

# A Geometric Algorithm For Automated Design Of Multi-Stage Molds For Manufacturing Multi-Material Objects

Malay Kumar

Department of Mechanical Engineering  
University of Maryland  
College Park, MD20742

Email: malayk@eng.umd.edu

Satyandra K. Gupta

Department of Mechanical Engineering  
University of Maryland  
College Park, MD20742

Email: skgupta@eng.umd.edu

## ABSTRACT

This paper describes a geometric algorithm for automated design of multi-stage molds for manufacturing multi-material objects. In multi-stage molding process, the desired multi-material object is produced by carrying out multiple molding operations in a sequence, adding one material in the target object in each mold-stage. We model multi-material objects as an assembly of single-material components. Each mold-stage can only add one type of material. Therefore, we need a sequence of mold-stages such that (1) each mold-stage only adds one single-material component either fully or partially, and (2) the molding sequence completely produces the desired object. In order to find a feasible mold-stage sequence, our algorithm decomposes the multi-material object into a number of homogeneous components to find a feasible sequence of homogeneous components that can be added in sequence to produce the desired multi-material object. Our algorithm starts with the final object assembly and considers removing one component either completely or partially from the object one-at-a-time such that it results in the previous state of the object assembly. If the component can be removed from the target object leaving the previous state of the object assembly a connected solid then we consider such decomposition a valid step in the stage sequence. This step is recursively repeated on new states of the object assembly, until the object assembly reaches a state where it only consists of one component. When an object-decomposition has been found that leads to a feasible stage sequence, the gross mold for each stage is computed and decomposed into two or more pieces to facilitate the molding operation. We expect that our algorithm will provide the necessary foundations for automating the design of multi-stage molds and therefore will help in significantly reducing the mold design lead-time for multi-stage molds.

## Keywords

Mold Design, Geometric Reasoning, and Spatial Partitioning

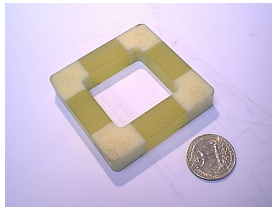
## 1 INTRODUCTION

Multi-stage molding refers to a molding process in which multiple materials are added in a sequence to produce multi-material

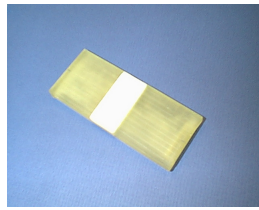
objects. Multi-stage molding technique can be used to create multi-material objects by pouring multiple materials in different mold-stages. A partially assembled mold is used to pour one material. After completing one mold-stage, the mold assembly is modified by adding/removing mold-pieces and a different material is poured to produce a different portion of the object. By using multiple stages, a complex multi-material object can be manufactured. Figure 1 shows examples of four multi-material objects produced using multi-stage molding technique in our laboratory. The ability to create multi-material objects allows designers to select different materials for different portions of the object, thus helping to improve material-functional compatibility for the overall object.

If a given object is manufacturable using multi-stage molds, then we need to generate a sequence of feasible mold-stages for manufacturing the multi-material object. This requires spatial partitioning of gross mold shape of the object. The spatial partitioning of the gross mold is driven by a number of factors, which include material variation in the object, manufacturability of the mold-piece, feasibility of assembly and disassembly sequence of mold-stages. Following are the steps involved in automated mold design for multi-stage molding:

1. *Initialization*: As a first step the solid model of the gross mold for final mold-stage is created by subtracting the solid model of the multi-material object from the solid model of a large rectangular block. The rectangular block should completely enclose the object.
2. *Material variation based object decomposition*: If the target object is made of multiple materials, then the object is produced by carrying out multiple mold-stages in sequence adding one homogenous component in the target object in each mold-stage. The multi-material object is thus decomposed recursively into a number of homogeneous components. This gives a feasible sequence of adding homogeneous components to the object one at a time to produce the desired multi-material object.
3. *Mold-Stage generation*: The solid model of the gross mold is decomposed to get the solid models of the mold-pieces used in different mold-stages. Each mold-stage is defined by computing the mold-pieces that are to be removed from the previous stage and the mold-pieces that are to be added in the current stage.
4. *Manufacturing driven spatial decomposition*: Mold-pieces of each stage should be manufacturable. The mold-pieces used in different mold-stages may not be manufacturable by any of the available manufacturing process and may need to be decomposed further.
5. *Addition of assembly features*: Once the mold decomposition is completed, assembly features are added to the mold-pieces



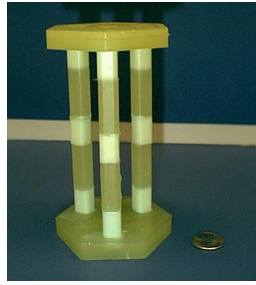
(a) Four-Bar Linkage



(b) Snap-fit Cap



(c) Gripper



(d) Torsion Table

Figure 1. Example of Multi-Material Objects

to ensure that each mold-piece is kinematically constrained in the mold assembly.

