A Reverse Glue Approach to Automated Construction of Multi-Piece Molds

Yong Chen
3D Systems, Inc.,
Valencia, CA
e-mail: cheny@3dsystems.com

David W. Rosen
School of Mechanical Engineering,
Georgia Institute of Technology,
Atlanta, GA 30332-0405
e-mail: david.rosen@me.gatech.edu

Mold design can be a difficult, time-consuming process. Determining how to split a mold cavity into multiple mold pieces (e.g., core, cavity) manually can be a tedious process. This paper focuses on the mold construction step of the automated mold design process. By investigating glue operations and its relations with parting faces, an approach based on a new reverse glue operation is presented. The key to the reverse glue operation is to generate parting faces. A problem definition of parting face generation for a region is provided. Correspondingly, three face generating criteria are identified. Based on the parting lines of a region, our algorithms to generate the parting faces are presented. Our mold construction algorithms for two-piece molds and multi-piece molds are also presented with brief discussions. Some industrial examples are provided which illustrate the efficiency and effectiveness of our approach. We tested our mold designs by fabricating stereolithography mold inserts (a rapid tooling method) and molding parts.
[DOI: 10.1115/1.1603308]

Keywords: *Injection Mold, Rapid Tooling, Automated Construction, Geometric Reasoning*

1 Introduction

Many consumer products involve the design and fabrication of injection molded thermoplastic parts. To reduce time to market, it is necessary to reduce the time required to test molded part designs, as well as mold designs. Rapid tooling, which uses a rapid prototyping (RP) technique to fabricate tools or patterns, can reduce tooling cost and time especially when only small numbers of parts are needed [1]. By using rapid tooling, delivering prototype parts with turn-around times of less than two weeks becomes feasible. However, achieving the goal requires fast, proven mold design methods.

Mainly from the geometric perspective, we develop a systematic approach and a related system to automate several important mold design steps, including selection of parting directions, parting lines, parting surfaces, and construction of mold pieces. We divide the mold design process into three phases as shown in Fig. 1. Our approaches for the first two phases, region generation and combination, are presented in [2]. In this paper, we present an approach for constructing mold pieces based on the mold configurations generated in Phase 2. Our approach is presented in the general context of designing configurations of mold cavity plates, side actions, and form pins, those parts of the mold that form the molded part shape. We have applied our methods primarily to the design of mold inserts that are fabricated using stereolithography. The general term ''mold piece'' is used to describe cavity plates, inserts, or other parts of the mold used to form the part shape. We will refer to the collection of all of these ''mold pieces'' as the ''mold cavity;'' hence the main challenge being addressed in this paper is how to decompose the mold cavity into a collection of two or more mold pieces.

After the region combination process, several mold piece regions are generated with their parting directions (***PD***) and parting lines (PLs). We use an example that is shown in Fig. 2 to illustrate the region combination results. For a simple part with a through hole and two grooves, three mold piece regions ($R_1 \sim R_3$) are generated in Phase 2, whose faces are marked by magenta, yellow, and blue (medium, light, dark shading), respectively (Fig. 2(a)). This is the fewest number of mold piece regions possible to ensure the mold can be disassembled and the part ejected. Since the regions cannot be combined further, they are called mold piece regions and each of them corresponds to a mold piece ($M_1 \sim M_3$). In the combination process, a feasible parting direction ($\boldsymbol{PD}_1 \sim \boldsymbol{PD}_3$) is recorded for each region. That is, along the parting direction of region R_i, the mold piece M_i can be translated from the injection molded part without any interference. In the combining process, we also record the parting lines of each region (PL_i), which are the non-intersecting closed continuous loops defined by edges along the boundary of the region (refer to Fig. 2(b)). If a mold cavity (MC) related to the part is given, we can construct mold pieces as shown in Fig. 2(c) for the mold piece regions.

So the problem considered in this paper is defined as follows.

Problem MPC: Mold Piece Construction. A solid polyhedral part P and a mold cavity MC are given. Suppose regions $R_i(F,E,\boldsymbol{D})$ $(1 \leqslant i \leqslant k)$ are also given, where

1. F consists of a set of connected faces of P that belong to region R_i;
2. E is the boundary edges of the faces in F, representing the parting line of region R_i;
3. $\boldsymbol{D}$ is a direction which makes an angle of at most 90 deg with the outward normals of all faces in F;

Generate bodies $M_1 \sim M_k$ such that:

(a) After $M_1 \sim M_k$ are assembled together, they form MC with a cavity P inside. That is, $MC - P = M_1 \cup M_2 \cup \ldots \cup M_k$, where $-$ and $\cup$ are the Boolean operators, subtraction and union, respectively.
(b) Faces of R_i are formed and only formed by M_i in the injection process.
(c) M_i can be disassembled in direction $\boldsymbol{PD}_i$ without interference with other mold pieces.

The remainder of this paper has been organized in the following manner. In Section 2, we review the related work in the automated construction of mold pieces. Both academic and industrial approaches are investigated. In Section 3, the principles and representations related to mold piece construction and the reverse glue

Contributed by the Computer-Aided Product Development (CAPD) Committee for publication in the JOURNAL OF COMPUTING AND INFORMATION SCIENCE IN ENGINEERING. Manuscript received Dec. 2002; Revised Jun. 2003. Associate Editor: R. Crawford.

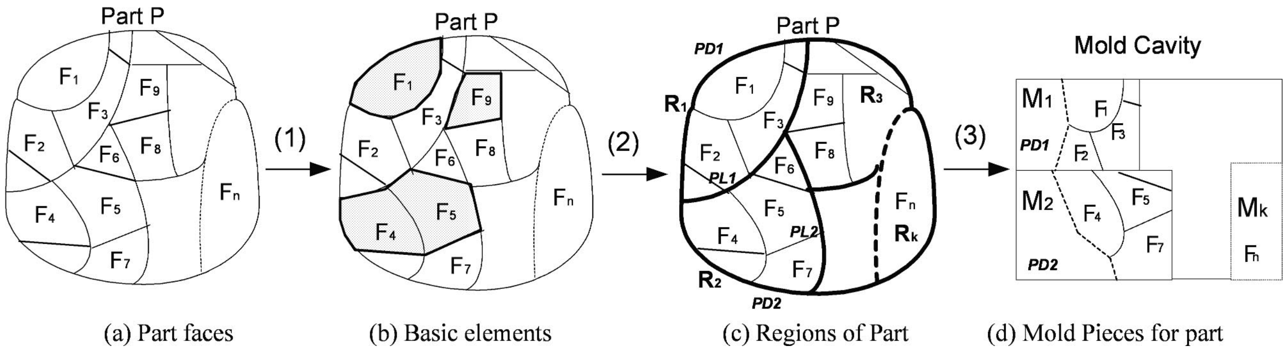


Fig. 1 Mold design process

operation are presented. Criteria for generating parting faces, the key step of our approach, are presented in Section 4. An algorithm for parting face generation and two examples are presented in Section 5. We present our algorithms for constructing two-piece and multi-piece molds in Section 6. The system implementation and results are discussed in Section 7. Conclusions are summarized in Section 8.

2 Review of Related Work

Considerable work has been performed in the injection mold design area. Recently, work has focused on automated methods for multi-piece mold design [3] and improved methods for parting line and surface generation. In the latter area, Majhi et al. [4] presented a parting line generating approach for a convex polyhedron. The authors identified two flatness criteria and generated a parting line that is as ''flat'' as possible. However, for a given parting line, we did not find literature discussing generating ''flattest'' parting face to form mold pieces. Currently two approaches are proposed for automatically splitting the mold core and cavity. They are the approach based on extending parting lines and that based on sweeping.

2.1 Approach Based on Extending Parting Lines. A mold cavity can be cut into two pieces by a parting surface. For parting lines not in a plane, Tan et al. [5] presented a parting surface generation method. After getting the parting lines for a given parting direction, an outer loop and inner loops are generated. A convex hull algorithm is applied to the outer loop. Each edge of the hull is projected to an adjacent side face of the mold cavity. The projection direction is perpendicular to the parting direction but parallel to the surface normal of the side face of the mold cavity. All planar faces generated by projecting hull edges can form a parting surface for the part. For each inner loop, triangular facets are created within the loop. Shin and Lee [6], Serrar [7], and Nee et al. [8] used a similar approach to form the parting surface to split mold cavity into two halves. More recently, Priyadarshi and

