

A Region Based Method to Automated Design of Multi-Piece Molds with Application to Rapid Tooling

Yong Chen

3D Systems, Inc.,
26081 Avenue Hall,
Valencia, CA 91355
e-mail: ChenY@3DSYSTEMS.COM

David W. Rosen*

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

Particularly for rapid tooling applications, delivering prototype parts with turn-around times of less than two weeks requires fast, proven mold design methods. We present a region-based approach to automated mold design that is suitable for simple two-piece molds (consisting of core and cavity), as well as molds with many additional moving sections. In our region-based approach, part faces are partitioned into regions, each of which can be formed by a single mold piece. The basic elements of our approach are concave regions (generalized pockets) and convex faces since these elements are central to the identification of regions. This paper focuses on the initial steps of automated mold design, including a problem formulation, methods for identifying the basic elements from part faces, and combining them into regions. By seeking to minimize the number of mold pieces, different partitions of faces into regions are explored until the smallest number of regions is found. During this process, a linear programming problem is adopted for finding a satisfactory parting direction of a region. Algorithms are presented for the region generating and combining process. Our approach is illustrated with several examples of industrial injection molded parts. [DOI: 10.1115/1.1505030]

1 Introduction

Many consumer products involve the design and fabrication of injection molded thermoplastic parts. Due to the interrelationships between part design, mold design, and fabrication process planning, the design of injection molded parts is a complex process with several iterations of prototype parts. As a result, the lead-time for prototypes can be a considerable portion of the whole design process time.

Rapid tooling, which uses a rapid prototyping technique to fabricate tools or patterns, can reduce tooling cost and time especially when only small volumes of a part are needed [1]. Figure 1 shows the main steps of a typical rapid tooling process, direct AIM (ACES Injection Molding), which makes use of the stereolithography (SLA) process. Molds can be classified into two-piece molds and multi-piece molds. Two-piece molds are the most commonly used molds because they are relatively easy to design and manufacture. Multipiece molds contain more than two mold pieces, which are required for many complex part shapes. In one type of multi-piece mold, each piece is hand-loaded into a mold base mounted on the injection molding machine platens. During material injection and part cooling processes, the molds are clamped into the holding device. Finally each piece can be hand-removed, if necessary, from the mold base to release the part. The above process is used only for producing small volumes of parts. One commercial example of multi-piece molds is from Space Puzzle Moulding[®] of Protoform GmbH (<http://www.protoform.de>). Combining multi-piece molding and Rapid Tooling techniques, it is possible to build injection molding tools for complex parts in a very short period of time. However, since multi-piece molds have more than one pair of opposite parting directions, it is more difficult and time-consuming to generate a good mold design. Particularly for rapid tooling applications, de-

livering prototype parts with turn-around times of less than two weeks requires fast, proven mold design methods.

Given the geometry of a part, depending on the selection of mold design variables (e.g., parting direction and lines), a different number of mold pieces may be required to form the part. It is desired to minimize the number of required mold pieces because fewer mold pieces reduces the tooling cost and simplifies the operation of the mold. Therefore in this paper we will consider the mold design problem for multi-piece mold design as:

Problem MD: Mold Design. *Given a solid part and a mold base, design the minimum number of mold pieces that can form the cavity of the part in the material injection process, and can be disassembled properly in the part ejection process.*

Mainly from the geometric perspective, a systematic method, Multi-Piece Mold Design Method (MPMDM), is developed to automate several important mold design steps, including selection of parting directions, parting lines, and parting surfaces, and construction of mold pieces. In this paper, we present the approaches of MPMDM in designing mold configurations for a given part. The construction of mold pieces based on the mold configurations generated in this paper is presented in [2].

2 Review of Related Work

The automation of mold design for injection molding has been studied in many publications. The current state of research on the automation of mold design mainly focuses on two-piece molds because they are more commonly used and relatively easier to design and manufacture than multi-piece molds. For the design of two-piece molds, works cover the determination of parting direction, parting line, parting surface, and undercut detecting individually (definitions of these terms can be found in mold design handbooks [3]). A systematic approach to consider all the above important considerations in a computer-aided mold design system is not found. In this section we will review related works on automatic mold configuration design in determining parting direction and parting lines.

2.1 Parting Direction. As an important step in the auto-

*Corresponding author.

Contributed by the Committee for publication in the JOURNAL OF COMPUTING AND INFORMATION SCIENCE IN ENGINEERING. Manuscript received November 2001; Revised June 2002. Associate Editor: P. Wright.

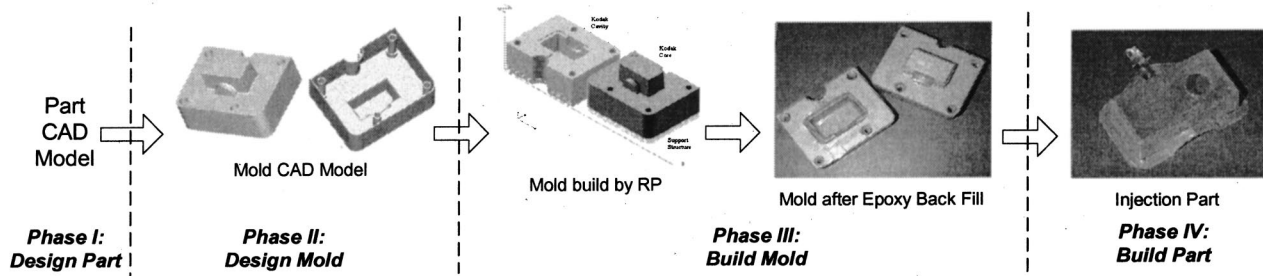


Fig. 1 Direct AIM tooling process

matic mold design process, the selection of parting direction has received much attention. In this section we review the proposed approaches from their basic elements and the synthesis methods.

2.1.1 Basic Element. There are three kinds of basic elements that are used in the determination of parting directions: pockets, features and faces.

1 Pocket: Chen et al. [4] first formulated the determination of parting directions as a visibility problem. Based on the visibility of faces, the notion of a pocket is presented for detecting global interference. For an object Ω , pockets of the object are the regularized difference between its convex hull $CH(\Omega)$ and Ω . Weinstein & Manoochehri [5] and Vijay et al. [6] presented approaches based on the same basic elements for the selection of parting directions.

2 Feature: Research results in feature modeling methods and feature recognition have been used in automatic mold design in several publications. Gu et al. [7] used a universal hint-based feature recognition algorithm to recognize all features of a molded part, such as holes, steps, pockets, protrusions, etc. Fu et al. [8] classified undercut features as Inside vs. Outside and Internal vs. External. Based on their feature characteristics and geometric entities (three-edge, four-edge and more than four-edge), algorithms to recognize these undercut types are presented. More recently, Yin et al. [9] and Ye et al. [10] presented a volume-based method and a hybrid method to recognize undercut features, respectively. Dhaliwal et al. [11] presented a novel approach for creating multi-piece sacrificial molds from feature-based part representations, such that each mold piece must be machinable using a 3-axis machine.

3 Face: Face is a basic element of a CAD model. It is also used in automatic mold design. Hui and Tan [12] heuristically generated candidate parting directions from normal vectors of planar faces and from center-lines of holes and bosses to evaluate the geometry of an undercut. Later, Hui [13] developed a partitioning scheme to subdivide the cavity solid of a component along a given direction. In the search for main and side core directions, the search space is the set of all normals to individual faces of the object and the opening of cavity solid. Urabe and Wright [14] selected three principal coordinate directions as candidate parting directions for calculating mold factors to determine a parting direction. Part faces are used in the calculation of mold factors. Also based on part faces, Lu and Lee [15] presented a three-dimensional ray-detection method to analyze the interference elements and release directions in die cast or injection molded components.

2.1.2 Synthesis Approaches of Basic Elements. All the methods used a similar synthesis process to evaluate a parting direction. That is,

(1) For each candidate direction \mathbf{d}_k , $V(\mathbf{d}_k) = \sum_i \text{weight_factor}_i$

$\times \text{mold_factor}_i$;

(2) Find parting direction $\mathbf{d}: V(\mathbf{d}) = \text{Min}[V(\mathbf{d}_k)]$.

In Step (1), mold factors are some criteria used to determine the

parting directions, which may include number of undercuts, projected area, draft angles, etc. Weight factors distinguish the importance of one mold factor relative to another. The assignment of these weight factors is based on how the corresponding mold factor affects the cost, quality, and productivity of the mold.

The methods to evaluate mold factors for different basic elements are summarized as follows.

(i) **Evaluate mold factor for pockets/features.**

In [4], candidate directions were the directions computed from the Visibility Maps of pockets. Only one mold factor, number of undercuts, was considered. Hence, the mold factor for each candidate direction is actually the number of V-maps that do not contain the candidate direction. In [5,9,15], a similar approach was used in evaluating parting directions. Besides number of undercuts, Gu et al. [7] also considered the projected area and thickness of the molded part, which were evaluated based on the bounding box of a part.