6. *Postprocessing*: After the mold assembly for a mold-stage has been designed, the user has to select a mold-piece for creating the sprue. A sprue is a passage through which the liquid material is poured into the mold-stage.

This paper describes algorithms that cover Steps 1 to 3. The other three steps are currently performed manually. Designing sequence of mold-stages for manufacturing of geometrically complex multi-material objects requires sophisticated geometric reasoning which are difficult to perform manually. Therefore, we believe that automation of the mold design function will significantly reduce the mold design time and the cost of deploying multi-stage molds.

## 2 RELATED WORK

Shape deposition manufacturing (SDM) is a layered manufacturing process in which parts are fabricated by deposition of layers in a certain build orientation and machining each layer to give them three dimensional shape [16]. SDM is being used to fabricate multi-material parts [1, 18]. In SDM, by varying the materials used in the deposition process, spatial variation in the material properties of the part can be achieved. The desired object and sacrificial support material is partitioned into layers in order to eliminate accessibility problems in machining. Faces that form undercut with respect to the build orientation on the part are the primary factors that cause accessibility problems and require decomposition. Ramaswami et al. presented an approach for detecting undercut faces and partitioning objects for SDM for a given build orientation [19]. Rajagopalan et al. [18] and Cham et al. [3] provide additional details on manufacturing of heterogeneous objects. Mold-pieces of multi-stage molds can be manufactured in multiple different orientations. Thus, undercut faces do not necessarily imply accessibility problems. Therefore this approach cannot be directly used in the multi-stage mold decomposition.

There have been several efforts in the area of object slicing for SFF. Most work in this area focuses on object decomposition using 2.5D layers [13, 14, 23]. Therefore, it is not directly applicable to mold decomposition problem. Horváth et al. have presented an algorithm for morphological segmentation for layered manufacturing of large parts [8]. A four-stage algorithm is used to reduce a complex object to fabrication elements. Due to significant differences in the nature of constraints between this process and mold manufacturing, this approach is not applicable to mold decomposition process.

In the area of mold decomposition, most work is concentrated on parting line selection for traditional molds [4, 5, 9, 20, 24]. The presence of undercuts plays a significant role in determining the parting line. Some researchers have also discussed cases in which cores are incorporated into the molds to handle undercuts [21, 22]. Krishnan et al. have developed a mold design algorithm that generates multi-piece molds for homogeneous parts that are created by stacking 2.5D solids along the Z direction [12]. The parting surface directions are restricted to be along the principal axes. Parting surface design approach provides valuable insight into the mold design process and identifies several factors that contribute to quality of molded part. However, in order to design multi-stage molds, we need to develop a truly three-dimensional partitioning algorithm.

Several processes are being developed for manufacturing of heterogeneous objects. Multi-material selective laser sintering (MMSLS) has been developed to fabricate functionally gradient material (FGM) objects [2, 11]. Another layered manufacturing process, which is capable of producing heterogeneous objects is 3D Printing [10, 25]. Locally composition control (LCC) components are fabricated by printing different materials in different locations, each through its own ink-jet nozzles. Qui et al. have developed a multi-material layered manufacturing system for the design and fabrication of multiphase electromechanical parts [17]. The developed system generates multi-material tool path and does virtual simulations for defect quantification of multi-material layered manufacturing. Multi-material injection molding techniques are being used to fabricate multi-material plastic objects. This is achieved in multi-shot as well as single shot injection molding machines. One such company Fickenscher [7] offers two and three material molded parts. This technique is a variation of standard injection molding.

## 3 PROBLEM FORMULATION

### 3.1 Problem Statement

Multi-stage molds can be used for manufacturing multi-material objects. Multi-material objects can be modeled as an assembly of homogenous components. Each component  $c_i$  of the object assembly is represented as a solid model and has material attribute  $a_m$  associated with it. The material attribute  $a_m$  defines the material type of each the homogenous component. Figure 2 shows an example of a multi-material object modeled as an assembly of eight homogenous components. The object in Figure 2 is consists of five different materials. The final multi-material object  $O_f$  is produced using a sequence of mold-stages  $T$ . Each stage  $t_i$  in  $T$  is defined by:

1. The mold assembly  $M_i$  used in the current stage  $t_i$
2. The set of mold-pieces  $M_i''$  that need to be removed from previous stage to create  $M_i$ .
3. The set of mold-pieces  $M_i'$  that need to be added to the previous stage to create  $M_i$ .