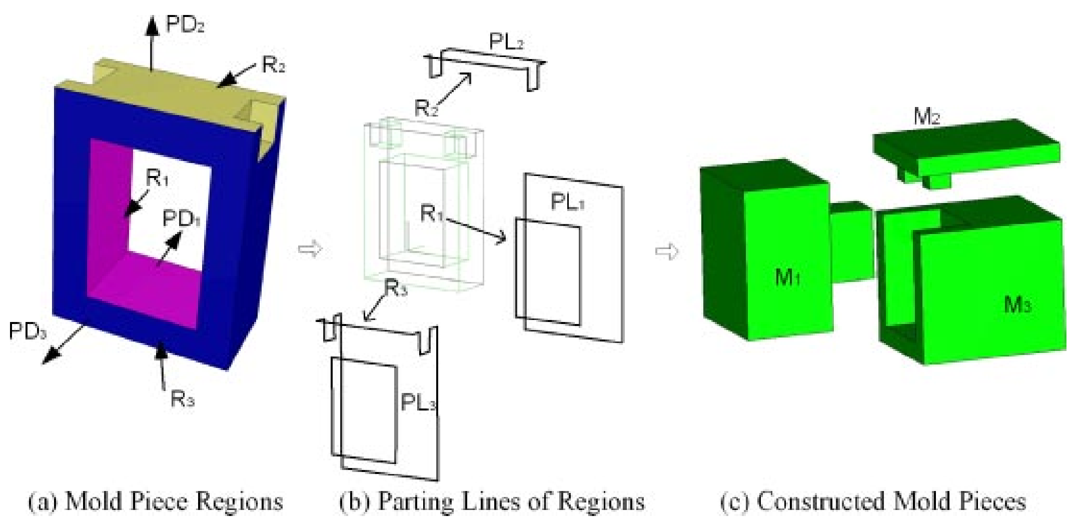


Fig. 2 Regions and mold pieces of a part

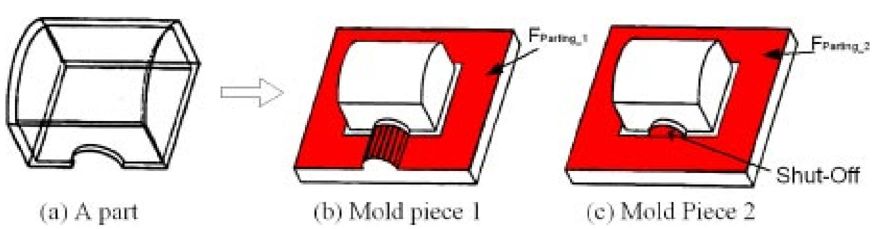
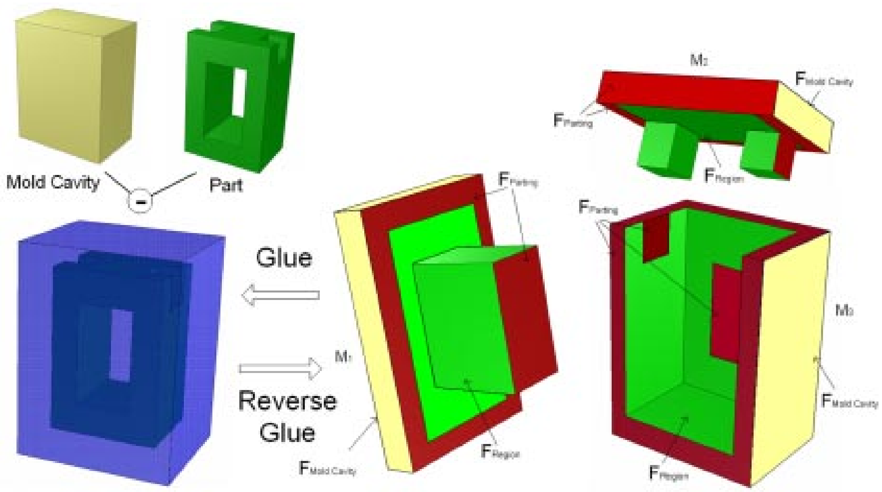


Fig. 3 Part with mold pieces generated by two approaches



(a) Boolean Mold Cavity (b) Mold Pieces

Fig. 4 Mold piece face classification

Gupta [9] adapted Tan's approach to create parting surface for an outer loop while used a surfacing method, called covering, to fit a surface over the inner loops.

This approach is quite straightforward. Extending the given parting lines outward into faces splits the mold cavity into two mold pieces. However, two problems are identified.

(1) For non-flat parting lines, the parting surface generated by this approach is also not flat. This is not accordance with the best practice of mold design, that is, the parting surface should be planar to decrease manufacturing complexity and to increase the shut-off force to reduce material flash [10]. For example, for the part in Fig. 3(a) one mold piece generated by the above approach is shown in Fig. 3(b), and the mold piece generated by our algorithm is in Fig. 3(c). Parting faces are shown in red (shaded). Compared with Fig. 3(b), the mold piece in Fig. 3(c) is less expensive to fabricate. Furthermore, less material flash is expected in molded parts because the parting surface has better accuracy and surface finish. This is very important in rapid tooling since mold pieces fabricated by layered manufacturing methods will have rough surface finish for near-flat faces.

(2) Two different algorithms (extending parting lines and triangulation/covering) based on different principles were developed individually in the literature to handle the outer loop and inner loops of parting lines, respectively. In comparison, the approach presented in this paper can handle both outer and inner parting lines using the same approach.

2.2 Approach Based on Sweeping. Conceptually removing an injection molded part from a mold in the parting direction is similar to sweeping part faces in the same parting direction. Hui and Tan [11] described an algorithm using sweep operations and Boolean operations to generate a mold core and cavity. First, sweeping the part in the parting direction generates a solid. Then, using two mold plates to subtract each end of the solid, two mold pieces are generated. The algorithm does not consider internal parting lines. So for a shape with a through hole, the algorithm will not generate the mold pieces correctly. Urabe and Wright [12] also present a mold construction method based on the sweeping of faces into bodies. Then they are united with plates and mold walls to form core and cavity pieces. The sweeping operations and Boolean operations (unite) are time consuming and can have robustness problems.

2.3 Industrial Approaches. Currently several commercial mold design software systems can automatically split mold core and cavity for a part. We investigated five leading systems as listed below.

(1) **MoldWizard:** From *Unigraphics* [13], this is a highly automated product that incorporates some industry best practices to guide users through the steps required to construct a mold. Based on user's input, the main steps for constructing core and cavity for an industrial part include the following. First, parting lines for the given part are identified by the designer. Then for each parting edge, a sweeping direction is specified and a parting surface is generated. These parting surfaces together with other generated from inner loops cut the mold cavity into two mold pieces.

(2) **Moldplus:** As a supplementary program for *Mastercam®*, *Moldplus* [14] generates parting surfaces for given parting lines, apparently in a similar manner to MoldWizard.

(3) **QuickSplit:** This splitting tool developed by *Cimatron Ltd.* can separate core and cavity with sliders and inserts [15].

(4) **Magics RP:** *Materialise* developed a rapid tooling module for this system [16], which automates the design of the insert tool, based on the STL file of a part.

(5) **IMOLD:** This is a supplementary program for *Unigraphics* and *SolidWorks* [17]. It has a module, *Core/Cavity Builder*, to handle the parting of cores and cavities for both solid and surface product models.

The approach to generation of parting surfaces used in IMOLD was presented in [18], which is based on extending parting lines. For other systems, we did not find published works on their splitting approaches. However, we believe all systems use the approach based on extending parting lines.

3 Mold Construction Principle and Related Representations

The basic idea in our mold construction approach is to separate the mold cavity into pieces at parting lines by generating parting faces that are parallel or perpendicular to the parting direction and such that the resulting parting surfaces are as flat as possible. The main challenges are to identify parting lines that lead to the best set of parting faces and to generate those parting faces. After generating parting lines and faces, mold pieces are constructed by a new "reverse glue" operation that is an application of boundary representation Euler operations, rather than the more complex and time consuming sweep and Boolean operations used by others.

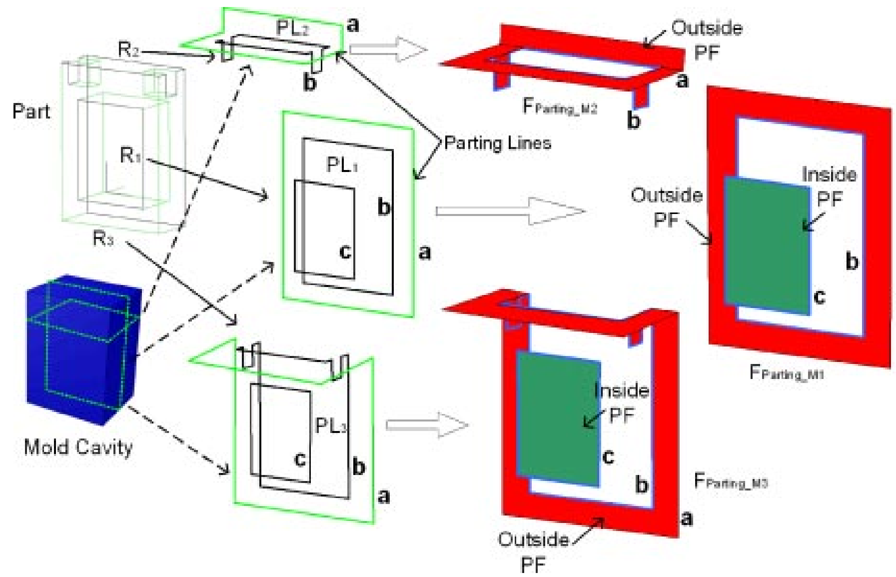


Fig. 5 Parting face generation based on edge loops

Suppose a mold cavity (MC) is given for a molded part (P). Furthermore MC contains P entirely inside (Fig. 4(a)). As defined in Problem MPC, mold pieces should form P inside and MC outside when they are assembled together. Therefore, any generated mold piece M_i related to region R_i consists of three kinds of faces (Fig. 4(b)):

