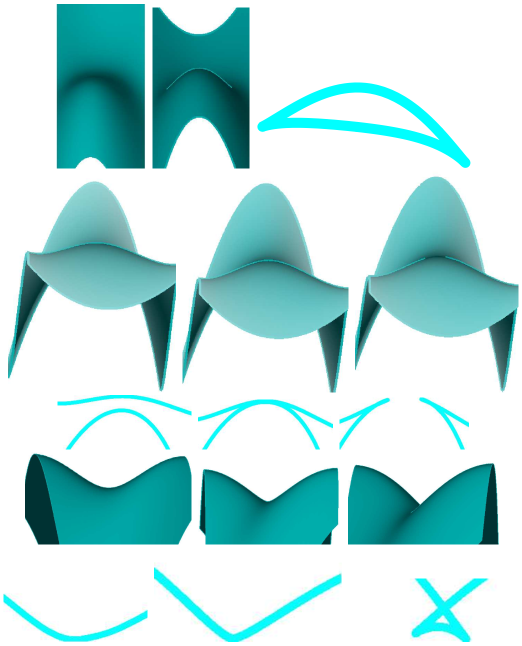


Figure 1. Local Vision Events under orthographic projection. *Row 1 shows a lip event at a parabolic point of elliptic type. The left and middle images are before and after the vision event. The right image is a closeup silhouette corresponding to the middle image. Row 2&3 show a beak-to-beak event at a parabolic point of hyperbolic type. Row 4&5 show a swallow-tail event at a flecnodal point. From left to right, each of these four rows shows images before, at and after the vision event respectively.*

(e) A *cusp crossing event* occurs when the viewing direction crosses the manifold of those bi-tangent rays, which happen to be the asymptotes at one of the two surface points

(f) A *triple point event* occurs when the viewing direction crosses tri-tangent(touching 3 surface points at the same time) ray manifold.

3 Mold Accessibility And Silhouettes & Visibility

For a generic compact surface S, viewed from direction V, its silhouette generator $\mathcal{G}_S^V$ is formed out of one or more disjoint closed regular curves. Genericity is a mathematical concept of openness, denseness and transversality. In the context of this paper, a generic surface is simply a surface whose silhouette topology, under any sufficiently small perturbation of the surface, does not change.

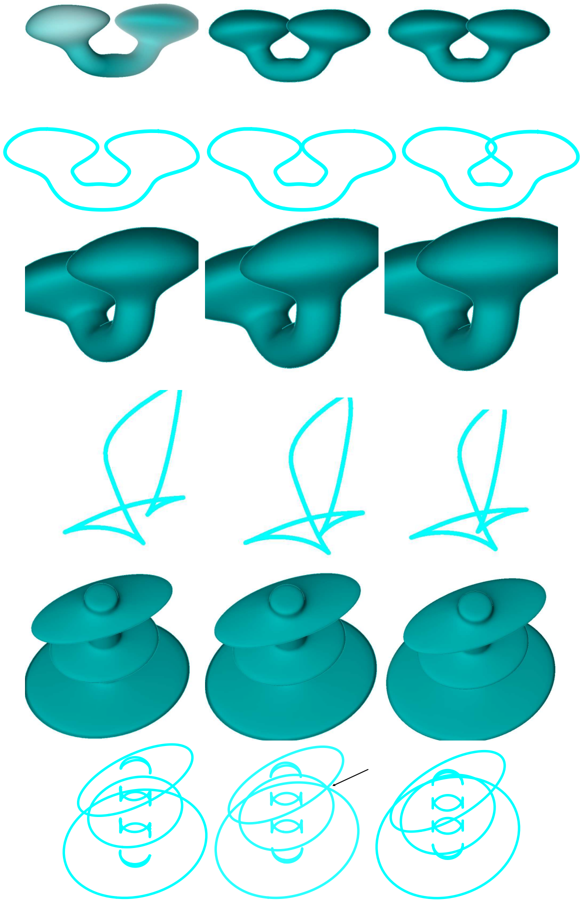


Figure 2. Multi-Local Vision Events under orthographic projection. *From left to right, Each row shows images before, at and after the vision event respectively. Row 1&2 show a bi-tangent cross event. Row 3&4 show a cusp-cross event. Row 5&6 show a triple-point event. The arrow on the middle image of the last row shows where the triple-point event occurs.*

$2\mathcal{PM}$ manufacturability is closely related to the visibility and silhouette processing problem, and is reduced to them by the following theorem;

Theorem 1 *A compact surface S of genus-g has a valid $2\mathcal{PM}_S^V$ if and only if the silhouette generator $\mathcal{G}_S^V$ has exactly $g+1$ loops, each of which is totally visible from the direction V.*

Intuitively genus can be understood as the number of handles or through-holes in a surface.

Proof:

(⇒). If the surface has a valid $2\mathcal{PM}$, then the two pieces can be separated in opposite V direction, interfering neither with each other nor with the molded object inside. In other words, if a ray parallel to V is emitted from a point p on the surface facing the viewer, the ray is not occluded by any other surface points, and so p is visible. Specifically the partition line $\mathcal{PL}$ is visible from V. Furthermore, $\mathcal{PL}$ is identified with $\mathcal{G}_S^V$, because,

1. $\mathcal{PL} \subset \mathcal{G}_S^V$. $\mathcal{PL}$ is visible from V as shown above. $\mathcal{PL}$ is also visible from $-V$ by exactly the same reason.