Since face connectivity and faces not belonging to pockets or features are not considered in these approaches, the combinations of pockets/features can be explored rather quickly, usually with the aid of a visibility map. However, problems may arise in constructing mold pieces for non-connected faces as shown in Section 3. Also, limitations may occur in considering more mold design knowledge by adding extra mold factors.

(ii) **Evaluate mold factor for all faces.**

In this approach, mold factors are evaluated for all faces, enabling more mold factors to be considered. Urabe and Wright [14] considered boxed area, projected area, number of non-hidden faces, number of undercuts and cone surfaces. Hui and Tan [12] considered number of undercuts and projected area. A more comprehensive decision model was presented in [16] for the selection of parting surfaces for casting parts that included: projected area, flatness, draw distance, draft, number of undercuts, volume of flash, and dimensional stability. Related approaches and expressions for most of these criteria were also presented.

However, because it is quite time-consuming to test the mold factors for all faces, only a small number of candidate parting directions or parting surfaces are considered in these approaches.

2.2 Parting Lines. There are fewer published works on the automatic determination of parting lines. Tan, et al. [17] proposed a parting line generation method for a triangular sub-division of the product model's surfaces. In the method, a draw direction is selected first. Then, the algorithm triangulates and classifies the surfaces into visible faces and invisible faces. Ravi and Srinivasan [16] proposed a sectioning and silhouette method for parting line generation that is capable of dealing with nonplanar parting surfaces. Wong and coauthors [18] used a slicing strategy to locate the parting lines of a product model along a draw direction. A recursive uneven slicing method was developed to locate several parting surfaces for further evaluation. The approach is primarily proposed to deal with free-form surfaces in product design. Majhi and coauthors [19] discussed the problem of computing an undercut-free parting line that is as flat as possible in mold design

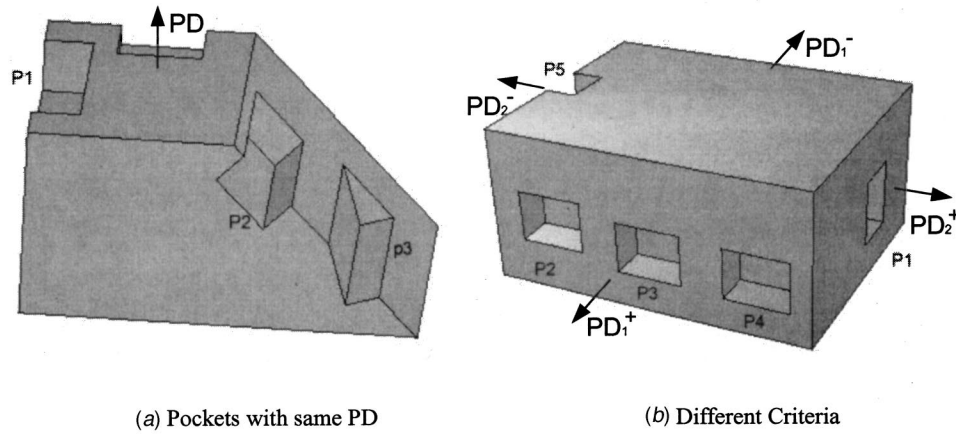


Fig. 2 Examples for the problem formulation

for a convex polyhedron. Two flatness criteria for parting lines were given and algorithms were presented to compute a parting line based on the criteria.

The above approaches all used part faces as the basic elements. In these approaches, either the parting direction was already set [16,17,18], or the parting direction will not cause any undercut because of convex polyhedron [19]. Consequently, it may be difficult to combine the approaches to determine parting directions and parting lines into one mold design system, because different approaches used different criteria and basic elements. In contrast, Dhaliwal et al. [11] used part feature bounds to generate candidate parting planes. Because their concern was accessibility to mold surfaces during machining, and they used sacrificial molds that do not need to be opened and reused, their methods are not as applicable to the present work as others.

3 Problem Formulation of Mold Design Method

An accurate problem formulation is significant because it will affect the research approach and the problem solving process to be used. Since Problem MD given in Section 1 is not computable, a more accurate problem formulation is needed for the mold design process. Obviously **demoldability** is one important consideration in Problem MD, which is also considered in most existing problem formulations. However, another important consideration, **face connectivity**, is often omitted in the formulations. For example, Chen, et al. [4] presented a representative approach for determining the parting direction of a part. Based on pockets and the Visibility Map of surfaces, they transformed Problem MD to a new problem formulation as:

Problem SPCA (spherical polygon covering by antipodes): Given a set of spherically convex polygons V_1, V_2, \dots, V_m , find a pair of antipodal points p and $-p$ that minimize the number of V_i containing either p or $-p$.

This problem formulation is widely referenced and also adopted in several other approaches [5,6,9]. However, because face connectivity is not considered in the formulation, several problems exist. First, it is assumed in the formulation that if several pockets share the same **PD**, they can be formed by a single mold piece without considering their actual positions. However, it is not always true. For example, in Fig. 2(a) **PD** satisfies the V-maps of P_1 and P_3 . However, since P_2 , which lies between P_1 and P_3 , cannot utilize the same **PD**, it is difficult to construct a single mold piece for P_1 and P_3 although their V-maps intersect. Second, the criterion of minimum non-covered pocket number is not always the same as the criterion of minimum mold piece number. For example, for a part as shown in Fig. 2(b) by using the criterion of minimum pocket number, **PD**₁ will be chosen as the parting direction. Therefore two additional cores are needed to form

pockets P_1 and P_5 . However, if **PD**₂ is chosen, only one core is needed to form P_2, P_3 and P_4 . Hence, the best solution to minimize the mold pieces (**PD**₂) is different from the solution to minimize non-covered pockets (**PD**₁).

In light of the problems of the existing formulations, our problem formulation for Problem MD is presented as follows.

Problem MCD: Mold Configuration Design. Given a solid polyhedral part in the Boundary Representation [20], it can be transformed as a graph $G(N,A,L,E)$ where N is the set of nodes, A is the set of arcs, L is the set of attributes (labels) for nodes, and E is the set of attributes for arcs, such that:

- for each face N_i of the part, there exists only one node in N ;
- for each common edge between faces N_i, N_j of the part, there exists a unique arc a_{ij} in A connecting them;
- for each face N_i , an attribute l_i (an integer number) is assigned to represent the region number;
- for each arc a_{ij} , an attribute e_k is assigned to represent the edge property. That is, if $l_i \neq l_j$, $e_k = 1$, and the related edge is defined as a *boundary edge*; otherwise $e_k = 0$, and the related edge is defined as an *internal edge*.

Among all the combinations of $G(N,A,L,E)$, find a configuration $G(N,A,L_i,E_j)$ ($i=1, \dots, K$) such that:

- for each face pair N_m and N_n , if $l_m = l_n = l$, a path can be found to link them with all nodes having the same l . In other words, all faces with l are connected (face connectivity);
- A direction **PD** exists for all faces with same value l_i so that the related mold piece M_i can be disassembled (demoldability);
- The total number of different l values (K) is minimized.

From the Problem MCD, we can see mold configuration design is actually a process of exploring and evaluating different face combinations to find a graph G . For further discussion, we first give a definition as:

Definition 1. A mold piece region R_i is the set of faces that have the same attribute l_i in the graph $G(N,A,L_i,E_j)$.

In order to generate mold pieces for a given part P , we divide the mold design process into three phases (Fig. 3). First from the boundary faces of P , we generate basic elements that serve as starting points for exploring different face combinations (Fig. 3(b)). Second we combine these basic elements into several mold piece regions, and find parting direction (PD) and parting lines (PLs) for each region (Fig. 3(c)). Finally, for a given mold base, we construct mold pieces for each mold piece region according to its PD and PLs (Fig. 3(d)). Corresponding to the three phases, the remainder of this paper has been organized in the following manner. In Section 4, we discuss suitable basic elements for Problem

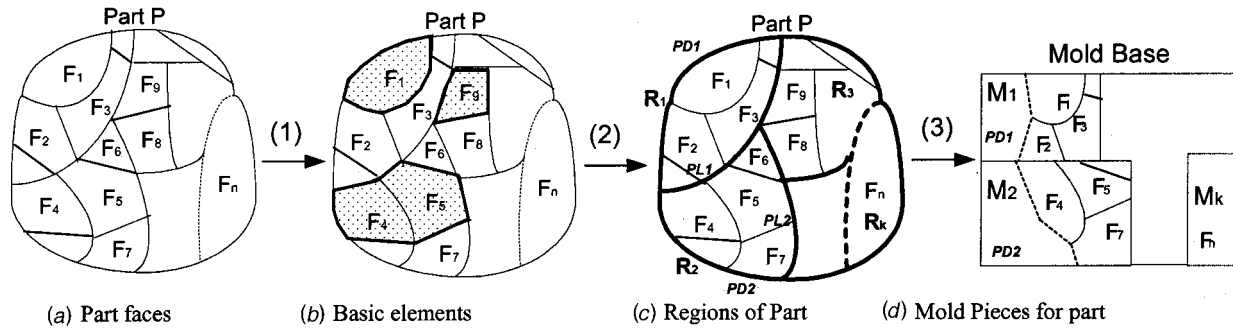


Fig. 3 Mold design process

MCD and how to generate them from a part P (Phase 1). In Section 5, we analyze how to combine the generated basic elements into mold piece regions (Phase 2). The final phase, construction of mold pieces (Phase 3), is discussed in [2]. The system implementation and results are discussed in Section 6. Finally, we give conclusions and future work in Section 7.