1. Faces related to part region R_i (F_{Region}): For each face f of R_i, there is a corresponding face f' in M_i. Faces f and f' are in the same position with opposite face normals. For part regions as shown in Fig. 2(a), their corresponding faces F_{Region} are represented in green (medium shading) in Fig. 4(b).
2. Faces related to the mold cavity (F_{MC}): Some faces of M_i come from the mold cavity. Each face of the mold cavity is assigned to one mold piece. For the mold cavity as shown in Fig. 4(a) faces F_{MC} are represented in yellow (lightest shading) in Fig. 4(b).
3. *Parting Faces* ($F_{Parting}$): Parting faces of a mold piece M_i are the faces that connect faces F_{Region} and F_{MC} to make M_i as a solid body. They are also the contacting faces between different mold pieces when they are assembled together. Parting faces are represented in red (dark shading) in Fig. 4(b). Parting faces usually have normal vectors that are parallel or perpendicular to ***PD***. To avoid non-planar parting surfaces (e.g., the mold piece in Fig. 3(b), parting faces with normals perpendicular to ***PD*** are generated, forming "shut-offs" in the mold. Shut-offs are pairs of vertical mold faces that meet to "shut off" the flow of polymer during molding [19]. Shut-off faces are generated by projecting edges in parting lines onto a parting surface, then constructing faces from the edges and their projections. This is discussed further in Sections 4 and 5.

Gluing operations are commonly used in Boolean operations to "glue" part boundaries into a single solid boundary. Central to a glue operation is a global topological Euler operation, $kfmrh$ (kill face, make ring, hole) [20]. A glue operation consists of three steps: 1) merge two half-edge data structures into one data structure that has two shells, 2) join the shells by applying the $kfmrh$ operation, and 3) merge coincident edges and vertices that result from $kfmrh$. Step 3 is implemented as a collection of low level Euler operations.

By reversing the steps in the gluing operation, a solid can be separated into multiple pieces, necessary for mold construction. The steps of the **Reverse Glue** operation are: 1) copy and separate edges and vertices that comprise parting lines, 2) generate separate shells using Euler operation $mfkrh$ (make face, kill ring, hole), and 3) generate separate solids. Step 1 involves swapping partner coedges and low level Euler operations.

Since MC and P are known while (M_1 on M_2) and (M_2 on M_1) are unknown, Problem MPC is transformed into a parting face generation problem, which is discussed in the next section.

4 Parting Face Generation Criteria

4.1 Parting Faces. Two kinds of parting faces, *outside parting faces* and *inside parting faces*, may exist in a mold piece, as shown in Fig. 5. Outside parting faces (Outside PF) connect part faces with mold cavity faces in the mold piece. Correspondingly their boundaries are an outer parting loop of the part (loops **b** in Fig. 5) and a parting line on the mold cavity (loops **a**). Inside parting faces (Inside PF) relate to the holes in the injection molded part. Their boundaries are the inner parting loops of the part (loops **c**). For any region, there is one and only one outer parting loop, with zero or more inner parting loops. We will use the term "parting loop" to denote parting lines and loops to emphasize that they are loops of edges in the solid model.

By definition, $F_{Parting\_MPi}=F_{Outside\_Parting\_MPi}+F_{Inside\_Parting\_MPi}$. Careful inspection of Fig. 5 shows that parting loops and faces are shared, in part, by the mold pieces. For example, the vertical part of loop PL_{3a} overlaps PL_{1a}; the remaining part of PL_{1a} overlaps PL_{2a}. Generalizing, we can write:

$$b(F_{Outside\_Parting\_MPi})+b(F_{Inside\_Parting\_MPi})$$
$$=b(F_{Region\_MPi})+b(F_{MC\_MPi}) \quad (1)$$

In Eq. (1), we assume a $b(F_{MC\_MPi})$ and can generate easily faces F_{Region_MPi} since they are determined by part region R_i. $b(F_{Region\_MPi})$ can be generated easily (e.g., $PL_1 \sim PL_3$ in Fig. 5) since an edge belongs to $b(F_{Region\_MPi})$ if its two neighboring faces F_{N1} and F_{N2} satisfy $F_{N1}\in F_{Region\_MPi}$ and $F_{N2}\notin F_{Region\_MPi}$. Hence, the boundary of $F_{Outside\_Parting\_MPi}$ and $F_{Inside\_Parting\_MPi}$ can be generated from Eq. (1). Therefore, we are trying to generate a face set based on its boundary edges in the parting face generation process (that is, loops $PL_i \Rightarrow$faces $F_{Parting\_Mi}$ in Fig. 5). Hence, the problem is a *geometric reconstruction* problem with an infinite number of solutions.

The approach to generating parting faces involves two main ideas. First, candidate parting planes are generated and tested.

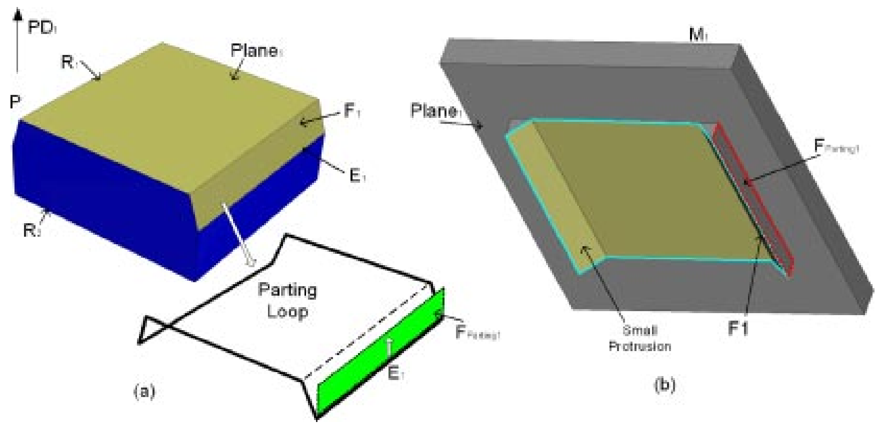


Fig. 6 Example of minimum strength criterion

Since parting loop edges are projected onto these planes, we call them "projection planes." Shut-off faces (normals perpendicular to $\boldsymbol{PD}$) are generated by sweeping a parting loop edge onto a projection plane (PP) by the operation: $f_e = sweep(e, \boldsymbol{PD}, PP)$, where $e(V_1, V_2)$ is an edge of the loop and f_e is the generated parting face. Suppose the projected vertices of V_1 and V_2 into PP are V_1' and V_2'. If $V_1 = V_1'$ and $V_2 = V_2'$, f_e is null; otherwise, f_e is the face formed by V_1V_2, V_2V_2', $V_2'V_1'$, and $V_1'V_1$. Second, horizontal parting faces (normals parallel to $\boldsymbol{PD}$) are generated by tessellating polygons that are defined by portions of parting loops in PP and the projected edges from the *sweep*( ) operations. An additional assumption is that $\boldsymbol{PD}$ is along the part's Z axis.

Given this approach, specific parting face generation criteria for Problem MPC can be presented.

4.2 Generation Criteria.

The three criteria we identify are given as follows.

Criterion 1: Disassemblable in direction $\boldsymbol{PD}_i$.

To disassemble mold piece M_i in $\boldsymbol{PD}_i$, all the faces of M_i should be able to be swept in the parting direction $\boldsymbol{PD}_i$ without interference with other faces. Consequently, each generated parting face $F_{Parting\_Mi}$ should satisfy: *normal* $(F_{\text{Parting\_Mi}}) \cdot \boldsymbol{PD}_i \geq 0$, where $\cdot$ is the vector dot product.

Criterion 2: Flatness of parting faces.

It is desired to have a planar parting surface to decrease manufacturing complexity and to increase the shut-off force in the injection molding process for less material flash. One way to measure the flatness of parting faces is based on the sum of face areas: $\rho(F) = \sum_{i=1}^{k} area(F_i)$. With the same boundary loops, smaller $\rho(F)$ leads to flatter parting faces. For example, comparing the parting faces of the two mold pieces shown in Figs. 3(b)) and (c), we can see $\rho(F_{Parting\_2}) < \rho(F_{Parting\_1})$.

However, the minimum sum of parting face areas cannot be calculated during parting face generation, since these faces are unknown. Instead, we use three heuristics that are listed below based on their priorities (from high to low).