2. $\mathcal{G}_S^V \subset \mathcal{PL}$. If not, there would be added silhouette generator segments. Consequently there would be a back-facing region and overlap at least one piece, so pulling this piece away would necessarily interfere with the molded object inside, hence a contradiction.

Now we need to prove, $\mathcal{G}_S^V$ has exactly $g+1$ loops. This is readily proved by considering the fact that the surface has g through-holes, and that $\mathcal{PL}$ is the common contact of the two pieces. First, if it has more than $g+1$ loops, then each additional loop would necessarily introduce a through-hole, contradicting the hypothesis of genus g. On the other hand, the silhouette generator cannot have fewer than $g+1$ loops either; if it did, there would exist one through-hole completely embedded in only one of the two mold pieces, which is impossible.

(⇐). This can be most readily proved if we consider constructing a viewing ray cylinder for each loop on $\mathcal{G}_S^V$. There are $g+1$ such cylinders, all with V as the generator, but with distinct loops of $\mathcal{G}_S^V$ as their distinct base curves.

Because all loops on $\mathcal{G}_S^V$ are totally visible, it is obvious no pair of such cylinder surfaces can intersect. Considering further that the model surface is connected, the only possible layout of these cylinders is that there is one outer cylinder that completely contains the remaining g cylinders, none of which is inside any of themselves. Actually these g cylinders exactly correspond to the g through-holes of the model surface which is of genus g.

Therefore, all the $g+1$ loops on $\mathcal{G}_S^V$ actually only cut the model into two pieces.

Further, each piece must be totally visible. If not, there would exist some silhouette generator segment between the invisible back-facing area and the visible front-facing area. This segment is necessarily invisible, which contradicts the hypothesis that the silhouette generator is totally visible.

Finally, complete visibility of the two pieces ensures the successful separating along direction V. ■

Theorem 1 is powerful. But to verify the visibility of the silhouette generator directly is quite difficult. The following theorem gives us a relation between the visibility of a silhouette generator and the singularity of its corresponding silhouette. The singularity of the silhouette, in turn, can be readily predicted by the aspect graph.

> **Theorem 2** *If S is a compact surface with genus-g , then $\mathcal{G}_S^V$ has $g+1$ loops, each of which is totally visible, if and only if $\mathcal{S}_S^V$ has $g+1$ loops, each of which is non-singular.*

To prove this theorem, we need the concept of contact order of a tangent line touching the surface [6, 17]. Intuitively, a line touches surface with order n if, under slight perturbation, it will possibly intersect the surface at n but no more distinct points.

Proof:

(⇒). $\mathcal{G}_S^V$ with $g+1$ loops is projected onto $\mathcal{S}_S^V$ with $g+1$ loops, each of which must be free of singularity, because both cusp and T-junction singularities imply occlusion of $\mathcal{G}_S^V$ (see Section 2).

(⇐). $\mathcal{G}_S^V$ can not have fewer than $g+1$ loops. On the other hand, if it has more than $g+1$ loops, then 2 loops must project onto the same loop on $\mathcal{S}_S^V$, which is the singular case of a degenerate T-junction. So $\mathcal{G}_S^V$ has exactly $g+1$ loops.

Second, $\mathcal{G}_S^V$ must be totally visible. If not,

1. either there would be a non-local occlusion of $\mathcal{G}_S^V$. In this case, $\mathcal{S}_S^V$ would have a T-junction, a contradiction.

2. or there would be a local occlusion of $\mathcal{G}_S^V$. At this local point, say p, a visible silhouette generator segment and an invisible silhouette generator segment meet. The visible segment has normal 2-order contact with the surface. The invisible segment locally has normal 2-order contact. However it will pierce the surface at an extra point since it is invisible. Approaching p along the invisible segment, this extra point merges with p and the visual ray will have 3-order contact there, i.e. it is an asymptotic direction. Therefore, there must be a cusp on the silhouette at p, a contradiction.

So there is a contradiction in both cases, and the theorem is proved. ■

By Theorems 1 and 2, a straightforward way to find a valid $2\mathcal{PM}$ for a genus-g compact surface is to enumerate all the possible topologies of its silhouette under all orthographic viewing directions, and to see whether there is one that satisfies the condition in Theorem 2. This is exactly the information contained in vision events and aspect graphs. Methods to compute aspect graphs of different kinds of surfaces, from polyhedral and quadric surfaces, to surfaces of revolution and general implicit surfaces, have been presented [8, 9, 10, 18, 21]. One result of this paper is a method to compute aspect graphs of freeform compact NURBS surfaces under orthographic projection. However, since we are using aspect graphs to solve the mold separability problem, we do not have to compute triple-points and cusp-crossings. This is because cusps or T-junctions exist both before and after the events. So, by Theorems 1 and 2, the validity of the $2\mathcal{PM}$ does not change. We also only have to render the transparent silhouette.