4 Basic Elements and Their Generation

For Problem MCD, conceptually if we try all different combinations of attribute R for a part represented by graph $G(N,A,R,E)$, we can always get the best mold design according to our requirements and criteria. However, it is quite obvious that this problem is strongly NP-hard. To solve this problem, suitable basic elements for Problem MCD should be considered first. In this section we present our basic elements based on analysis of mold piece demoldability. Another requirement of Problem MCD, face connectivity, will be considered in the combination process of basic elements, presented in Section 5.

4.1 Demoldability of Mold Pieces. In previous work [4,15], the demoldability of a face F is governed by the notion of *complete visibility*. That is, for every point p on F , if the ray from p to infinity in the direction d does not intersect the part, d is a good parting direction for face F . However, for multiple-piece mold design or form pin design, the requirement is less strict. That is, mold pieces can be translated in different orders and in more than one direction in the disassembly process. For example, shape P in Fig. 4 has an empty Visibility Map. However, it can be formed by two mold pieces, M_1 and M_2 . In the disassembly process, after removing M_1 , M_2 can be moved out in direction PD_2 , and then PD_1 . Therefore PD_2 is a feasible solution for face F even if F does not have complete visibility in direction PD_2 . Therefore instead of complete visibility, we will use positional relations of part faces to determine the demoldability of mold pieces. In this paper, we assume that a mold piece is demoldable if it can be translated away from the related mold piece region and

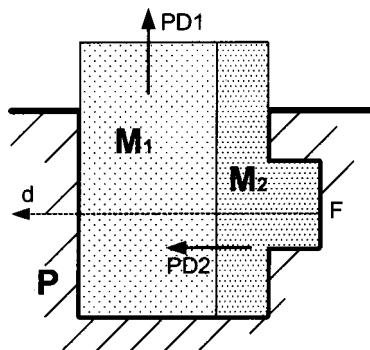


Fig. 4 Mold pieces for a pocket with empty V-map

its neighboring faces. Although the assumption will bring in the problem that further interference with other faces may happen after the mold piece is translated for some distance, a simulation module can handle the problem properly by face sweeping and interference testing.

For a mold piece M_i and its related mold piece region R_i , three lemmas are given as follows for the demoldability of M_i . The lemmas consider the faces of R_i , a neighboring face of R_i , and a neighboring mold piece M_j , respectively.

Lemma 1. Mold piece M_i can be removed from the faces of R_i by a translation in direction PD if and only if PD makes an angle of at most 90° with the outward normals of all faces F_{R_i} . Therefore, let $\eta(f) = (\eta_x, \eta_y, \eta_z)$ be the outward normal of a face f , and $PD = (d_x, d_y, d_z)$. Each face of R_i will induce a constraint:

$$\eta_x d_x + \eta_y d_y + \eta_z d_z \geq 0. \quad (1)$$

Suppose F_i and F_j are faces of regions R_i and R_j respectively. If F_i and F_j share an edge PE_i , F_j is called a *neighboring face* of R_i , and PE_i is called a *neighboring edge* of R_i . Suppose PE_i is related to two vectors CE_i and CE_j (Coedges in the *Boundary Representation*). The vectors have reverse directions that define the interior side of faces F_i and F_j .

Lemma 2. Mold piece M_i can be removed from a neighboring face F_j by a translation in direction PD without interference if and only if (1) PE_i is a convex edge or (2) PE_i is a concave edge and $PD \cdot \eta(F_j) \geq 0$.

Lemma 3. Two neighboring mold pieces M_i and M_j can be removed by translations in directions PD_i and PD_j individually without interference with each other at edge PE_i if and only if $CE_i \cdot (PD_i \times PD_j) \geq 0$, or $CE_j \cdot (PD_j \times PD_i) \geq 0$.

From the lemmas, it is obvious that the demoldability of M_i depends on the convex and concave properties of R_i and their neighboring faces. Further considering the properties of combining faces into R_i , we have two additional lemmas.

Lemma 4. Suppose all the edges of a face F_i are convex. Face F_i can be added to any neighboring region R_j without increasing the mold piece number or changing the demoldability of mold pieces, if a parting direction PD exists which makes an angle of at least 90° with the outward normals of all the faces of the region and F_i .

Lemma 5. Suppose all bounding edges E_1, E_2, \dots, E_k neighboring regions R_1 and R_2 are convex. These regions can be combined into one region without increasing the mold piece number or changing the demoldability of mold pieces, if a parting direction PD exists which makes an angle of at least 90° with the outward normals of all the faces F_1, F_2, \dots, F_n .

The proofs of the above lemmas are given in [21]. Based on the lemmas, two observations are given:

Observation 1. After combining faces into regions, two faces with a concave neighboring edge are more likely to be in the same region than those with a convex edge.

Observation 2. A face with only convex edges is more easily

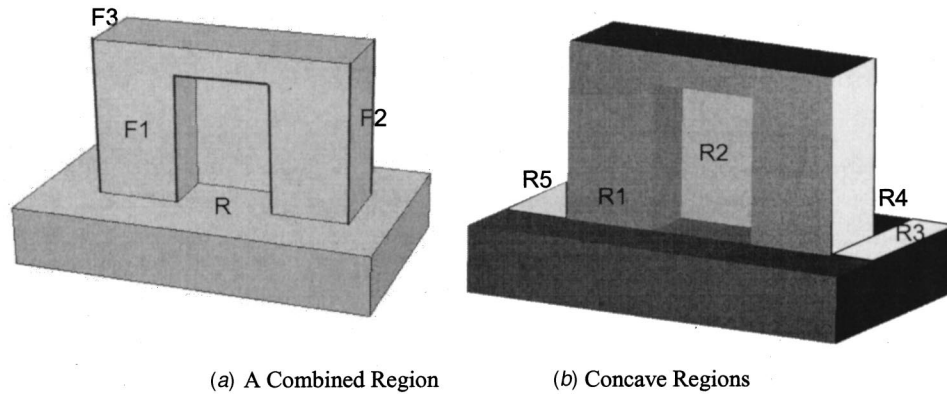


Fig. 5 Basic element comparison example

combined than a face with concave edges.

Based on the above analysis, the basic elements we propose to use for Problem MCD are concave region and convex face, which are defined as:

Definition 2. A *Concave Region*, $CVR(r)$, is a subgraph of $G(N,A,R,E)$ given in Problem MCD, such that: (1) for every node N_i that belongs to the subgraph, related attribute $r_i=r$; (2) for each arc A_{ij} related to N_i and N_j , if attribute $r_i=r_j=r$, the edge associated with the arc is concave; (3) for each arc A_{ij} related to N_i and N_j , if attribute $r_i=r$ and $r_j \neq r$, the edge associated with the arc is convex.

Definition 3. A face F is a *convex face* if all edges of F are convex edges.

A related definition, combined region, is also given as follows.

Definition 4. A *Combined Region*, $CR(r)$, is a subgraph of $G(N,A,R,E)$ given in Problem MCD, such that: (1) for every node N_i that belongs to the subgraph, related attribute $r_i=r$; (2) for each arc A_{ij} related to N_i and N_j , if attribute $r_i=r$ and $r_j \neq r$, the edge associated with the arc is a convex edge.

Therefore, a combined region can be a concave region, or a region with several convex faces, or a combination of some concave regions and convex faces.

4.2 Analysis of Basic Elements. Comparing the basic elements of our approach with those of others, we list the similarities and differences below.

(1) **Comparison with Pockets**

“Pockets” [4] and “concave regions” [5], are actually combined regions as defined in this paper. However by adding the concept of concave region, our approach is able to handle more general situations. From Definitions 3 and 4, it is clear that a concave region is different from a combined region in its internal edges. As an example, pocket R in Fig. 5(a) is a combined region since it has seven convex internal edges (marked in blue). By splitting R along the convex internal edges, R can be decomposed into five concave regions $R_1 \sim R_5$. Therefore mold design for the part can be generated by considering $R_1 \sim R_5$ instead of R , which has an empty Visibility Map. Since a combined region may consist of several concave regions, we believe a *concave region is a more basic element than a combined region*.