1. Make as many parting faces parallel or perpendicular to parting direction ($\boldsymbol{PD}_i$) as possible. Flat parting surfaces are generally easier to fabricate. The best surface finishes on parts made by RP machines are achieved when the surface is horizontal or vertical in the machine [21]. Additionally, flat parting surfaces help minimize flash during molding.
2. Generate the smallest number of parting faces. Our approach is to find the smallest number of projection planes in the sweeping process. With fewer projection planes used, parting faces will have a smaller number of faces, and generally look "flatter."
3. Use a projection plane that contains as much of the parting loop as possible and that minimizes the total projection distance of the edges in the parting loop.

Criterion 3: Maintain a minimum strength of mold pieces.

Theoretically only one projection plane is sufficient for criteria 1 and 2. However, this may lead to some mold pieces that are theoretically correct but infeasible to fabricate. For example, it is possible to form narrow "slivers" in the mold when projecting parting loop edges. To identify and prevent such situations, we determine the projection range of each parting edge, then classify these edges and ranges. Alternative parting planes can be evaluated based on the classifications. This is presented next.

4.3 Projection Ranges.

A part P with two regions R_1 and R_2 is shown in Fig. 6(a). The parting loop related to the two regions has a parting edge E_1. Suppose only one projection plane $Plane_1$ is used. The parting face related to E_1 will be $F_{Parting1}$. The generated mold piece M_1 will have two small features, which are easily broken (Fig. 6(b)). In some extreme cases, e.g., if the normal of F_1 is perpendicular to $\boldsymbol{PD}_1$, mold pieces cannot be fabricated since $F_{Parting1}$ overlaps with F_1.

Suppose parting direction $\boldsymbol{PD}$ is along the Z axis. In our approach, a **projection range** $PR[Z_1, Z_2]$ is defined as the range of Z coordinates of projection planes onto which a parting edge can be projected without passing through the part. A projection range is computed for each parting edge.

The projection range of a parting edge is calculated based on the position of its two neighboring faces. Each edge of a solid body has two coedges $\boldsymbol{CE_1}$ and $\boldsymbol{CE_2}$ [20]. Suppose two neighboring faces F_1 and F_2 with normals $\boldsymbol{n_1}$ and $\boldsymbol{n_2}$ belong to regions R_1 and R_2, respectively (Fig. 7(a)). They define a parting edge $E_{Parting}$ whose maximum and minimum Z values are $Z_{\max E}$ and $Z_{\min E}$. The coedge of $E_{Parting}$ in F_1 is $\boldsymbol{CE_1}$, while the coedge in F_2 is $\boldsymbol{CE_2}$. For a parting direction $\boldsymbol{PD_1}$ and a projection plane PP higher than $Z_{\min E}$, the parting face $F_{Parting\_up} = sweep(E_{Parting}, \boldsymbol{PD_1}, PP)$ should form an angle γ_1 with F_1 that is greater than some minimum, γ_s, that reflects mold strength (Fig. 7(b)). Otherwise, the generated mold pieces will be too weak. Similarly $F_{Parting\_down} = sweep(E_{Parting}, -\boldsymbol{PD_1}, PP)$ should form an angle γ_2 greater than γ_s with F_2 for a projection plane PP lower than $Z_{\max E}$ (Fig. 7(c)).

(a) If $E_{Parting}$ is not along $\boldsymbol{PD_1}$, then the projection range of a parting edge can be classified into four categories:

1. Up Range: if $\gamma_1 \geq \gamma_s$ and $\gamma_2 < \gamma_s$, edge $E_{Parting}$ has a projection range $[Z_{\max E}, +\infty)$;

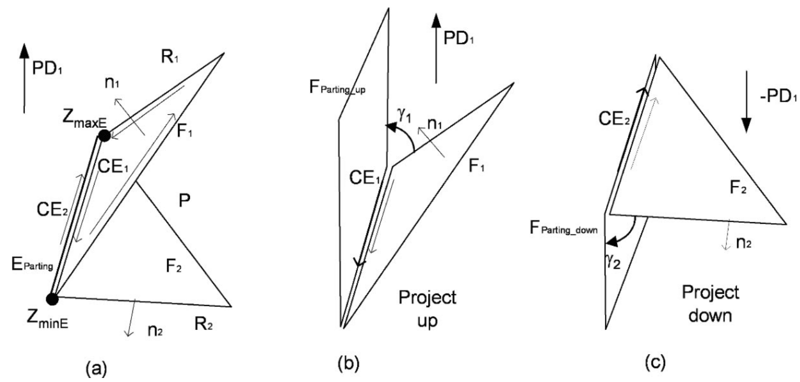


Fig. 7 Approach to compute projection range of parting edge

2. Down Range: if $\gamma_1 < \gamma_s$ and $\gamma_2 \geqslant \gamma_s$, edge $E_{Parting}$ has a projection range $(-\infty, Z_{\min E}]$;
3. Both Range: if $\gamma_1 \geqslant \gamma_s$ and $\gamma_2 \geqslant \gamma_s$, edge $E_{Parting}$ has a projection range $(-\infty, +\infty)$;
4. Null Range: if $\gamma_1 < \gamma_s$ and $\gamma_2 < \gamma_s$, edge $E_{Parting}$ has a projection range Z_E if $Z_{\max E} = Z_{\min E} = Z_E$; otherwise the projection range is null;

(b) If $E_{Parting}$ is in the direction of $\boldsymbol{PD_1}$, its projection range is determined by its two neighboring parting edges NE_1 and NE_2 and, again, can be classified into the same four categories with an additional category:

Limited Range: edge $E_{Parting}$ has projection range $[Z_1, Z_2]$.

Consequently, projection ranges can be computed for each parting loop related to region R_i and $\boldsymbol{PD}_i$. For example, for the parting loop shown in Fig. 8(a), the projection ranges are all Down Ranges $(-\infty, Z_E]$ (part from Fig. 3). In comparison, the projection ranges of the parting loop PL_2 shown in Fig. 8(b) are all Up Ranges $[Z_{\max E}, +\infty)$, for the part from Fig. 5.

The edge loop $b(F_{Mold\ cavity\_MPi})$ in Eq. (1) is unspecified in Problem MPC. In the beginning of the process, we use a temporary loop defined by the intersection of the mold cavity with a Z plane which contains the most vertices of the outer parting loop of $b(F_{Region\_MPi})$. The selection of the beginning loop will not affect the results of generated parting faces (refer to Example 5.1). Based on the criteria, we define the Parting Face Generation problem for Problem MPC as follows.

Problem PFG: Parting Face Generation. Suppose a projection direction $\boldsymbol{PD}$ and several closed parting loops (PL_i) are given with their projection ranges. Generate faces $F_1 \sim F_m$ such that:

a. The edge loops PL_i are the boundaries of face set $\{F_1, F_2, \ldots, F_m\}$;
b. $normal(F_i) \cdot \boldsymbol{PD} \geqslant 0$ $(1 \leqslant i \leqslant m)$ (criterion 1);
c. Suppose faces $\{F_k\}$ are the faces whose normals are in direction $\boldsymbol{PD}$. The number of projection planes which contain $\{F_k\}$ is minimum (criterion 2);
d. Suppose $\{F_n\}$ are the faces whose normals are perpendicular to direction $\boldsymbol{PD}$ and are generated by $sweep(e_n, \boldsymbol{PD}, PP)$. The projection plane PP is within the projection range of all e_n (criterion 3).

Our method for Problem PFG is presented next.

5 Parting Face Generation Method

Three types of edges may exist in an edge loop: *part edges, mold cavity edges*, and *insert edges*. Part edges come from the region faces of the injection molded part. Their projection ranges are defined in Section 4. *Mold cavity edges* come from the intersection of the mold cavity with parting faces. *Insert edges* are added during the parting face generation process. The projection ranges of *mold cavity edges* and *insert edges* are *Both Ranges*.

5.1 Parting Face Algorithm. For each parting loop PL (except PL_0=the parting loop in the outside surface of the mold cavity (loops **a** in Fig. 5)) and related projection ranges, the basic steps of the parting face generation process include: identify a parting plane based on the projection ranges of all unprocessed edges; sweep edges into the plane to generate shut-off faces; add *insert edges* in PP to form 2D polygons; triangulate 2D polygons; form new edge loops from unprocessed edges and new edges defined by the generated parting faces. This process can be stated more formally as:

Algorithm: *Parting_Face_Generation*

Input: One parting loop PL, Projection Ranges PR for each edge in PL.