4 Aspect Graph of a NURBS Surface

Since vision events occur when the viewing direction crosses the generating cones of axis cylinder developables, flecnodal scrolls and limiting bi-tangent developables, it is necessary to compute the intersection of their generating cones with the viewing sphere. From the resulting partition of the viewing sphere, we select a representative point, i.e. a representative viewing direction, from each region, and compute the silhouette corresponding to that viewing direction. If the silhouette is free from singularities, and has $g+1$ loops (for a surface of genus g), then we are sure that all the viewing directions in that region are valid separating directions of the 2-piece mold.

Now, we give all the algorithms to solve the 2-piece mold separability problem.

Note, algorithms are given only to compute the featured curves on surface; one must evaluate asymptotic directions to get the axis cylinder developable and the flecnodal scroll.

4.1 Computation of the Axis Cylinder Developable

A parabolic curve([7]) is the locus of zero Gaussian curvature points on the surface. In [11], a method that combines symbolic computation of the Gaussian curvature field of NURBs surfaces and numeric extraction of the parabolic lines as its zeros, is presented, via subdivision. Bezier surfaces are treated in [20] using a similar approach.

The parabolic curves separate separates an elliptic region ($K > 0$) from a hyperbolic region ($K < 0$). The Gaussian curvature equals [7]

$$K = \frac{||\boldsymbol{L}||}{||\boldsymbol{F}||},$$

where $\mathbf{F}$ and $\boldsymbol{L}$ are the matrices of the first and second fundamental forms [7] respectively. Since the zeros of K equates with the zeros of $||\boldsymbol{L}||$ for regular surfaces, the set of parabolic points, $\mathcal{P}$, of surface $\mathbf{S}(u,v)$ can be computed as,

$$\mathcal{P} = \left\{(u,v) \;\middle|\; 0 = (S_{uu} \cdot n)(S_{vv} \cdot n) - (S_{uv} \cdot n)^2 \right\}, \tag{4.1}$$

where $n = n(u,v)$ is the unit normal field of S.

Algorithm 1 Compute Parabolic Curves

```
Input:
  $S(u,v)$, a $C^{(2)}$ surface to compute its parabolic set.
Output:
  $\mathcal{P}$, the set of parabolic points of $S$.
Begin
  $||\overline{\boldsymbol{L}}|| \Leftarrow$ Determinant of second fundamental form
    of $S$ using unnormalized normal field of $S$;
  $\mathcal{P} \Leftarrow$ Zeros of $||\overline{\boldsymbol{L}}||$;
End
```

The unit normal field $n(u,v)$ is not rational for the given parametrization, even for a rational surface S, due to its normalization factor. However, $||\mathbf{L}||$ is rational due to the fact that it contains terms with only the square of the normalization factor. Because we seek only the zeros of $||\boldsymbol{L}||$, we can substitute $\overline{n}(u,v) = S_u \times S_v$ for $n(u,v)$. And as the surface is assumed to be regular, the zeros of $||\overline{\boldsymbol{L}}||$, using $\overline{n}(u,v)$, are identical to the zeros of $||\boldsymbol{L}||$.

The zeros of the rational modification of constraint (4.1), using $\overline{n}(u,v)$, can be computed using the equation solver presented in [12]. This solver employs the subdivision and convex hull containment of the NURBS representation and combines it with multivariate Newton Raphson improvement steps. See [12] for more.

Having 2 degrees of freedom, u and v, and 1 constraint, $||\boldsymbol{L}||$, $\mathcal{P}$ is a univariate curve in the parametric domain of S, in general (see Algorithm 1).

4.2 Computation of the Flecnodal Scroll

The flecnodal curve is the locus of inflection points of either family of asymptotic curves. At a flecnodal point, the asymptotic ray has contact of order 4 with the surface [6, 17].

At any surface point p, a local (typically not orthogonal) coordinate system for 3D space is $(S_u, S_v, \overline{n}_p)$, where $\overline{n}_p = S_u \times S_v$. (S_u, S_v) is a coordinate system for the tangent plane at p.

Note all the derivatives are evaluated at surface point p through this section.

Suppose $aS_u + bS_v$ is an asymptotic direction of 4-order contact with the surface. Then, by order 3 contact,