(2) **Comparison with features**

Dividing part faces into concave regions and convex faces is more general than classifying faces into features and undercuts (see Section 2.1.1). Based on their edge characteristics, features and undercuts can be divided further into three-edge, four-edge and more than four-edge. However, *all these features are actually special cases of concave region*. So if we just consider the relation of parting direction and faces in Problem MCD, concave region and convex face are much more general.

(3) **Comparison with faces**

Faces are one of the most basic elements of a 3D CAD model.

Although much flexibility can be obtained by using faces as basic elements, the number of part faces is much greater than that of regions, pockets, or features. For example, the pocket R shown in Fig. 5(a) has 10 faces. In comparison, the numbers of pocket and concave regions are 1 and 5, respectively.

In the next section, we present our approach to generate concave regions and convex faces from a given part.

4.3 Generating Approach of Basic Elements. Since a combined region can be decomposed into concave regions and convex faces, we use an approach with three steps to generate the basic elements of a part P .

(1) **Classify all edges of P into concave and convex;**

It is straightforward to determine if the dihedral angle of two neighboring faces is less than 180° .

(2) **Generation of combined regions and convex faces;**

This step is similar to the algorithm *FIND_CVR* for the generation of concave regions given in [4]. Basically for any two neighboring faces F_i and F_j , if the edge between them is concave, they should belong to the same region. Alternatively, the approach given in [4] could be used to construct a convex hull $CH(P)$ first, then generate pockets by the regularized difference between $CH(P)$ and P .

(3) **Generation of concave regions and convex faces for combined regions.**

The essential step to generate concave regions is to split the combined region based on convex internal edges. The algorithm *Split_Region* (SR) uses a face defined by convex internal edges (i.e., convex edges internal to a combined region) to split a given region. For each newly generated region, the function is executed recursively.

Algorithm: *Split_Region*

Input: A combined region CR .

Output: A set of concave regions S_{CVR} .

1. Test the parting direction (PD) of CR : if PD is not $(0, 0, 0)$, add CR to S_{CVR} and return.
2. Find all convex internal edges of CR .
3. If no convex internal edges exist, then add CR to S_{CVR} , and return.
4. Find a face F_{split} which has convex internal edges, and construct a split surface f_{split} from F_{split} .
5. Split each face F_i of CR into F_i^+ and F_i^- by f_{split} .
6. Generate a region CR^+ by adding all faces F_i^+ .
7. Generate a region CR^- by adding all faces F_i^- .
8. Add connecting faces in CR^+ and CR^- to new regions CR_i .
9. Call *Split_Region* for each CR_i .

Usually a combined region needs to be split only when it does not have a feasible parting direction, as shown in Figs. 5, 6 and 12(c).

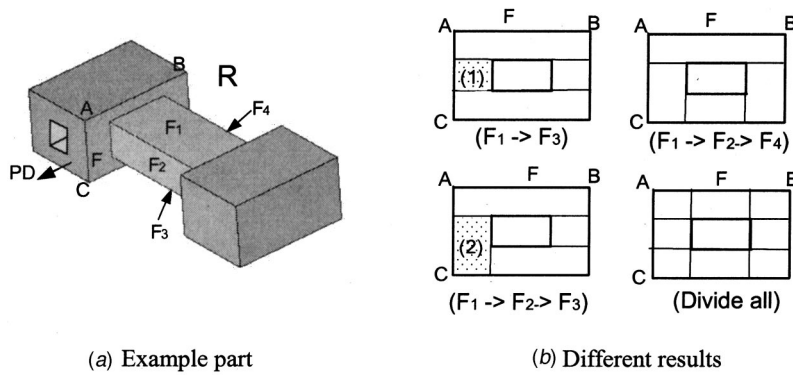


Fig. 6 Different splitting faces and orders

Since the main criterion in region combining (Section 5) is to test if a parting direction exists for two regions, we can use the parting direction of a region to judge if further splitting is necessary. The test in Step (1) subjects only the combined regions with no feasible parting direction to steps (2)–(9).

For a part with n faces, the algorithm can generate concave regions in $O(n^2)$ time. In the algorithm, face splitting in Step (5) is studied in [22]. Except for Step (4), the steps are straightforward. In Step (4), several faces may be candidate splitting faces. By using different splitting faces and different splitting orders, different regions will result, as will different mold designs. For example, for combined region R shown in Fig. 6(a) there are four convex internal edges. Consequently, any of faces (F_1 , F_2 , F_3 , or F_4) can be selected as the splitting face, resulting in 24 ($4!$) different combinations. The splitting results of F for some of the selections are shown in Fig. 6(b). Since the generated concave regions are different, the mold pieces generated for the part may be different. A straightforward way to get a unique dividing result is to use all the faces that have one or more convex internal edges to split the faces of all regions (including the newly generated faces). So face F in Fig. 6(a) will be divided into 9 small faces as shown in Fig. 6(b) (Divide all). However, by algorithm analysis, the complete splitting requires $\Theta(n \cdot b^{c-n})$ time, where n is the face number of part P , and b , c are some constants. For large n , the algorithm for complete splitting is obviously not acceptable. Therefore a good approach to select the splitting faces and their orders is necessary. The heuristic we use is: *find a face composing the largest number of convex internal edges to construct f_{split}* . Comparing our results to the results of complete splitting, we may miss some face combinations in the latter process.

In the next section, we discuss how to combine the basic elements into mold piece regions.

5 Combination of Basic Elements

After basic elements are generated, solving Problem MCD becomes a process to find a good combination of concave regions

and convex faces. In this section, we first discuss our evaluation approach for region combination. Then the combining process and algorithm are presented in Sections 5.2 and 5.3, respectively.

5.1 Evaluation of Parting Direction. The three criteria considered in Problem MCD are *face connectivity*, *a feasible parting direction*, and *minimum number of mold pieces*. In the combining process, face connectivity can be determined quickly and easily based on boundary representation data structures of faces [20]. The number of mold pieces can be determined based on recording the current region number. However, much more effort is needed to evaluate feasible parting directions quickly. Therefore, we focus on how to evaluate a feasible parting direction for the faces of a region.

Suppose a region R is composed of planar faces F_i ($1 \leq i \leq n$), with outward normals $\boldsymbol{\eta}(F_i) = (\eta_{ix}, \eta_{iy}, \eta_{iz})$. If direction $\boldsymbol{PD} = (d_x, d_y, d_z)$ is a feasible parting direction of R , from Lemma 1 we know:

$$\begin{aligned} \eta_{1x}d_x + \eta_{1y}d_y + \eta_{1z}d_z &\geq 0 \\ \eta_{2x}d_x + \eta_{2y}d_y + \eta_{2z}d_z &\geq 0 \\ &\dots \\ \eta_{nx}d_x + \eta_{ny}d_y + \eta_{nz}d_z &\geq 0 \end{aligned}$$

For the above problem, Woo [23] defined “Visibility Map (V-map)” to calculate the feasible directions that satisfy the above constraints. For a planar surface F , its V-map is a hemisphere centered on the unit outward normal. By calculating the intersections of all V-maps of region faces, allowable draw ranges of R can be computed. Several approaches and algorithms based on spherical polygons have been presented for different applications [23–26]. An example to illustrate the above process is shown in Fig. 7(a) which is presented in [9]. Based on the recognized undercut features, a cell (f_m) is calculated based on the intersection of their V-maps. Finally the central point of the cell f_m is selected as the optimal parting direction of the part.

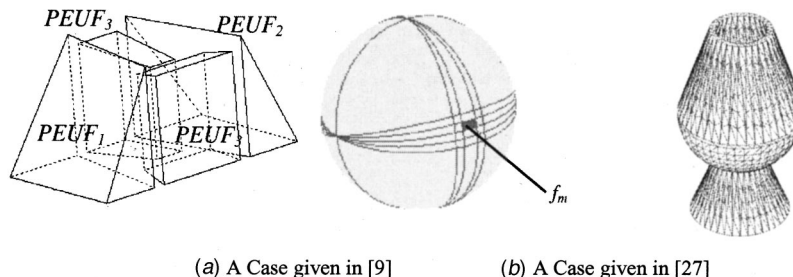


Fig. 7 Two examples of V-map.

The V-map and the calculation approaches based on spherical surfaces are widely referenced and used in most mold design papers. However, the algorithms and related data structures are rather complicated, and take considerable computational time. As a simple example, a case given in [27], which is shown in Fig. 7(b) would take 77 minutes on a Sun Ultra10 workstation.