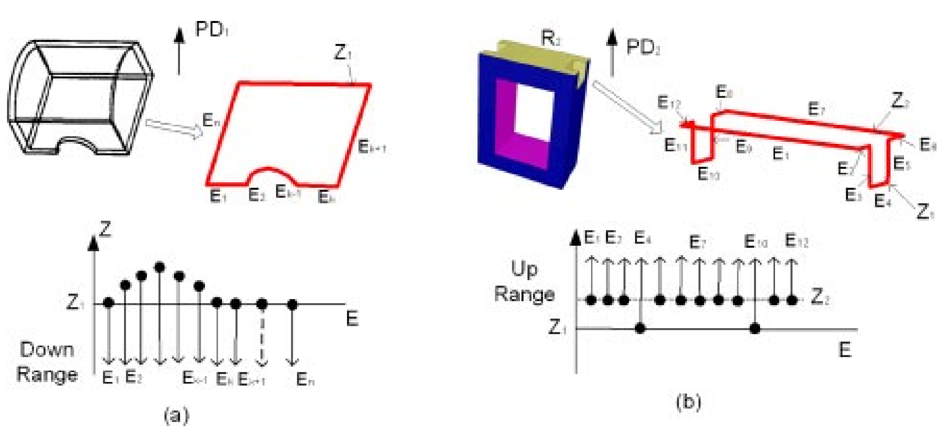


Fig. 8 Projection ranges of two parting lines

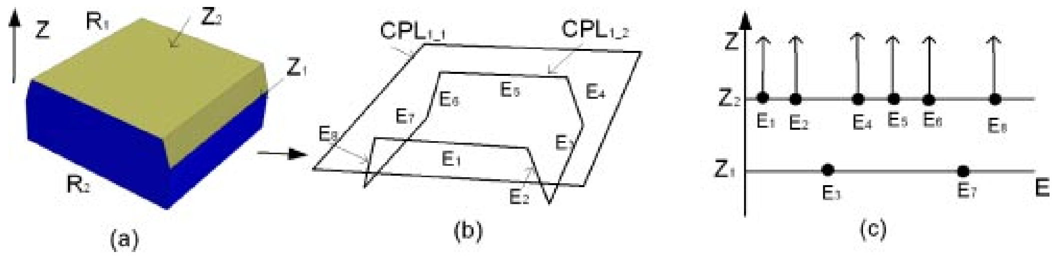


Fig. 9 An outside parting face example

Output: Set of parting faces, $F_{parting}$.
1. $F_{parting}$=NULL
2. *elist*=edges in PL or $PL+PL_0$ if PL is the outer parting loop
3. while *elist* contains one or more part edges // *elist* is list of unprocessed edges
4. form edge loops from *elist*
5. for each loop with one or more part edges
6. PP=identify projection plane from PR of part edges
7. L_{p}=edges in PL that are in PP
8. L_{o}=edges in PL that are not in PP nor vertical to PP
9. F_{so}=create shut-off faces using *sweep*( ) for edges of L_{o}
10. E_{p}=edges from F_{so} that are in PP
11. E_{v}=edges from F_{so} that are vertical to PP
12. For each F_{so} created by a projectable part edge or insert edge
13. $F_{\mathrm{parting}}+=F_{\mathrm{so}}$
14. For each projectable edge e of L_O
15. $elist=elist-e+E_p(e)+E_v(e)$ // edges of E_p and E_v that are related to e
16. $L_{\mathrm{p}}+=E_{\mathrm{p}}$ // Add insert edges if necessary for non-closed parting loop
17. for each dangling vertex in L_{p}
18. add *insert edge* e_{in} that connects PL or PL_0
19. e_{m_in}, e_{m_Out}=split edge e_{m} in PL or PL_0 and classify segments by loop directions
20. $L_{\mathrm{p}}+=e_{\mathrm{in}}+e_{m\_in}$,
21. $elist=elist-e_m+e_{m\_in}+e_{m\_Out}+\{\text{coedge of } e_{\mathrm{in}}\}$
22. form polygon(s) from edges in L_{p}
23. $F_{\mathrm{parting}}+=$triangulate polygon(s)
24. $elist=elist-E_{polygon(s)}$
25. delete edges in *elist* which form a pair of coedges // they are already correct in topology.

To find a projection plane in Step 6, a three step process is used. First, all overlaps are determined among the projection ranges of edges in a parting loop, effectively partitioning the edges. Second, the partition that contains the maximum number of edges is selected. This corresponds to heuristic 3 of Criterion 2 from Section 4.2. Third, a specific projection plane is selected that minimizes the total swept face area of edges in the selected partition. Assume the edges in the selected partition are $E_i(1\leqslant i\leqslant k)$. The two vertices that bound edge E_i are $V_{i1}(X_{Vi1},Y_{Vi1},Z_{Vi1})$ and $V_{i2}(X_{Vi2},Y_{Vi2},Z_{Vi2})$. The Z coordinate of the best parting plane (PP) can be computed as:

$$Z_{parting\_plane}=Z_i:\min\sum_{j=1}^{k}\left[|Z_{V_{j1}}-Z_i|+|Z_{V_{j2}}-Z_i|\right]\cdot\sqrt{(X_{V_{j1}}-X_{V_{j2}})^2+(Y_{V_{j1}}-Y_{V_{j2}})^2} \tag{2}$$

Steps 17–21 require explanation. Each edge has an attribute *Projectable* that is updated to identify if it can be projected into the selected projection plane. If all projectable edges do not form a polygon, *insert edges* must be added to connect the projectable edges at their dangling vertices. Suppose e_i is a projectable edge while e_{i+1} is a non-projectable edge. Vertex V_i is their common (dangling) vertex. Our approach to find an insert edge e_{in} is given as follows. First, create half lines l_i from V_i to infinity, where l_i bisects the angle formed by e_i and e_{i+1} and l_i passes through the mold material (i.e., l_i is on the material side of e_i and e_{i+1}). Check the intersection of the l_i with all edges in PL and PL_0. Suppose e_m is the edge with the closest intersection vertex V_{em}. Then, (1) if e_m is projectable, split e_m into two edges (e_{m1},e_{m2}) by V_{em}, and generate an insert edge $e_{in}(V_i,V_{em})$ if V_i is the end vertex of e_i, or $e_{in}(V_{em},V_i)$ if V_i is the start vertex of e_i; (2) if e_m is not projectable, iterate e_{m+j} ($j=1,2,\ldots$) until the next pro-

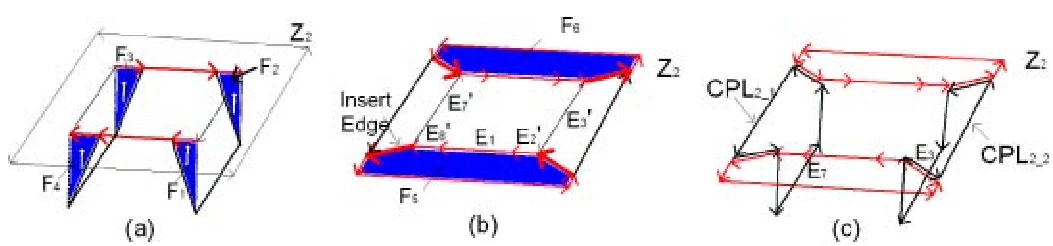


Fig. 10 Parting face generation process (Iteration 1)

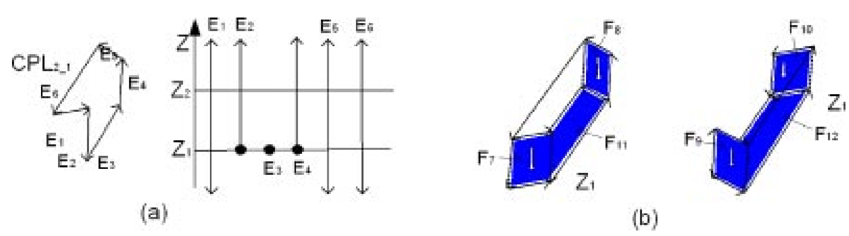


Fig. 11 Parting face generation process (Iteration 2)

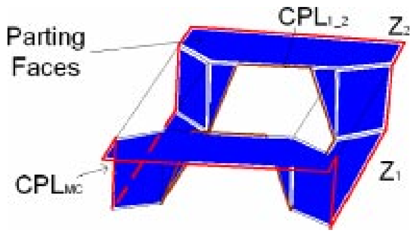


Fig. 12 Edge loop in the mold piece

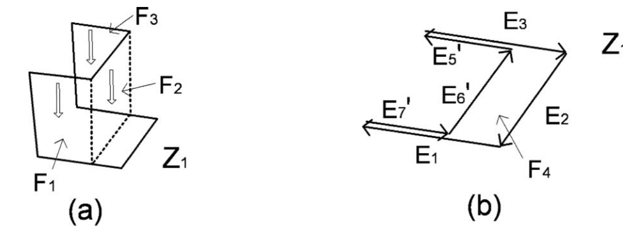


Fig. 15 Parting face generation process

jectable edge is found, compute a feasible intersection vertex if possible, and generate an insert edge e_{in}. The coedge of e_{in} is also added to the unprocessed edge loops. The two split edges, e_{m1} and e_{m2}, are classified as e_{m_in} and e_{m_out} such that e_{m_in} is in the same direction as L_p, while e_{m_out} is in the opposite direction. The examples in this section should clarify this presentation.

After closed edge loops are formed with all their projectable edges, parting faces in the projection plane can be generated by triangulating the 2-D polygons formed by the closed edge loops. This topic has been studied extensively (e.g., see [22–24]).