$$(a^2 S_{uu} + 2abS_{uv} + b^2 S_{vv}) \cdot \overline{n}_p = 0, \tag{4.2}$$

and by order 4 contact,

$$(a^3 S_{uuu} + 3a^2 bS_{uuv} + 3ab^2 S_{uvv} + b^3 S_{vvv}) \cdot \overline{n}_p = 0. \tag{4.3}$$

```
Algorithm 2  Compute Flecnodal Curves

Input:
    $S(u,v)$, a $C^{(3)}$ surface to compute flecnodal curves.
Output:
    $\mathcal{B}$, the set of flecnodal scrolls of $S$.
Begin
    $\mathcal{F}_1(u,v,a,b) \Leftarrow (a^2 S_{uu} + 2ab S_{uv} + b^2 S_{vv}) \cdot \overline{n}_p$;
    $\mathcal{F}_2(u,v,a,b) \Leftarrow$
        $(a^3 S_{uuu} + 3a^2 b S_{uuv} + 3ab^2 S_{uvv} + b^3 S_{vvv}) \cdot \overline{n}_p$;
    $\mathcal{F}_3(u,v,a,b) \Leftarrow a^2 + b^2 - 1$;
    $\mathcal{B} \Leftarrow$ Simultaneous solution of $\{\mathcal{F}_i(u,v,a,b) = 0\}_{i=1}^{3}$ ;
End
```

```
Algorithm 3  Compute Bi-tangent Curves

Input:
    $S(u,v)$, a $C^1$ surface to compute bi-tangent curves.
Output:
    $\mathcal{B}$, the set of bi-tangent sheets of $S$.
Begin
    $\overline{n} = \frac{\partial S(u,v)}{\partial u} \times \frac{\partial S(u,v)}{\partial v}$;
    $\mathcal{F}_1(u,v,s,t) \Leftarrow \langle \overline{n}, S(u,v) - S(s,t) \rangle$ ;
    $\mathcal{F}_2(u,v,s,t) \Leftarrow \left\langle \overline{n}, \frac{\partial S(s,t)}{\partial s} \right\rangle$ ;
    $\mathcal{F}_3(u,v,s,t) \Leftarrow \left\langle \overline{n}, \frac{\partial S(s,t)}{\partial t} \right\rangle$ ;
    $\mathcal{B} \Leftarrow$ Simultaneous solution of $\{\mathcal{F}_i(u,v,s,t) = 0\}_{i=1}^{3}$ ;
End
```

Equation (4.2) simply means that the second fundamental form is 0 for an asymptotic direction (order-3 contact). Equation (4.3) can be understood as follow.

Define a function $\mathcal{F}$ as

$$\mathcal{F} : S \longrightarrow R, \mathcal{F}(x) = (x - p) \cdot \overline{n}_p,$$

for any point x on surface S.

If there is a order-4 contact at surface point p along direction $aS_u + bS_v$, then the third directional derivative of $\mathcal{F}$ should be 0 at p, which immediately gives Equation (4.3).(Of course, the second derivative should also be 0, which gives Equation (4.2), exactly as second fundamental form does).

Finally,

$$a^2 + b^2 = 1. \qquad (4.4)$$

Note that this equation is not the condition for a unit vector because (S_u, S_v) is not necessarily an orthonormal basis. Instead, it just lets us pick one unique vector from each direction.

Putting Equations (4.2), (4.3) and (4.4) together, we have 4 variables (parameter u and v, tangent direction variables a and b) and 3 equations, so the solution should be one dimension set, i.e., a curve on the surface (see Algorithm 2).

Again we use the constraint solver, presented at [12], to find the solution. This technique is applicable because all the above 3 equations are the sum, product, and derivative of NURBS, and therefore $\mathcal{F}_i$ are also NURBS [11].

4.3 Computation of the Bi-tangent Developable

Given a C^1 continuous parametric surface S, surface points $S(u,v)$ and $S(s,t)$ are bi-tangent points if :

$$0 = \langle \overline{n}(u,v), S(u,v) - S(s,t) \rangle, \qquad (4.5)$$

$$0 = \left\langle \overline{n}(u,v), \frac{\partial S(s,t)}{\partial s} \right\rangle, \qquad (4.6)$$

$$0 = \left\langle \overline{n}(u,v), \frac{\partial S(s,t)}{\partial t} \right\rangle, \qquad (4.7)$$

where $\overline{n}(u,v) = \frac{\partial S(u,v)}{\partial u} \times \frac{\partial S(u,v)}{\partial v}$.

Constraint (4.5) ensures that the normal at (u,v) is orthogonal to the line segment connecting $S(u,v)$ and $S(s,t)$. Constraints (4.6) and (4.7) guarantee that the two normals at (u,v) and at (s,t) are indeed in the exact same direction.

Once again, we employ the constraints solver of [12] to solve Equations (4.5) to (4.7).

Having 4 degrees of freedom, u, v, s and t, and 3 constraints, $\mathcal{F}_i(u,v,s,t)$, $i = 1,2,3$, the solution $\mathcal{B}$ is a univariate over S, in general (see Algorithm 3). Here, each solution is a pair of points, and defines a bi-tangent line in space from $S(u,v)$ or $S(s,t)$. A developable sheet could then be easily constructed by connecting these corresponding bi-tangent points for all the solution set of $\mathcal{B}$.

4.4 Partitioning the Viewing Sphere Via Image Dilation

After computing all the intersections between $\mathscr{S}^2$ and the various generating cones corresponding to axis cylinder developables, flecnodal scrolls and bi-tangent developables, it is necessary to find a representative point in each partitioned region on $\mathscr{S}^2$.

Our algorithm uses image dilation [14], and is implemented in open GL. It is robust and efficient.

First, the intersection curves and their antipodals are centrally projected onto left, back and top faces of a cube that bounds the unit sphere. Note the right, front, and bottom faces would yield similar antipodal regions to the first three faces and hence are ignored. The intersection curves are rendered into 3 binary images. Dilation is then used 3 times to find, for each region of pixel value 0 (background), the central point, which is then centrally projected back onto $\mathscr{S}^2$.

The first dilation is applied to the binary image to achieve the effect of shrinking the boundary foreground curves inward to find the representative points, i.e., $\mathcal{P}$ in Algorithm 4. The second dilation is applied to the binary image of reversed polarity (i.e., regarding background points as foreground points, and vice versa.) to partition $\mathcal{P}$ into equivalent classes, i.e., $\mathbb{P}$ in Algorithm 4. Two points are equivalent if they are 4-connected, i.e. reachable to each other via dilation using the cross kernel from equation 4.8. The third dilation is applied to each equivalent class of points to find the heuristically optimal point, which has the maximal distance to the surrounding foreground curve. (see Algorithm 4).