In order to explore the combinations of basic elements better, we need an approach that can evaluate the parting directions of a region more quickly and robustly. For Problem MCD, we make two simplifications for determining a feasible parting direction, which are:

(1) **Calculate a feasible direction instead of the whole feasible range.**

For Problem MCD, we calculate the parting directions of regions for two reasons:

(i) to determine if a removable mold piece exists for a region to be combined. Since the main concern here is whether a parting direction exists for two regions or a region and a face, the actual draw-ranges are not important.

(ii) to find good parting directions to construct mold pieces. Even if the feasible draw ranges for each region are generated, a parting direction \mathbf{PD} still needs to be selected based on some criteria in constructing a mold piece.

Therefore only one \mathbf{PD} , instead of the whole range, needs to be calculated in combining two regions or a region and a face. Also if the criteria used in selecting \mathbf{PD} from the feasible range are followed in the calculation of a direction, the same \mathbf{PD} should be found for each mold piece after the region combination process. Therefore *the evaluation of \mathbf{PD} in the combining process can be formulated as an optimization problem*. Since only one direction is calculated, it is evident that the optimization approach is much faster than the use of spherical algorithms.

The ease of ejection is usually used as the criterion to choose a parting direction from a feasible range [4,9]. For a face F in a mold piece, its ease of ejection can be determined by the draft angle and the area in shear contact with the related part face during the mold-opening operation. In this paper we also use it as a criterion to calculate \mathbf{PD} . Suppose A_i is the area of a face F_i . The face normal, $\boldsymbol{\eta}_i$, forms an angle α with a direction \mathbf{PD} . In this paper we use the value $A_i \cdot dp_i = A_i \cdot (\boldsymbol{\eta}_i \cdot \mathbf{PD})$ to evaluate the ease of ejection for F_i in \mathbf{PD} . If we take $\boldsymbol{\eta}$ and \mathbf{PD} as unit vectors ($|\boldsymbol{\eta}| = 1, |\mathbf{PD}| = 1$), $dp_i = \cos(\alpha)$ and $A_i \cdot dp_i$ is actually the projected area of the face in \mathbf{PD} . Therefore, we can determine if a feasible parting direction exists for region R by solving an optimization problem for $\mathbf{PD}(d_x, d_y, d_z)$:

$$\text{Maximize: } f(d_x, d_y, d_z) = \sum_{i=1}^n A_i \cdot dp_i$$

$$\text{Subject to: } dp_i = \eta_{xi}d_x + \eta_{yi}d_y + \eta_{zi}d_z \geq 0$$

for each face F_i (plane constraints).

$$d_x^2 + d_y^2 + d_z^2 = 1 \quad (\text{sphere constraint})$$

(2) **Use a set of planar surfaces to approximate the unit sphere.**

To solve the optimization problem more quickly, we further simplify the problem by approximating the unit sphere with a set of linear faces. By setting an appropriate surface tolerance (chord height), a sphere can be approximated by triangles (we used 144 triangles). These triangles can be pre-generated and used for any regions. Suppose the equations of a triangle S_i are $s_{xi}x + s_{yi}y + s_{zi}z - s_i = 0$, and face normal (s_{xi}, s_{yi}, s_{zi}) is toward the inside. We can formulate a linear problem for evaluating a feasible parting direction.

Problem PDLP: Parting Direction Linear Problem.

$$\text{Maximize: } f(d_x, d_y, d_z) = \sum_{i=1}^n A_i \cdot dp_i$$

$$\text{Subject to: } dp_i = \eta_{xi}d_x + \eta_{yi}d_y + \eta_{zi}d_z \geq 0$$

for each face F_i (plane constraints).

$$s_{xi}d_x + s_{yi}d_y + s_{zi}d_z \geq s_j \quad \text{for each face } S_j (\text{sphere constraints}).$$

After solving Problem PDLP, we know: (i) if a solution $\mathbf{PD}(d_x, d_y, d_z)$ makes $f(d_x, d_y, d_z) > 0$, \mathbf{PD} is a feasible parting direction; (ii) if the solution $d_x = d_y = d_z = 0$, check unit vectors $\mathbf{PD}_1 = \boldsymbol{\eta}_1 \times \boldsymbol{\eta}_2$ and $\mathbf{PD}_2 = \boldsymbol{\eta}_2 \times \boldsymbol{\eta}_1$ by plane constraints, where $\boldsymbol{\eta}_1$ and $\boldsymbol{\eta}_2$ are the normals of two region faces. If the constraints are satisfied, \mathbf{PD}_1 or \mathbf{PD}_2 is also the solution of Problem PDLP; (iii) otherwise, there is no feasible parting direction.

Problem PDLP is a linear optimization problem. It is well studied in operations research [28]. For a linear programming problem in 3 dimensions, several algorithms can solve it on $O(n)$ time (n is the number of constraints) and linear storage [22,29]. Therefore, the running time to solve Problem PDLP is satisfactory even for a region with a large number of faces. A test example was used to verify this, consisting of a cylindrical boss on top of a planar face. By setting different surface deviations, the cylindrical surface of the boss can be approximated by different numbers of faces. The face number and corresponding running time for solving Problem PDLP are listed below. The testing time was based on the LINGO system (www.lindo.com) on a PC with a 700 MHz processor. The results obtained for the three cases are all (0.0, 0.0, 1.0).

(a) Surface deviation $-0.002''$; Region face number: 34; Time=0.16 second.

(b) Surface deviation $-0.0001''$; Region face number: 143; Time=0.17 second.

(c) Surface deviation $-0.00001''$; Region face number: 224; Time=0.20 second.

By formulating the evaluation of parting direction as a linear program, the solution process becomes much faster. Therefore we can explore more combinations of regions and faces in less time. Related to Problem PDLP, two discussions are given as follows.

Discussion 1: Quadric and Parametric Surfaces.

Although the algorithms in this paper are limited to planar surfaces, quadric and parametric surfaces can also be handled by approximating them with a series of planar surfaces. This is demonstrated in Section 6.

Discussion 2: Verification of Minimal Draft Angle.

For a part to be fabricated by the injection molding process, the surfaces that are parallel to the parting direction must be drafted at least an angle γ in order to ease the ejection of the part and reduce the possibility of damaging the part and molds [3]. For a complex model, the designer may lose track of which surfaces are drafted and which are not. Consequently a tool to automatically detect those non-drafted and under-drafted surfaces is necessary.

Suppose the minimum draft angle for a part is γ_{\min} . If the parting direction is given, the only task is to find faces with outward normals forming an angle between $90^\circ - \gamma_{\min}$ and $90^\circ + \gamma_{\min}$ with the parting direction [30]. However, the draft angle of a face depends entirely on the parting direction of the face. Therefore *the verification of draft angle is actually an integrated problem that should be considered in the mold configuration design process*.

In our approach, we can replace 0 in the plane constraints of problem PDLP with a variable $df = \sin(\gamma_{\min})$. Therefore we can use the constraints $dp_i = \eta_{xi}d_x + \eta_{yi}d_y + \eta_{zi}d_z \geq df$ to find all concave regions with non-drafted or under-drafted faces. Also if we follow the same criteria in the region combination process, we can be sure that all mold piece regions have only well-drafted faces in their parting directions. However the assignment of minimum draft angle γ_{\min} may influence the resulting number of mold pieces.

5.2 Region Combination Process.

After the basic elements of concave regions and convex faces are generated, we need to find a combination that satisfies the requirements of Problem MCD. In this paper, we assume all the concave regions of a part

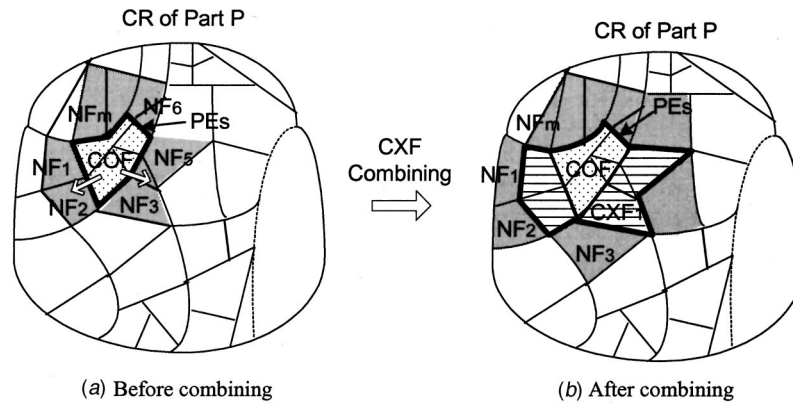


Fig. 8 Edges and faces of a region in a CXF combining step

have a feasible parting direction. Also the faces of a concave region will not be divided into different regions in the combining process. Two observations we make about the combining process are:

Observation 3. To get the minimum number of mold pieces, different regions and faces are to be combined into larger regions to the extent possible.