The computational complexity of algorithm *Parting_Face_Generation* will be analyzed briefly. Assume that there are n_{pe} edges in all parting loops. Then, the complexity of the algorithm is $O(n_{pe}^2)$ in the worst case. This can be realized by estimating the number of times each of the three nested loops will execute. Note that the number of parting loop edges, n_{pe}, is typically much less than the number of edges in the part. The number of dangling vertices is related to the number of loops formed in the previous step which, again, is much smaller than n_{pe}.

Algorithm *Parting_Face_Generation* satisfies the three criteria presented in Section 4.2. The first criterion (disassemblable in $\boldsymbol{PD}_i$) is ensured due to the treatment of projection ranges. The second criterion (flatness of parting faces) has three sub-parts. First, only parting faces are added that are parallel or perpendicular to $\boldsymbol{PD}_i$. Second, the number of projection planes is minimized (although we cannot claim a global minimum) in part by selecting planes that contain as much of the parting loop as possible. Third, the final selection of parting planes is based on minimizing projection distances. The third criterion (maintain minimum strength) is ensured by enforcing a minimum angle, γ_s, between part faces and projected faces during the determination of projection ranges.

Both outside and inside parting faces can be generated via the above process. We use two examples to illustrate these basic steps.

5.2 Example 1: Outside Parting Faces With Insert Edges. A molded part with two regions (R_1 and R_2) is shown in Fig. 9(a). The related parting loop is $CPL_{1\text{-}2}$, and the edge loop in the mold cavity is $CPL_{1\text{-}1}$ (Fig. 9(b)). Edges E_3 and E_7 of $CPL_{1\text{-}2}$ are *Null Ranges* while all other edges are *Up Ranges*. The projection ranges of $CPL_{1\text{-}2}$ are shown in Fig. 9(c).

During the first iteration through the *while* loop (Step 3), the projection plane Z_2 is selected based on the projection ranges. By sweeping the edges of CPL_{1_2}, four parting faces $F_1 \sim F_4$ are generated (Fig. 10(a)). In the projected edge loop in plane Z_2, edge E_8' is projectable, while E_7' is not projectable. Therefore an insert edge is added (extension of E_8'). Similarly, additional insert edges are added for edges E_2', E_4', E_6' and E_8' (Fig. 10(b)). Two polygons (F_5,F_6) are formed in plane Z_2, which are then trian-

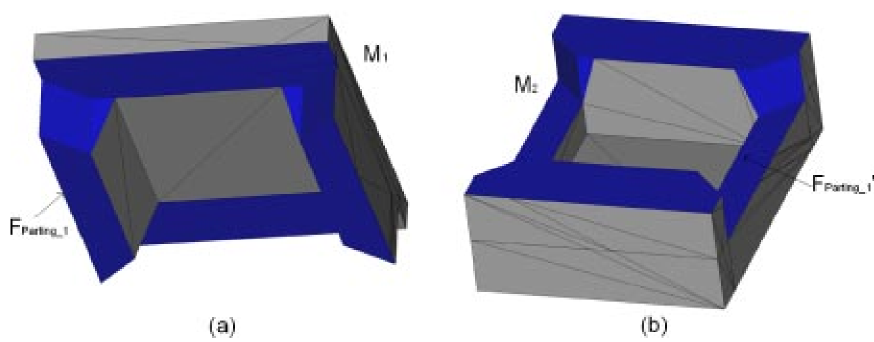


Fig. 13 Mold pieces related to parting faces

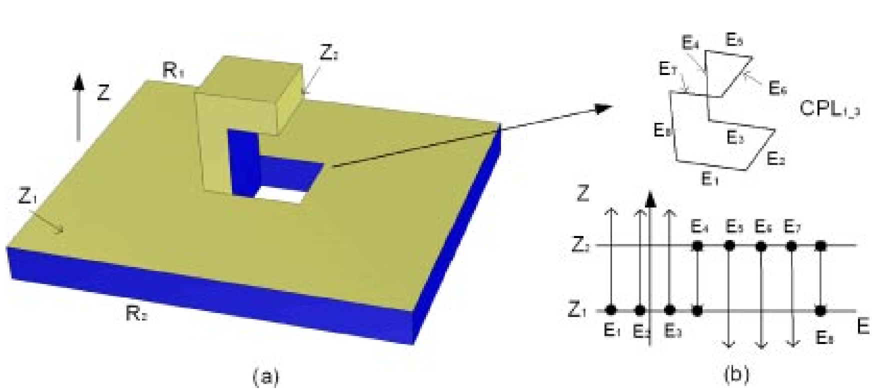


Fig. 14 An inside parting face example

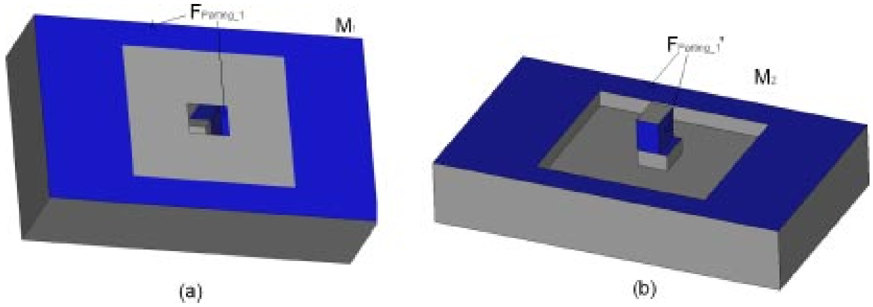


Fig. 16 Mold pieces related to parting faces

gulated. Eight edges of the boundary of face set ($F_1 \sim F_6$) are added to the unprocessed edge loops (Fig. 10(c)). Now the unprocessed edge loops are CPL_{2_1} and CPL_{2_2}.

Edge loops CPL_{2_1} and CPL_{2_2} are processed during the second iteration of the *while* loop. Based on projection ranges (Fig. 11(a)), plane Z_1 is selected as the projection plane. Parting faces $F_7 \sim F_{10}$ are generated in the projection process, while F_{11} and F_{12} are formed by edge loops, then triangulated (Fig. 11(b)).

After all the edges are processed, the generated parting faces ($F_1 \sim F_{12}$) form two closed edge loops CPL_{MC} and CPL_{1_2} (Fig. 12). Loop CPL_{MC} is different from CPL_{1_1} which is initially set as the mold cavity loop. Based on CPL_{MC}, we can split the mold cavity into two face sets and generate two mold pieces M_1 and M_2 (Fig. 13) related to the parting faces.

5.3 Example 2: Inside Parting Faces With Different Projection Planes. A second molded part is shown in Fig. 14(a). In this example, we will focus only on the inside parting faces. Related to regions R_1 and R_2, an inner parting loop is CPL_{1_3} (Fig. 14(b)). Its projection ranges are also shown in the figure. Edges $E_1 \sim E_3$ have *Up Ranges*, edges E_4 and E_8 have *Limited Ranges*, and Edges $E_5 \sim E_7$ have *Down Ranges*.

Based on our process, the projection range of CPL_{1_3} is (Z_1, Z_2), and the selected projection plane is Z_1 plane since it contains more edges in CPL_{1_3}. Sweeping the edges of CPL_{1_3} into Z_1 generates parting faces $F_1 \sim F_3$ (Fig. 15(a)). All the projected edges are projectable. They define a polygon that forms face F_4 (Fig. 15(b)). Related to the parting faces, two mold pieces M_1 and M_2 are generated as shown in Fig. 16.

Instead of plane Z_1, we could have used plane Z_2 in the above process. However the total area of the generated parting faces will be larger.

6 Mold Piece Construction Algorithms

Each region specified in a molded part is related to one mold piece. The order in which the regions are processed, and mold pieces are generated, affects the geometries of resulting molds. We utilize two heuristics to order regions, based on industry best practice [19]: (1) generate mold pieces for regions with parting directions along the main parting direction first [2]; (2) generate mold pieces for regions with larger volumes (calculate by bounding box) first. Accordingly, regions are rank ordered as R_1, $R_2, \ldots, R_{n-1}$. The overall mold piece construction algorithm is given below. Its complexity is linear in the number of regions.

Algorithm: *Multi_Mold_Piece_Generation*

Input: Part P with Regions R_i $(1 \leqslant i \leqslant n)$ in direction $\boldsymbol{PD}_i$, and mold cavity $MC = M - P$.

Output: Set of mold pieces M_i $(1 \leqslant i \leqslant n)$.

1. rank region number *i* according to the main parting direction and region volumes;
2. for $1 \leqslant i \leqslant n-1$;
3. generate closed parting loops (PL_j) and projection ranges (PR_j) for region R_i;
4. Two_Mold_Piece_Generation $(PL_j, PR_j, \boldsymbol{PD}_i, MC, M_1', M_2')$;
5. $M_i \leftarrow M_1'$;
6. if $i = n-1$, $M_n \leftarrow M_2'$;
7. else, $MC \leftarrow M_2'$.