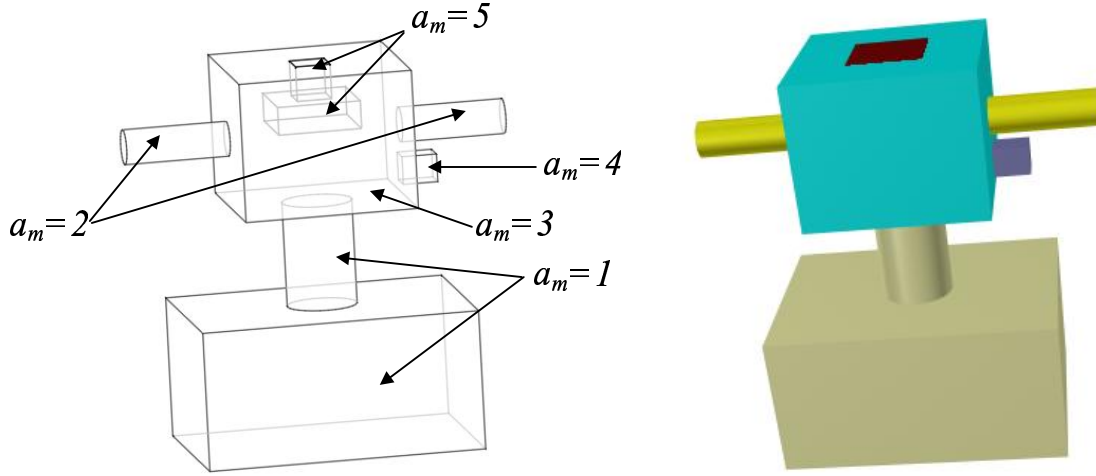


Figure 2. A multi-material object modeled as an assembly of homogeneous components

4. The resulting object assembly  $O_i$  after the current mold-stage.
5. The solid model of the component  $c_i$  to be added to the already fabricated portion of the target object  $O_{i-1}$  in the current stage.

In mold-stage  $t_i$ , two types of transformations are achieved. First, the mold assembly  $M_i$  is created from the mold assembly  $M_{i-1}$  by removing unnecessary mold-pieces  $M_i''$  and adding the required new mold-pieces  $M_i'$ . Second, the object component  $c_i$  is added to the already fabricated object  $O_{i-1}$  by pouring the material  $a_m(c_i)$  into the mold assembly  $M_i$ .

We have developed an algorithm that performs the following tasks:

1. It performs manufacturability analysis to make sure that the final multi-material object  $O_f$  is manufacturable using multi-stage molding process.
2. If  $O_f$  is manufacturable, then it generates a feasible mold-stage sequence  $T$ . Each stage  $t_i$  in  $T$  should meet the following feasibility conditions:
  - None of the mold-pieces or object components should intersect with each other.
  - Union of mold-pieces and object components should be a connected solid.
  - There should exist a feasible sequence to assemble/disassemble mold-pieces to/from the previous mold assembly that need to be added/removed during the transformations associated with the stage  $t_i$ .

Our algorithm is currently restricted to multi-material objects in which contacts between homogeneous components in the object assembly is only through planar faces.

### 3.2 Overview of The Approach

Each mold-stage only handles one type of material. Therefore, we need to decompose the object so that a feasible mold-stage sequence can be generated. Since the components made of the same material can be fabricated in a single stage, we first combine all components which are of the same material and when

combined together results in a connected solid. This step gives the final object assembly  $O_f$ .

We initialize the current object assembly  $O_i$  with final object assembly  $O_f$  and consider removing one component  $c_i$  from the current object assembly  $O_i$  such that it results in the previous object assembly  $O_{i-1}$ . We try to find a component  $c_i$  that can be removed from the object assembly  $O_i$  leaving the remaining assembly  $O_{i-1}$  completely connected (i.e., union of all components in the remaining assembly is a connected solid). If a valid component  $c_i$  exists, then we find a plane or a set of planes that can be used to separate  $c_i$  from  $O_i$ . Section 4 describes how to find (1) a valid component for the object decomposition, and (2) a plane or a set of planes for separating a valid component from the object assembly.