Observation 4. To maintain the connectivity of a region, only the region's neighboring faces need to be evaluated.

Since a convex face is more flexible than a concave/combined region, we can focus on concave/combined regions in the combining process. That is, we let regions grow individually by combining neighboring convex faces and regions, and updating their parting directions and parting lines. This region-growing process continues until no further change happens. The resulting regions are the mold piece regions of P . Some definitions of edges and faces are necessary.

Definition 5. For a concave/combined region R that is generated as a basic element, all its composing faces are called the *core faces* (COF) of the region R , and all the edges of the core faces are the edges of R . An illustrative example is shown in Fig. 8(a).

Definition 6. *Neighboring faces* (NF) of a region R are the faces that do not belong to the region but share at least an edge with R . Before the combining process, the parting direction of a region can be calculated by solving Problem PDLP. The parting lines of the region are composed by the edges such that one of its coedges has an owner face that belongs to R , and another coedge has an owner face that does not belong to R . The faces that do not belong to R can be recorded as the neighboring faces of the region.

In the region-growing process, there are two possible cases: (1) **A Region Combined with a Convex Face** (Fig. 8).

Suppose convex faces NF_1, \dots, NF_k are the neighboring faces of CR . To determine if CR can combine with NF_1 , the unit normal $(\eta_{xNF_1}, \eta_{yNF_1}, \eta_{zNF_1})$ of NF_1 is added as a plane constraint: $dp_{i+1} = \eta_{xNF_1}d_x + \eta_{yNF_1}d_y + \eta_{zNF_1}d_z \geq 0$ to the formulation of Problem PDLP for CR .

We can simplify the solution approach according to the properties of linear programming. For a linear problem with three variables, each linear constraint is actually a half plane in 3D space. So the feasible region is a polyhedron obtained by the intersection of all the half planes. According to [28], one of the corner points of the feasible region is always an optimal solution. Suppose PD (d_x, d_y, d_z) is the solution of Problem PDLP with i half plane constraints h_1, \dots, h_i . If a half plane h_{i+1} is added as a new constraint, we know from [22] that:

1. If PD satisfies the constraint h_{i+1} , the new optimal solution $PD' = PD$.
2. If PD does not satisfy constraint h_{i+1} , PD' must be one of the intersection points of h_{i+1} with $h_1 \sim h_i$, or the linear problem is infeasible.

For the latter case, an algorithm running in linear time is also given to find the new optimal solution. So the approach also runs in $O(n)$ time, but it should have a smaller coefficient of n compared to the approach of solving the problem again.

If PD' exists for a region and its neighboring face NF_i , they can be combined into one region.

Definition 7. The neighboring convex faces that are combined into a region R are called the *Convex Faces* (CXF) of R .

Accordingly, the faces of a region are composed of core faces and convex faces. One difference is that a core face will always

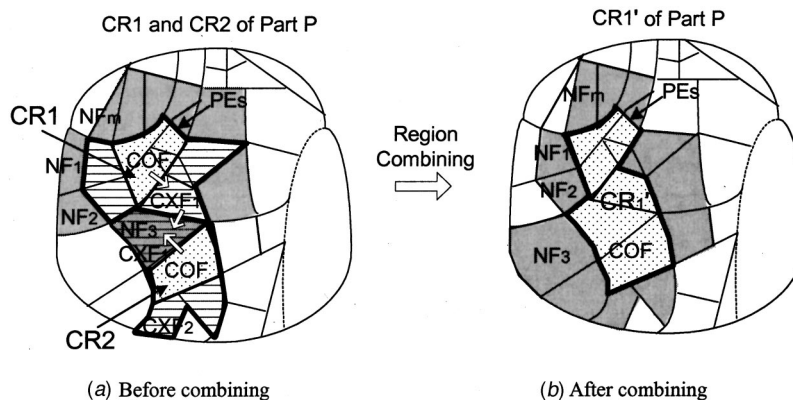
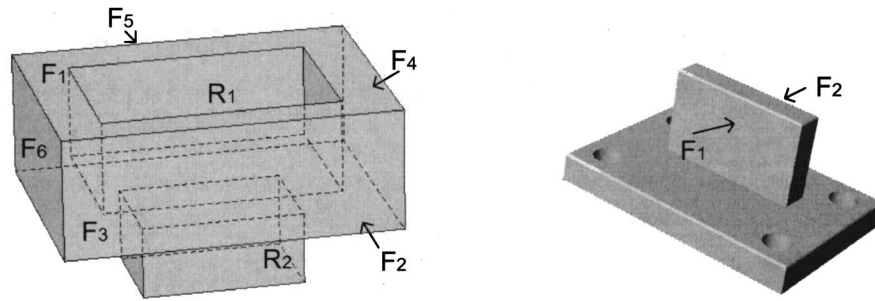


Fig. 9 Edges and faces of two regions in a combining step



(a) Different Combining Results of Regions and Faces (b) Mold Fabrication Knowledge

Fig. 10 Two example parts

belong to the region, and a convex face may leave the region to join another region as shown later. As shown in Figure 8.b, the region's parting lines and neighboring faces change after the convex faces are combined.

(2) A Region Combined with Another Region (Fig. 9).

Suppose a neighboring face (NF_3) of CR_1 is also a convex face (CXF_1) of another region CR_2 (Fig. 9(a)). So CR_1 and CR_2 are two neighboring regions that are candidates for combining into a larger region. Since the convex faces of CR_1 and CR_2 are more easily switched between regions than their core faces, convex faces are not considered in the test for region combining. That is, only COFs of CR_1 and CR_2 , and the convex faces to connect COFs, are formulated as planar constraints in Problem PDLF. If the linear problem has a solution, two regions are combined into one region CR'_1 (refer to Fig. 9(b)). And the core faces of CR'_1 are the faces considered in the planar constraints of Problem PDLF. All the convex faces of CR_1 and CR_2 are then tested as neighboring faces to determine if they can be combined into CR'_1 .

To get convex faces to connect COFs of two regions, the original face of a convex face also needs to be recorded in the combining process. That is, if a neighboring face F_1 shares an edge with a region face F_2 , we should record origin (F_1)= F_2 besides recording F_1 as the convex face of CR . So as shown in Fig. 9(a), if NF_3 of CR_1 and CXF_1 of CR_2 are the same face, the original face of NF_3 in CR_1 is added to planar constraints of CR_1 . We can recursively do the same thing for the face origin (NF_3) until its original face is a core face. Similarly, the original faces of CXF_1 can be added to CR_2 . The resulting core faces of CR'_1 are shown in Fig. 9(b).

Based on the combining process presented in this section, an algorithm for the region combining process is presented in the next section.

5.3 Combination Algorithm and Design Knowledge

Based on the data structures given in the last section, our region combination algorithm is listed below as Algorithm Combine_Region (NF and PE are sets of NF and PE respectively):

Algorithm: Combine_Region

Input: A set of combined regions S_{CVR} and convex faces of P .

Output: A set of combined regions S_{CVR} and convex faces of P .

- (1) change ← FALSE.
- (2) for a region CR_i in S_{CVR} ,
- (3) for each NF_k in NF ,
- (4) rn ← region_num(NF_k);
- (5) if $rn=0$, then $\|NF_k$ is not in any region
- (6) if combinable_convex_face(CR_i, NF_k) = TRUE, then
- (7) combine NF_k with CR_i , update PE and NF of CR_i , change ← TRUE.

- (8) else $\|NF_k$ belongs to region CR_{rn}
- (9) if combinable_region(CR_i, CR_{rn}) = TRUE, then
- (10) combine CR_{rn} with CR_i , update PE and NF of CR_i , change ← TRUE.
- (11) if change = TRUE, then Go to Step (1).
- (12) else return.

Based on our algorithm analysis, the computation time of the algorithm is $O(n^2)$, where n is the face number of P . A heuristic to reduce the running time is that if the tests in Steps (6) or (9) are negative in an iteration, they need not be repeated in later iterations. Face and region number arrays are added to each region to record neighboring convex faces and regions that are not combinable, respectively, and check these arrays first before solving Problem PDLF.

Steps (6) and (9) need to be discussed further. Conceptually, by using different combination orders and different rules in functions *combinable_convex_face* and *combinable_region*, different mold designs can be obtained. An illustrative example is shown in Fig. 10(a). Suppose a cavity (R_1) and an extrusion (R_2) are created in faces F_1 and F_2 respectively. Faces $F_3 \sim F_6$ are perpendicular to F_1 and F_2 . No draft angle is considered. Sixteen combination results can be obtained for the part by trying different combinations of convex faces ($F_3 \sim F_6$) and regions (R_1 and R_2). For example, $R_1 + F_1 + F_2 + F_3 + F_4 + F_5 + F_6$ versus R_2 ; and $R_1 + F_1$ versus $R_2 + F_3 + F_4 + F_5 + F_6$. Since the related mold designs have the same number of mold pieces and each mold piece can be disassembled properly, they all satisfy the requirements given in Problem MCD. Therefore they are equally good designs from the geometric perspective.