The algorithm to construct two-piece molds is:

Algorithm: *Two_Mold_Piece_Generation*

Input: Closed parting loops (PL_j) and related projection ranges (PR_j); parting direction $\boldsymbol{PD}_i$;

Boolean mold cavity MC with regions R_k $(k = i, \ldots, n)$.

Output: Mold pieces M_1 and M_2.

1. rotate MC to align $\boldsymbol{PD}_i$ with the Z axis;
2. classify PL_j into an outer loop PL_1 and several inner loops PL_k;
3. initialize face set $F_{Parting} \leftarrow$NULL;
4. find plane Z_0 which contains the maximum number of vertices of PL_1;
5. generate loop PL_0 by intersecting MC with Z_0;
6. *Parting_Face_Generation* $(PL_1, F_{Parting})$; // generate $PF_{outside}$;
7. generate loop PL_{MC} as the outside boundary of face set $F_{Parting}$;
8. for each inner loop PL_k;
9. *Parting_Face_Generation* $(PL_k, F_{Parting})$; // *generate* PF_{inside};
10. set $F'_{Parting} \leftarrow$copy_reverse_faces $(F_{Parting})$; // faces in same position but reverse directions;
11. use PL_{MC} to split the mold cavity into face sets F_{up} and F_{down};
12. set $F'(R_1) \leftarrow$copy_reverse_faces (R_i); // faces in same position but reverse directions;
13. set $F'(R_2) \leftarrow$copy_reverse_faces (R_k) $(k = i+1, \ldots, n)$;
14. generate $M_1 \leftarrow$reverse_glue $(F_{Parting}, F_{up}, F'(R_1))$;

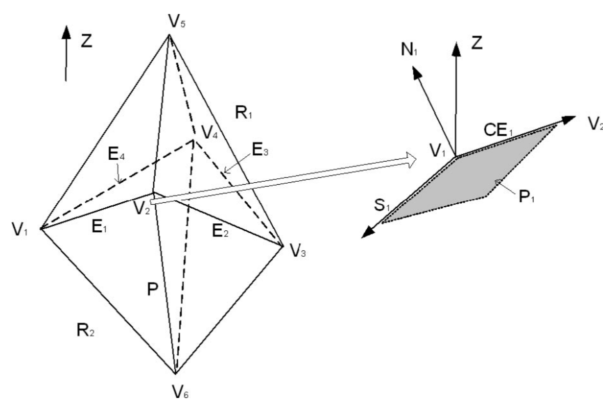


Fig. 17 An example of null projection range

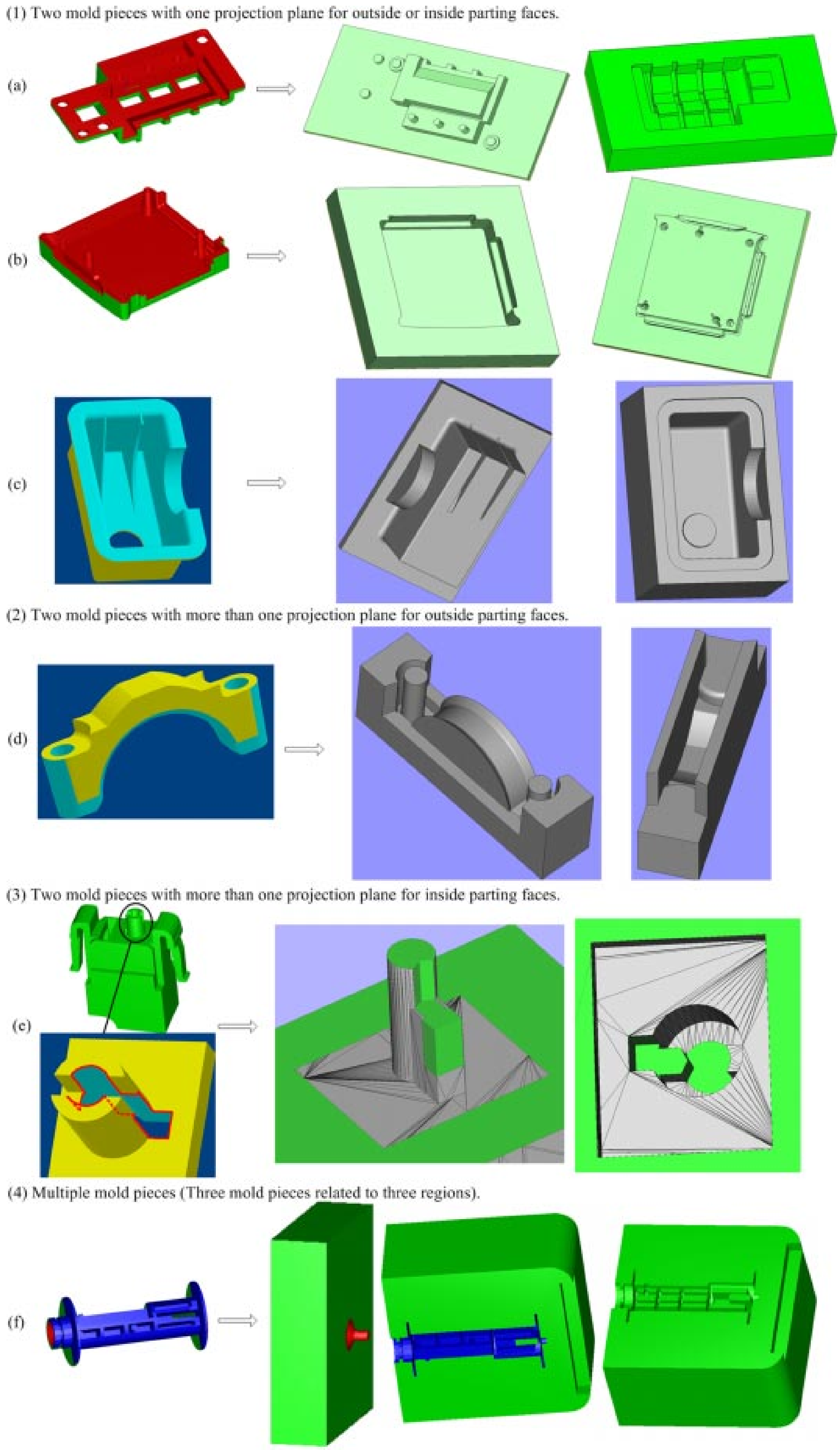


Fig. 18 Results of tested industrial parts (part regions with constructed molds)

15. generate M_2←reverse_glue $(F'_{Parting}, F_{down}, F'(R_2))$.

Parting face generation in Steps (6) and (8) was discussed in detail in Sections 4 and 5. Algorithm *Parting_Face_Generation* is invoked in each Step. For Step (11), we utilize the approach of [25] based on retriangulation to split any surface by a closed edge loop in the surface. Face set F_{up} contains faces that are "above" PL_{MC} (larger Z coordinate values), while F_{down} contains faces that are below PL_{MC}. In Steps (14) and (15), mold piece solid bodies are generated based on the reverse glue operation that was introduced in Section 3. The steps of the reverse glue operation can be presented more precisely, given all necessary face sets, for a single mold piece (M_1):

1. Create a new body M_1 and add faces $F_{Parting}$, F_{up}, and $F'(R_1)$ to M_1;
2. Identify all common edges of $F_{Parting}$ and F_{up}; swap their partner coedges to seal the common edges;
3. Identify all common edges of $F_{Parting}$ and $F'(R_1)$; swap their partner coedges to seal the common edges.

As can be seen, our algorithms are very efficient. The computational complexity of algorithm *Two_Mold_Piece_Generation* is $O(n_{pe}^2 + n_e^* n_{pe}) = O(n_e^* n_{pe})$ (since $n_e \geqslant n_{pe}$), where n_e is the number of edges in the part. As noted, our algorithms do not rely on Boolean or face sweeping operations. Additional analysis and results are presented in [26].

Discussion: Edges With Null Range and Approach Based on Extending Parting Lines. Among the five types of projection ranges, a special case of *Null Range* needs further discussion. Part P in Fig. 17 has six vertices ($V_1 \sim V_6$) and parting edges $E_1 \sim E_4$ based on direction Z. Suppose the parting edges are all *Null Ranges*; hence, no projection plane Z can be identified.

The approach presented in Section 5 to identify a projection plane from projection ranges can be extended. If the current projection ranges are null, take the first edge E_1 in the loop. Suppose vector $\boldsymbol{CE_1} = V_2 - V_1$, then $\boldsymbol{S_1} = \boldsymbol{CE_1} \times \boldsymbol{Z}$, $\boldsymbol{N_1} = \boldsymbol{S_1} \times \boldsymbol{CE_1}$, where $\times$ is the cross product. See Fig. 17. Therefore a plane P_1, which passes through E_1 and has normal $\boldsymbol{N_1}$, can be used as the projection plane in the following steps in Section 5.

Note that when $ZV_1 = ZV_2$, $\boldsymbol{N_1}$ is actually the same as $\boldsymbol{Z}$. Therefore P_1 is a Z plane. Also, using projection planes with $(\boldsymbol{CE_i} \times \boldsymbol{Z}) \times \boldsymbol{CE_i}$ as coedge normals is actually the approach based on extending parting lines [8]. Therefore the approach based on extending parting lines is a special case in our approach that only handles edges with null projection ranges.