All dilations use the same cross kernel,

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \tag{4.8}$$

i.e., each foreground pixel is expanded to its 4-neighbors.

Because central projection of $\mathscr{S}^2$ onto the bounding cube is used to generate the bitmap, an extra boundary curve for each cube's face image is introduced. Therefore, for those regions spanning across the boundary of a face under the central projection, more than one representative point will be returned by our algorithms. However, this causes no harm beyond a slight computational overhead.

5 Examples

We give two examples. See Figure 3 and Figure 4.

For each example, we show its NURBS model, its partitioned viewing sphere, and its representative viewing directions and their corresponding silhouettes.

For the first example model, there are four representative non-singular silhouettes and correspondingly four representative viewing directions, from each of which there is a valid $2\mathcal{PM}$.

Thus, for this model, any viewing direction from the same region as any one of these four directions is a valid separating direction for a $2\mathcal{PM}$.

Algorithm 4 Partition the View Sphere

```
Input:
  $\mathcal{C}$, a set of curves and their antipodal curves on $\mathscr{S}^2$.
Output:
  $\mathcal{V}$, a set of points, one for each region on $\mathscr{S}^2$
  partitioned by $\mathcal{C}$ and cube edges projected onto $\mathscr{S}^2$.
Begin
  $\{I_i\}_{i=1}^{3} \Leftarrow$ bitmaps rendered by central projection
      of $\mathcal{C}$ onto left, back and top faces of the cube;
  $\mathcal{V} \Leftarrow \emptyset$;
  Foreach $I_i$, $i = 1, 2, 3$, Do
    $\mathcal{P} \Leftarrow \emptyset$;
    Repeat
      Do dilation;
      If background point $\mathbf{p}$, has filled 4-neighbors
              $\mathcal{P} \Leftarrow \mathcal{P} \bigcup \{\mathbf{p}\}$;
    Until no more background points;
    Partition $\mathcal{P}$ into set of 4-connected point sets $\mathbb{P}$;
    Foreach $\mathcal{P}'$ in $\mathbb{P}$, Do
      Foreach $\mathbf{p}$ in $\mathcal{P}'$, Do
        Set $\mathbf{p}$ as a foreground point;
        Expand $\mathbf{p}$ to the foreground boundary;
        If $\mathbf{p}$ has max dist to the boundary so far;
          $\mathbf{v} \leftarrow$ projecting $\mathbf{p}$ back onto $\mathscr{S}^2$
      Od
      $\mathcal{V} \Leftarrow \mathcal{V} \bigcup \{\mathbf{v}, -\mathbf{v}\}$;
    Od
  Od
End
```

Note, there are actually only two topologically distinct valid viewing directions. We have two extra since we project the sphere surface onto three faces of the bounding cube, but we apply dilation to them separately as if the three faces are *disconnected*.

For the second example model, there is only one representative non-singular silhouette, and correspondingly only one representative viewing direction, from which there is a valid $2\mathcal{PM}$. Any viewing direction from the same region as this direction is also a valid separating direction for a $2\mathcal{PM}$.

6 Computing the Partition Line

Let V be a selected unit vector viewing direction from which a regular $C^{(3)}$ surface $S(u, v)$ should be split, to create a proper $2\mathcal{PM}$ mold. The silhouette extraction problem can be reduced to the following algebraic constraint:

$$\mathcal{S} : \left\langle \frac{\partial S}{\partial u} \times \frac{\partial S}{\partial v}, V \right\rangle = \langle n(u, v), V \rangle = 0,$$