Instead of generating and recording all the alternatives, we consider more design knowledge in functions *combinable_convex_face* and *combinable_region*. The heuristic rules we consider are listed as follows with some explanations.

(1) Core/Cavity property of a region.

A region R can be classified as an internal or external region according to its parting lines. That is, if a loop of the parting lines is an internal loop of a neighboring face, R is an internal region; otherwise, it is an external region. For example, the parting lines of R_1 in Fig. 10(a) form an internal loop of F_1 . So R_1 is an internal region, while R_2 is an external region.

Generally internal regions are related to core mold pieces, and external regions are related to cavity mold pieces. To facilitate the ejection of a part from a core mold piece, convex faces with vertical normals to PD usually go with the cavity side instead of core side in the mold design. So the resulting mold design for the part given in Fig. 10(a) is R_1, F_1 in core side, and $R_2, F_2 \sim F_6$ in cavity side.

(2) Main parting direction of a region.

For a part with many regions R_i , we may get several parting directions PD_i . Among them, a pair of opposite directions is the

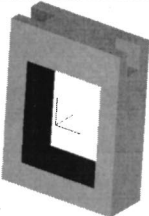
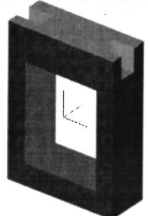
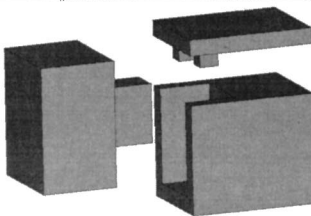
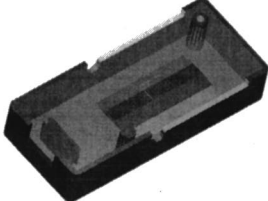
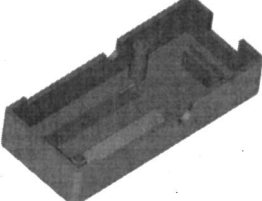
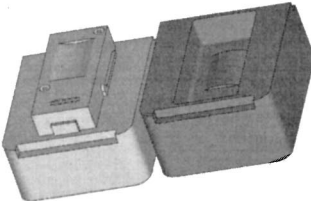
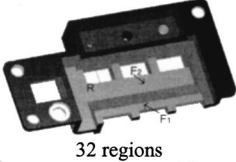
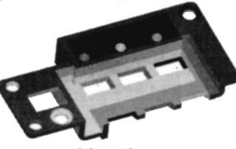
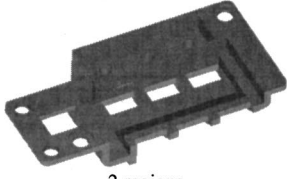
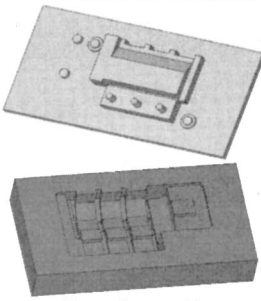
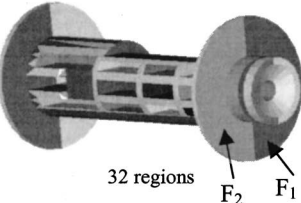
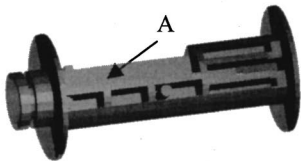
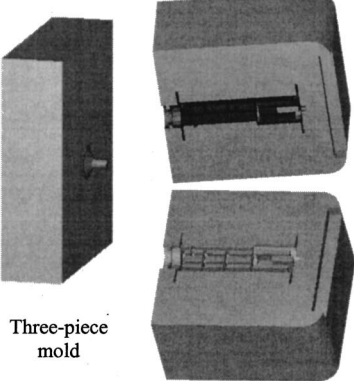
Part	Region Generation Results	Region Combining Results	Constructed Mold Pieces
(a)	 3 regions and 6 CXFs	 2 regions and 4 CXFs	 Three-piece mold
(b)	 28 regions	 2 regions	 Two-piece mold
(c)	 32 regions  38 regions	 2 regions	 Two-piece mold
(d)	 32 regions F_2 F_1	 3 regions A	 Three-piece mold

Fig. 11 Results of four tested parts

main parting direction (MPD) of the mold design. Our approach to determine MPD is to find a pair of directions with the maximum region volumes among all PD_k . That is, we find parting directions PD_k which are in the same or opposite directions. For each PD_k , a volume of the related region is calculated from its bounding box. The sum of all region volumes is assigned to the direction, and MPD is the direction with maximum volume. In general, it is preferred to combine a vertical face with a region in MPD than a region in a side direction.

(3) Combining order of a region.

The order of regions in S_{CVR} affects the combination result, as does the order of neighboring faces in NF for a region. For example, the part shown in Fig. 5(b) has five regions, $R_1 \sim R_5$. Depending on the order of R_3, R_5 related to R_1, R_2 and R_4 , we may get a combination result with R_1, R_2, R_3 , and R_5 as one mold piece region, and R_4 as the second region. We may also get a result with R_3, R_4 , and R_5 as a mold piece region, and R_1, R_2 as another region.

Therefore, before the combination process, we can reorder the

regions in S_{CVR} according to some heuristic rules. For example, regions can be ordered by their volumes. Hence, regions with larger volumes can be combined with neighboring faces and regions first. As another example, regions of S_{CVR} can be classified into two sets S_{CVR1} and S_{CVR2} according to some criterion, such as whether their PD is MPD. Then we can let regions in S_{CVR1} combine with neighboring faces and regions first.

(4) More design knowledge.

By considering more mold design knowledge, we can get mold designs that are more compatible with mold design practice. Our region-based approach is very flexible in enabling the addition of new design knowledge. To consider more design knowledge, related heuristic rules can be formulated and added to the *combinable_convex_face* and *combinable_region* functions in algorithm *Combine_Region* to generate different mold designs.

As shown in Fig. 10(b) a simple rib part is given as an illustrative example. Suppose mold pieces for the rib part are to be fabricated by SLA machines. Further assume surface finish require-

Table 1 Running Information of Test Parts.

Part	Part Information		Region Number		Running Time (sec.)	
	Face No.	Concave face No.	Region Generated	After Combining	Region Generation	Region Combining
(a)	18	12	3	2	0.50	0.50
(b)	330	207	28	2	2.46	7.19
(c)	606	489	38	2	8.17	10.53
(d)	544	336	32	3	3.05	35.68

ments of $2\ \mu\text{m}$ are specified for the two opposite faces F_1 and F_2 . According to our knowledge of the SLA process, such high surface finish can be achieved only by building the mold pieces such that F_1 and F_2 are all top surfaces [31]. Therefore we need to add a constraint that F_1 and F_2 cannot be combined into the same region in order to build them as the top surfaces. Consequently we can get a better design which satisfies the fabrication requirements, although this may lead to a three-piece design instead of a two-piece design.

Based on the generated regions, our approach to construct mold pieces is presented in [2].

6 System Implementation and Results

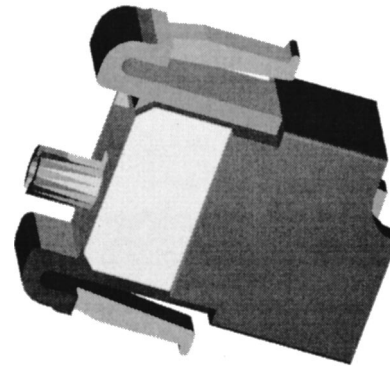
We implemented the proposed algorithms in an experimental system (Rapid Tooling Mold Design System) by using Microsoft Visual C++ 6.0 and ACIS6.2. ACIS is a 3D geometric modeler provided by Spatial Technology Inc. The linear programming solver we used for Problem PDLP is Lingo5.0 provided by Lindo Systems Inc. We have used our implementation to design molds to fabricate prototype parts of different complexities. Four test parts whose molds were generated by our system automatically are shown in Fig. 11. For each part, we display the graphical results of key steps. Different colors are used to represent regions and convex faces. We also list part information, region information, and execution time of each part in Table 1. All the test results are based on a PC with a 700 MHz Pentium III processor.

The speed of our system is satisfactory. The running time for the tested industrial parts is less than 40 seconds in generating mold piece regions. In case (c), since combined region R has no feasible parting direction, it is divided into eight concave regions by using Algorithm *Split_Region*. Faces F_1 and F_2 are chosen as the splitting surfaces because they have the largest number of convex internal edges (Section 4.3). In case (d), an additional step was added in the CAD file preparation process. That is, we used face A in Fig. 11(d) to split all the faces of the camera roller part. This is because the circular end faces of the flanges will not be split into two in the region generation and combination process. After splitting them (see F_1 and F_2), all faces in the part can be assigned successfully to regions. Recall that both the split region and region combination algorithms run in time $O(n^2)$.