7 Implementation and Tested Cases

We implemented the algorithms in an experimental system (Rapid Tooling Mold Design System) with Microsoft Visual C++6.0. The system was based on ACIS6.2, a 3D geometric modeler provided by Spatial Technology, Inc. We extended the system to enable the algorithms to directly manipulate triangles. Therefore, the inputs and outputs of the system can be STL files. Several initial steps of the system were described in [2]. We used our implementation to design molds to fabricate prototype injection molded parts of different complexity. In addition to the test cases shown in Figs. 5, 10, and 15, we tested several industrial parts by generating molds automatically using our system (Fig. 18). Among the tested parts, cases (a)~(c) are examples of two mold pieces with only one projection plane required; case (d) is an example which requires two projection planes; case (e) is an example with a complicated inner parting loop; and case (f) is an example of the multi-piece mold design with three regions. For each part, the part with regions highlighted, the given mold cavity, and the generated mold pieces are shown. The running time of our system for all cases was very satisfactory. None of them took more than 45 seconds on a PC with a 700 MHz Pentium III processor.

Fig. 19 Stereolithography mold pieces and camera roller prototypes

In addition, we tested our mold design results using the direct AIM tooling technique for the part shown in Fig. 18(f). The mold pieces were built in a SLA-3500 machine. The Stereolithography mold inserts were then installed in an injection molding machine to produce functional prototypes. A photo of the mold and molded prototypes is shown in Fig. 19.

8 Conclusions

In this paper we presented a novel method for the automated construction of multi-piece molds based on parting face generation and the reverse glue operation. Our method is general for both inner and outer parting loops, and two-piece and multi-piece mold constructions. The method utilizes planar parting surfaces, but can generate non-planar parting surfaces if necessitated by part geometry. We have demonstrated the mold construction method on molds to be fabricated by RP technologies, although it can also be applied to conventional machined molds.

The generation of parting faces is a geometric reconstruction problem, which is defined as Problem PFG in this paper. Different parting faces will lead to different mold piece designs. We proposed three criteria for parting face generation, which include disassemblability (undercut-free), flatness, and mold strength. Related to the criteria, a new representation, projection range, was proposed and incorporated into the new parting face generation algorithm. Multiple-piece mold and two-piece mold construction algorithms were presented that utilized the parting face generation algorithm and the reverse glue operation. Complexity analysis showed that the algorithms were quadratic in the number of part edges in the worst case. Also, the operations are not time consuming, in contrast to operations used in other mold construction approaches that utilize Boolean or face sweeping operations.

We implemented the algorithms in an experimental system and tested parts of various complexity. Based on our test results, we believe our mold construction algorithms are very efficient. The test cases executed quickly (in under 45 seconds) even for complex parts. The system provides users with instantaneous visual feedback on the mold design results, which is very important for design-for-manufacturing. The results also validate the efficacy and robustness of our approach.

Some limitations of our approach should be highlighted. First, our methods require faceted part models. Second, only a local disassemblability evaluation is performed when generating parting faces (see [3] for a global disassemblability method). Third, we have limited heuristics for generating mold designs that conform to industry best practice. Each of these limitations will be addressed in future work.

Acknowledgment

We gratefully acknowledge the support from NSF grant DMI-9618039 and from the member companies of the Rapid Prototyping and Manufacturing Institute at Georgia Tech. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. Also, we acknowledge the usage of Systems Realization Laboratory facilities.

Nomenclature

$b(\ )$	=	Boundary of face set = loop of edges and vertices
CE	=	Coedge
CLP	=	Closed Parting Loop
E	=	Edge
F	=	Face (of part, region or mold)
F_{Parting}	=	Parting Face
$F(\ )$	=	Face set
M, MP	=	Mold piece
MC	=	Mold cavity
P	=	Part
$\mathbf{PD}$	=	Parting Direction
PF	=	Parting Face
PL	=	Parting Line, Loop
PP	=	Parting Plane, Projection Plane
PR	=	Projection Range
R	=	Region of part that corresponds to one mold piece
γ_s	=	minimum angle for mold strength
Bold Symbols	=	vectors

References

[1] Jacobs, P., 1996, *Stereolithography and Other RP&M Technologies: From Rapid Prototyping to Rapid Tooling*, Society of Manufacturing Engineers, Dearborn, MI, Chap. 10.

[2] Chen, Y., and Rosen, D. W., 2002, "A Region Based Method to Automated Design of Multi-Piece Molds With Application to Rapid Tooling," ASME J. Comput. Info. Sci. Eng. **2**(2), pp. 86–97.

[3] Huang, J., Gupta, S. K., and Stoppel, K., 2003, "Generating Sacrificial Multi-Piece Molds Using Accessibility Driven Spatial Partitioning," accepted in Computer Aided Design.

[4] Majhi, J., Gupta, P., and Janardan, R., 1999, "Computing a Flattest, Undercut-Free Parting Line for a Convex Polyhedron, With Application to Mold Design," Computational Geometry, **13**(4), pp. 229–252.

[5] Tan, S. T., Yuen, M. F., Sze, W. S., and Kwong, K. W., 1990, "Parting Lines and Parting Surfaces of Injection Moulded Parts," IMechE Part B: J. of Engineering Manufacture, **204**, pp. 211–221.

[6] Shin, K. H., and Lee, K., 1993, "Design of Side Cores of Injection Moulds From Automatic Detection of Interference Faces," Journal of Design and Manufacturing, **3**, pp. 225–236.

[7] Serrar, M., 1995, "Applying Moldability Evaluation to Solid Models," Ph.D. thesis, Rensselaer Polytechnic Institute, Troy, NY.

[8] Nee, A. Y. C., Fu, M. W., Fuh, J. Y. H., Lee, K. S., and Zhang, Y. F., 1998, "Automatic Determination of 3-D Parting Lines and Surfaces in Plastic Injection Mould," CIRP Ann., **47**(1), pp. 95–98.

[9] Priyadarshi, A., and Gupta, S. K., 2003, "Geometric Algorithms for Automated Design of Multi-Piece Permanent Molds," accepted in Computer-Aided Design.

[10] Ravi, B., and Srinivasan, M. N., 1990, "Decision Criteria for Computer-Aided Parting Surface Design," Comput.-Aided Des., **22**(1), pp. 11–18.

[11] Hui, K. C., and Tan, S. T., 1992, "Mould Design With Sweep Operations—A Heuristic Search Approach," Comput.-Aided Des., **24**(2), pp. 81–91.

[12] Urabe, K., and Wright, P. K., 1997, "Parting Direction and Parting Planes for the CAD/CAM of Plastic Injection Molds," *ASME Design for Manufacturing Conference*, Sacramento, CA, DETC97/DFM-4336.

[13] MoldWizard, Unigraphics Corp. (http://www.ugs.com).

[14] Moldplus, Moldplus, SA. (http://www.moldplus.com).

[15] QuickSplit, Cimatron, Ltd. (http://www.cimatron.com).

[16] Magics RP, Materialise. (http://www.materialise.com).

[17] IMOLD, Manusoft Tech. (http://www.manusoft-tech.com/index.shtml).

[18] Zhang, Y. F., Fuh, J. Y. H., Lee, K. S., and Nee, A. Y. C., 1999, "IMOLD: An Intelligent Mould Design and Assembly System," *Computer Applications in Near Net-Shape Operations*, A. Y. C. Nee et al., eds., Springer-Verlag, Berlin, Chap. 7.

[19] Bralla, J. G., 1986, *Handbook of Product Design for Manufacturing*, McGraw-Hill, New York, USA, Chap. 5.4.

[20] Mäntylä, M., 1988, *An Introduction to Solid Modeling*, Computer Science Press, Maryland, USA, Chaps. 6 and 9.

[21] West, A. P., Sambu, S. P., and Rosen, D. W., 2000, "A Process Planning Method for Improving Build Performance in Stereolithography," Comput.-Aided Des., **33**(1), pp. 65–79.

[22] O'Rourke, J., 1998, *Computational Geometry in C* (2nd Edition), Cambridge University Press, UK, Chap. 1.

[23] Seidel, R., 1991, "A Simple and Fast Incremental Randomized Algorithm for Computing Trapezoidal Decompositions and for Triangulating Polygons," Computational Geometry: Theory and Applications, **1**(1), pp. 51–64.

[24] Held, M., 2001, "FIST: Fast Industrial-Strength Triangulation of Polygons," Algorithmica, **30**(4), pp. 563–596.

[25] Bruyns, C., and Senger, S., 2001, "Interactive Cutting of 3D Surface Meshes," Comput. Graph., **25**, pp. 635–642.

[26] Chen, Y., 2001, "Computer-Aided Design for Rapid Tooling: Methods for Mold Design and Design-for-Manufacture," Ph.D. thesis, Georgia Institute of Technology, Atlanta, GA.