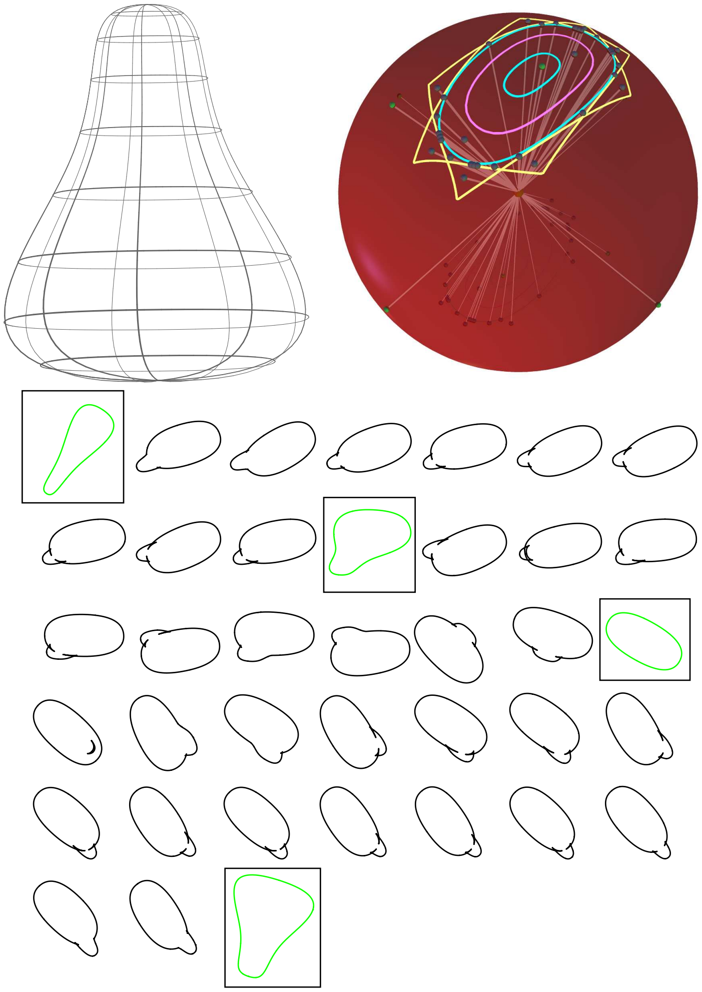


Figure 3. Mold Accessibility Example One. *Shown on top left is a NURBS model. Top right shows both the intersecting curves of the various generating cones with the viewing sphere, and the representative viewing directions of the partitioned regions. The curve corresponding to a parabolic developable is in cyan, to a bi-tangent developable in magenta, and to a flecnodal scroll in yellow. The representative viewing directions are represented as pairs of antipodal small spheres embedded on the transparent viewing sphere. Silhouettes viewed from all these directions are shown next. There are* four *representative non-singular silhouettes, shown inside boxes with green color, and correspondingly* four *representative viewing directions (shown in green), from each of which there is a valid $2\mathcal{PM}$. (One of them is difficult to see, since it is almost parallel to the view direction used to render this image.) Note, there are actually only two topologically distinct valid viewing directions. We have two extra since we project the sphere surface onto three faces of the bounding cube, but we apply dilation to them separately as if the three faces are* disconnected.

where $n(u, v)$ is the unnormalized normal field of $S(u, v)$.

However, in many practical cases, a proper mold should present no vertical walls. That is, the design of the mold should allow no surface regions where the surface normals are (almost) orthogonal to V. Such vertical surface regions greatly increase the difficulty in extracting the manufactured part out of the mold and hence should be eliminated. Mathematically speaking, we seek to eliminate all regions, $\mathcal{R}$, in the surface, with normals having an angular deviation from V of almost 90 degrees:

$$\mathcal{R} = \left\{(u_0, v_0) \mid \langle \overline{n}(u_0, v_0), V\rangle < \cos(\alpha)\right\},$$

where $\overline{n}$ is the normalized normal field of S and α is the permissible angular deviation from the normal.

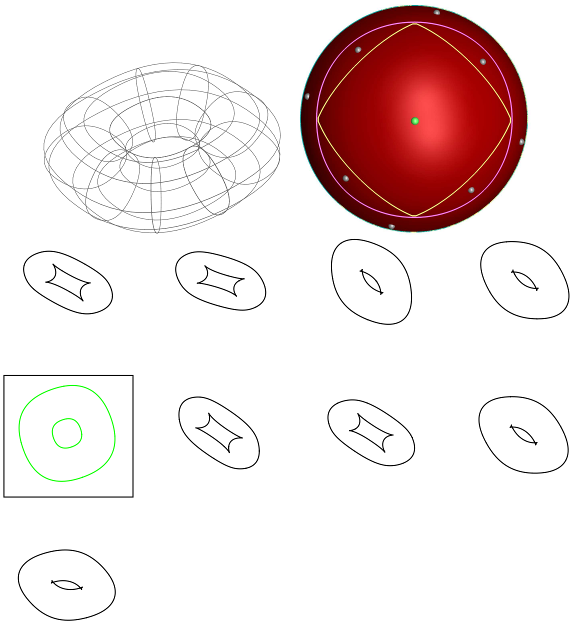


Figure 4. Mold Accessibility Example Two. *Shown top left is a NURBS model. The top right shows both the intersecting curves of the various generating cones with the viewing sphere, and the representative viewing directions of the partitioned regions. The curve corresponding to a parabolic developable is cyan, to a bi-tangent developable is magenta, and to a flecnodal scroll is yellow. The representative viewing directions are represented as small spheres on the opaque viewing sphere. Silhouettes viewed from all these directions are shown next. There is* one *representative non-singular silhouette, shown inside a box with green color, and correspondingly* one *representative viewing direction, colored green, from which there is a valid $2\mathcal{PM}$.*

The boundary of region $\mathcal{R}$ is prescribed by an *isocline* or *isophotes*, defined in [15] as the locus of points on $S(u,v)$ that presents a normal with a constant inclination angle with respect to a prescribed vector V, such as a light source direction. An isocline is typically a curve. The isocline's constraint could be formulated as

$$\mathcal{I} : \langle \overline{n}(u,v), V \rangle = \cos(\alpha),$$

which is not rational for the given parametrization. Hence, we square the expression to yield,

$$\mathcal{I} : \langle n(u,v), V \rangle^2 = \cos^2(\alpha) \, \langle n(u,v), n(u,v) \rangle \, . \tag{6.1}$$

Equation (6.1) extracts, due to the squaring operation, both front facing and back facing isoclines. In other words, we extact all surface points with normals with either a $90 - \alpha$ or a $90 + \alpha$ degree angular deviation from V.