7 Conclusions and Future Work

In this paper we presented a systematic approach based on regions for automated design of multi-piece molds. Face connectivity and demoldability of mold pieces are two main concerns in multi-piece mold design. For the mold design problem, we believe a three-phase solution process is suitable, which is shown in Figs. 3 and 11. For a given part, the convex and concave properties of its faces are the main factors in controlling demoldability of mold pieces, and therefore its mold design. Correspondingly, concave regions and convex faces are suitable basic elements for mold design. Algorithms were presented for generating regions and for combining them into mold pieces. The problem of finding a satisfactory parting direction of a region was formulated as a linear programming problem.

Several problems in mold design have exponential complexity. Heuristics based on mold design practice were introduced to yield

**Fig. 12 Regions of a fiber connector.**

polynomial-time algorithms and enable the design of useful molds. The algorithm for splitting regions is one example. Although complete splitting can generate a unique result of concave regions and convex faces, the running time of the approach is not feasible. So some heuristics are needed to select splitting surface and splitting criterion in region splitting. By generating small regions, the efficiency to explore face combinations is improved. The region combination algorithm also utilized heuristics, as discussed extensively in Section 5.3. The heuristics used were selected primarily for their utility in designing molds that were to be fabricated using stereolithography. Most likely, the heuristics should be modified for other methods of fabricating molds.

As a criterion to find feasible combinations of regions and faces, the parting direction of a region needs to be calculated. The process of calculating a V-Map and making selection from it can be simplified into a linear program. We demonstrate that solving a linear program provides a satisfactory solution much more quickly and easily than other methods in the literature. Additionally, detection of non-drafted surfaces is an important step in mold design. Since draft type is tightly related to a parting direction and parting lines, the detection should be executed in the determination process of parting direction.

The results shown in Section 6 validate the efficacy and robustness of our approaches. Parts b,c,d in Fig. 11, as well as others, were molded in stereolithography-produced molds that were designed by our system. We have not found any examples where mold designs produced by our system were infeasible.

The heuristics used in our methods have worked well in most parts studied. However, for some complex parts, such as the part in Fig. 12, we observed that some combinations of regions and faces are never explored due to the local effect of heuristics. These combinations of regions and faces may result in a better mold design from a global perspective. Currently we are exploring methods (e.g. building a region-combining tree) to generate and record all feasible mold design candidates, and then select one from them.

Acknowledgments

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. Also, we acknowledge the usage of Systems Realization Laboratory facilities.

References

- [1] Jacobs, P., 1996, *Stereolithography and other RP&M Technologies: from Rapid Prototyping to Rapid Tooling*, Dearborn, MI, Society of Manufacturing Engineers.
- [2] Chen, Y., and Rosen, D. W., 2001, "A Reverse Glue Approach to Automated Construction of Multi-piece Molds," Submitted to ASME JCISE (2001).
- [3] Rosato, D. V. and Rosato, D. V., 1995, *Injection Molding Handbook*, Chapman & Hall.

- [4] Chen, L., Chou, S., and Woo, T. C., 1993, "Parting Directions for Mould and Die Design," *Comput.-Aided Des.*, **25** (12), pp. 762–768.
- [5] Weinstein, M., and Manoochehri, S., 1996, "Geometric Influence of a Molded Part on the Draw Direction Range and Parting Line Locations," *Journal of Mechanical Design*, **118** (3), pp. 29–39.
- [6] Vijay, J. V. K., Shrinivasa, U., and Gurumoorthy, B., 1998, "Automatic Draw Direction Generation for Die Design," Proceedings ASME Computers in Engineering Conference, paper DETC98/CIE-5694, Atlanta.
- [7] Gu, Z., Zhou, Z., Gao, S., and Shi J., 1999, "Determination of Mold Parting Direction Based on Automatic Molding Feature Recognition," Proceedings ASME Computers in Engineering Conference, paper DETC99/CIE-9119, Las Vegas.
- [8] Fu, M. W., Fuh, J. Y. H., Nee, A. Y. C., 1999, "Undercut Feature Recognition in an Injection Mould Design System," *Comput.-Aided Des.*, **31**, pp. 777–790.
- [9] Yin, Z., Ding, H., et al. 2001, "Virtual Prototyping of Mold Design: Geometric Mouldability Analysis for Near-net-shape Manufactured Parts by Feature Recognition and Geometric Reasoning," *Comput.-Aided Des.*, **33**, pp. 137–154.
- [10] Ye, X. G., Fuh, J. Y. H., and Lee, K. S., 2001, "A Hybrid Method for Recognition of Undercut Features from Moulded Parts," *Comput.-Aided Des.*, **33**, pp. 1023–1034.
- [11] Dhaliwal, S., Gupta, S., Huang, J., and Kumar, M., 2001, "A Feature-Based Approach to Automated Design of Multi-Piece Sacrificial Molds," *ASME Journal of Computing and Information Science in Engineering*, **1**(3), pp. 225–234.
- [12] 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.
- [13] Hui, K., 1996, "Geometric Aspects of the Mouldability of Parts," *Comput.-Aided Des.*, **29**(3), pp. 197–208.
- [14] Urabe, K., and Wright, P. K., 1997, "Parting Direction and Parting planes for the CAD/CAM of Plastic Injection Molds," Proceeding ASME Design Automation Conference, paper DETC97/DAC-4336, Sacramento, CA.
- [15] Lu, H. Y., and Lee, W. B., 2000, "Detection of Interference Elements and Release Direction in Die-cast and Injection-moulded Components," *Journal of Engineering Manufacture*, **214**(B6), pp. 431–441.
- [16] Ravi, B., and Srinivasan, M. N., 1990, "Decision Criteria for Computer-Aided Parting Surface Design," *Comput.-Aided Des.*, **22**(1), pp. 11–18.
- [17] Tan, S. T., Yuen, M. F., Sze, W. S., and Kwong, K. W., 1988, "A Method for Generation of Parting Surface for Injection Moulds," *Conference on Computer Aided Production Engineering*, Edinburgh, UK.
- [18] Wong, T., Tan, S. T., and Sze, W. S., 1996, "Parting Line Formation by Slicing a Trimmed Surface Model," Proceedings ASME Design Automation Conference, paper DETC96/DAC-1058, Irvine, CA.
- [19] Majhi, J., Gupta, P., Janardan, R., 1999, "Computing a Flattest, Undercut-free Parting Line for a Convex Polyhedron, with Application to Mold Design," *Computational Geometry Theory and Applications*, **13**, pp. 229–252.
- [20] Mäntylä, M., 1988, *An Introduction to Solid Modeling*, Computer Science Press.
- [21] Chen, Y., "Computer-Aided Design for Rapid Tooling: Methods for Mold Design and Design-for-Manufacture," Ph.D. Dissertation, Georgia Institute of Technology, 2001.
- [22] de Berg, M., van Kreveld, M., Overmars, M., and Schwarzkopf O., 1997, *Computational Geometry-Algorithms and Applications*, Springer.
- [23] Woo, T. C., 1994, "Visibility Maps and Spherical Algorithms," *Comput.-Aided Des.*, **26**(1), pp. 6–16.
- [24] Chen, L. L., and Woo, T. C., 1992, "Computational Geometry on the Sphere with Applications to Automated Machining," *ASME J. Mech. Des.*, **114**, pp. 288–295.
- [25] Gupta, P., Janardan, R., Majhi, J., and Woo, T., 1996, "Efficient Geometric Algorithms for Workpiece Orientation in 4- and 5-axis NC Machining," *Comput.-Aided Des.*, **28**(8), pp. 577–587.
- [26] Kweon, S., and Medeiros, D. J., 1998, "Part Orientations for CMM Inspection Using Dimensioned Visibility Maps," *Comput.-Aided Des.*, **30**(9), pp. 741–749.
- [27] Dhaliwal, S., Gupta, S., and Huang J., 2000, "Computing Exact Global Accessibility Cones for Polyhedral Objects," Proceedings ASME Design for Manufacturing Conference, paper DETC00/DFM-14034, Baltimore.
- [28] Reklaitis, G. V., Ravindran, A., Ragsdell, K. M., 1983, *Engineering Optimization: Methods and Applications*, New York, John Wiley and Sons.
- [29] Megiddo, N., 1984, "Linear Programming in Linear Time When the Dimension is Fixed," *ACM*, **31**, pp. 114–127.
- [30] Serrar, M., 1995, "Applying Moldability Evaluation to Solid Models," *Mechanical Engineering*. Troy, NY, Rensselaer Polytechnic Institute.
- [31] 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.