Figure 5 (a) shows the isoclines extracted for a surface in the shape of a wine glass, at 90 ± 10 degrees inclinations. In (b), the regions near the silhouette, between $90 - 10$ and $90 + 10$, are trimmed away and the result is represented as a trimmed surface.

The trimming of the regions with normals that are almost vertical leaves gaps in the geometry (See Figure 5 (b)). We should fill these gaps with the maximally valid mold-extraction slope as prescribed by α. This ruling extension is conducted in pairs from both $90 - \alpha$ and $90 + \alpha$, toward the local silhouette (at 90) and further clipped against each other. Figure 5 (c) shows the result of this ruling extension for the wine glass.

To summarize, we end up with a modified model that is better suited as a mold with no vertical walls and maximal slopes of $90 \pm \alpha$ degrees. Figure 6 shows two shaded examples of the wine glass from Figure 5 and of the handle of the Utah teapot model, after applying the presented isoclines' clipping and ruling extension processes.

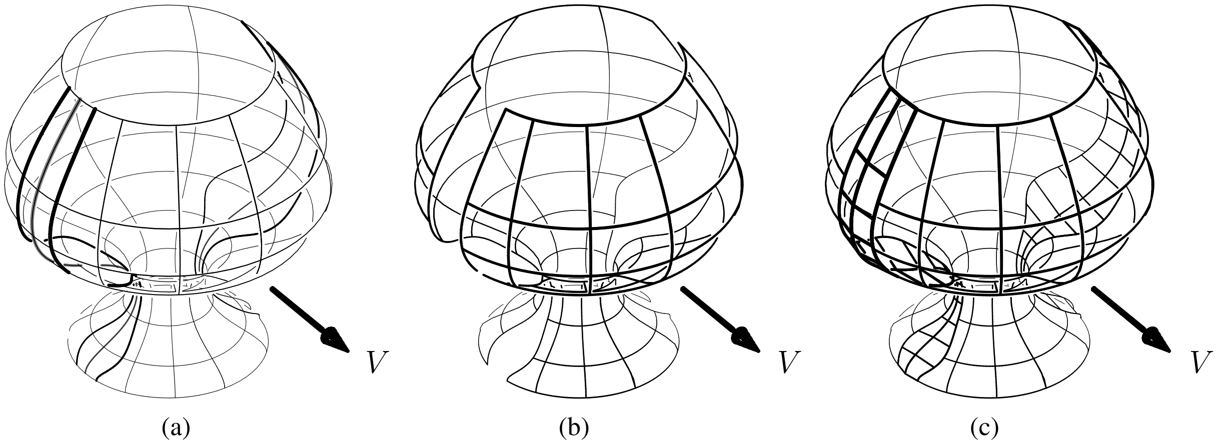
7 Conclusion

In this paper, we have presented a complete solution to the 2-piece mold separability problem for a model bounded by $C^{(3)}$ surfaces represented as NURBS and a set of algorithms for computing the aspect graph of such a model. Clearly, the presented solution is not limited to NURBS surfaces and any representation that supports the differential analysis conducted in this work might be equally employed. In addition we show how to compute and deal with surfaces having *almost vertical parts*.

The solution to the 2-piece mold separability problem is based on the aspect graph. All algorithms have been implemented and results are demonstrated by several examples in this paper. Our solution requires that the boundary be $C^{(3)}$ only because that condition is necessary to compute continuous flecnodal curves.

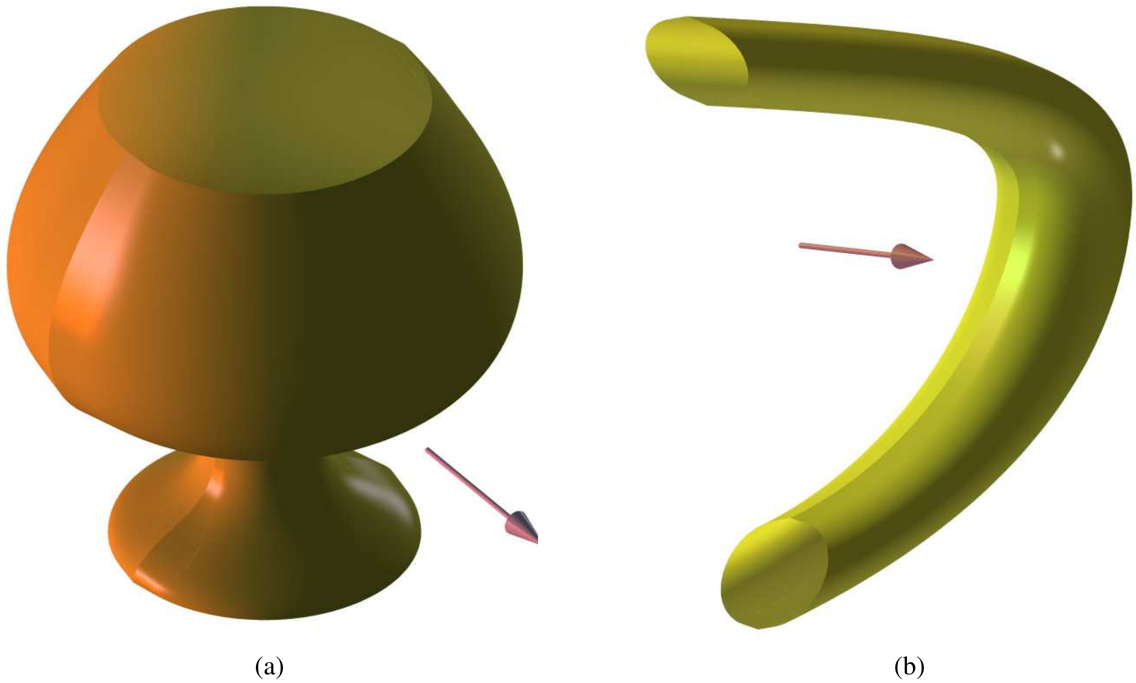
Extensions to this work include the support of models bounded by surfaces which are only piecewise smooth, handling the surfaces one at a time and processing the boundaries between surfaces as special cases. The treatment of planar facets (polyhedral models) might be of interest as well, being a simple yet common case.

It takes a few minutes to achieve the results presented in this work on a Pentium 4 machine, but it is expected to take much more time for any practical models. Hence, methods to optimize the extract of all the feature curves must be sought along with robust methods to automatically detect singularity in the silhouettes.



(a) (b) (c)

Figure 5. 90 ± 10 **isoclines (thick black lines) from** V **for a wine glass surface (a). The silhouettes of the surface from** V **are shown in gray color. In (b), the regions with normals that are almost vertical are trimmed away. (c) shows the ruled extension that covers up these gaps at the proper valid slopes.**



(a) (b)

Figure 6. Final clipped and ruled extended models of the wine glass from Figure 5 (a) and of the handle of the Utah teapot (a). Note the sharp, normal discontinuity, along the partitioning line.

Acknowledgments

The authors are grateful to David Johnson, Rich Riesenfeld and other members of the geometric design and computation group for their multiple research discussions.

References

[1] H.-K. Ahn, M. D. Berg, P. Bose, S. Cheng, D. Halperin, J. Matousek, and O. Cheong, "Separating an Ojbect from its Cast," *Computer-Aided Design*, 2002, pp. 547–559.

[2] V. Arnold, "Singularities of Systems of Rays," *Russian Mathematical Survey*, vol. 38, no. 2, 1983, pp. 87–176.

[3] V. Arnold, *Catastrophe Theory*, 3 edition, Springer-Verlag, 1992.

[4] J. Bown, *Injection Moulding of Plastic Components*, McGraw-Hill, 1979.

[5] L. Chen, S. Chou, and T. Woo, "Paring directions for mould and die design," *Computer-Aided Design*, vol. 25, 1993, pp. 762–768.

[6] R. Cipolla and P. Giblin, *Visual Motion of Curves and Surfaces*, Canbrudge University press, 2000.

[7] M. do Carmo, *Elementary Differential Geometry*, 2 edition, Prentice-Hall, 1976.

[8] D. Eggert and K. Bowyer, "Computing the orthographic projection aspect graph of solids of revolution," *Workshop on Proceedings of Interpretation of 3D Scenes*, November 1989, pp. 102 –108.

[9] D. Eggert and K. Bowyer, "Perspective projection aspect graphs of solids of revolution: an implementation," *Workshop on Directions in Automated CAD-Based Vision*, June 1991, pp. 44 –53.

[10] D. Eggert and K. Bowyer, "Computing the perspective projection aspect graph of solids of revolution," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, Feb 1993, pp. 109 –128.

[11] G. Elber and E. Cohen, "Second Order Surface Analysis Using Hybrid of Symbolic and Numeric Operators," *ACM Transaction on graphics*, vol. 12, no. 2, 1993, pp. 160–178.

[12] G. Elber and M.-S. Kim, "Geometric Constraint Solver using Multivariate Rational Spline Functions," *The Sixth ACM/IEEE Symposium on Solid Modeling and Applications*, June 2001, pp. 1–10.

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

[14] R. Gonzalez and R. Woods, *Digital Image Processing*, Addison-Wesley Publishing Company, 1992.

[15] J. Hoschek, D. Lasser, and L. L. Schumaker, *Fundamentals of computer aided geometric design*, A. K. Peters, Ltd., 1993.

[16] K. Hui and S. Tan, "Mould design with sweep operations - a heuristic search approach," *Computer-Aided Design*, vol. 24, 1992, pp. 81–91.

[17] J. J. Koenderink, *Solid Shape*, MIT press, 1990.

[18] D. Kriegman and J. Ponce, "Computing exact aspect graphs of curved objects: solids of revolution," *Workshop on Proceedings of Interpretation of 3D Scenes*, Nov. 1989, pp. 116–122.

[19] B. O'Neill, *Elementary Differential Geometry*, 2 edition, Academic Press, 1997.

[20] T. S. Smith and R. T. Farouki, "Gauss map computation for free-form surfaces," *Computer Aided Geometric Design*, vol. 18, no. 9, 2001, pp. 831–850.

[21] S.Petitjean and J. Ponce, "Computing exact aspect graphs of curved objects: algebraic surfaces," *International Journal of Computer Vision*, vol. 9, no. 3, 1992, pp. 231–255.