If the object assembly reaches to a state where it cannot be fabricated in a single stage and none of the homogeneous component can be removed from the current state of the object assembly, then we need to perform component decomposition to facilitate object fabrication. Section 5 describes how to partition a homogenous object component. Once a component has been decomposed into multiple components, we find a plane or a set of planes for separating a component from the object assembly.

Using the same plane or the set of planes that are used to separate a component from the object assembly, we partition the mold assembly needed to add component  $c_i$  in the previous state of the object assembly  $O_{i-1}$ . This partitioning defines the mold-pieces that need to be removed from the previous mold-stage and mold-pieces that need to be added to the current stage to achieve mold-stage transformation. Section 6 describes how to perform this step.

From  $O_{i-1}$  we remove another component to get further object decomposition. This process will be repeated recursively until the object assembly reaches a state where it can be fabricated in a single stage. During each step of object decomposition, a mold-stage is created to add the decomposed component to the previous state of the object assembly. Therefore, a feasible sequence of decomposing the object concurrently produces a feasible mold-stage sequence.

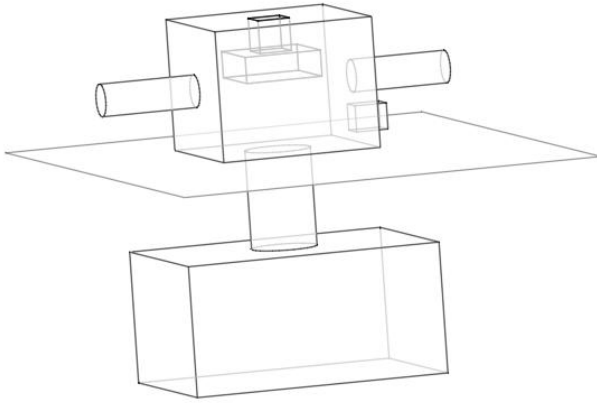


Figure 3. A valid single parting plane to separate a component out from rest of the assembly

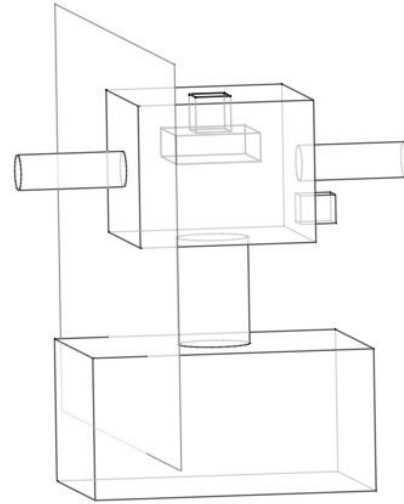


Figure 4. An invalid single parting plane

#### 4 MATERIAL VARIATION BASED PARTITIONING OF OBJECT

In order to generate a feasible mold-stage sequence, we need to find and separate a homogeneous component  $c_i$  from the current object assembly  $O_i$ . During the molding process, this component will be added to the previous state of the object assembly  $O_{i-1}$  to create  $O_i$ . We also need to find the set  $S_i$  of partitioning planes, which can be used to separate the component  $c_i$  from assembly  $O_i$ . The following steps are performed to find a component  $c_i$  that can be removed from  $O_i$  and the set  $S_i$  of associated partitioning planes:

1. Find a set of components  $R_i$  that can be removed from the assembly  $O_i$ . A component  $c_i$  can be removed from  $O_i$  if  $c_i$  is disassemblable from the assembly  $O_i$  and removing  $c_i$  from the assembly  $O_i$  leaves the remaining object assembly a connected solid. We used algorithm by Woo and Dutta [6] for disassemblability analysis.
2. Compute the decomposition priority of the components in  $R_i$ . High priority is given to components that share faces with the component that has been removed from the object in one of the previous decomposition steps.
3. Examine components in  $R_i$  in the order of their descending priorities. Find the first component  $c_i$  in  $R_i$  such that a single partitioning plane can separate  $c_i$  from the rest of the object assembly. If  $c_i$  has been found, then return  $c_i$ , return the partitioning plane, and stop. (Section 4.1 describes an algorithm for finding a single partitioning plane.)
4. Examine components in  $R_i$  in the order of their descending priorities. Find the first component  $c_i$  in  $R_i$  such that a set of partitioning planes can separate  $c_i$  from the rest of the object assembly. If  $c_i$  has been found, then return  $c_i$ , return the set of partitioning planes, and stop. (Section 4.2 describes an algorithm for finding a set of partitioning planes.)
5. If none of the component in  $R_i$  can be separated using either a single partitioning plane or a set of partitioning planes, then return failure. In such a case one or more components will need to be decomposed using the algorithm described in Section 5.

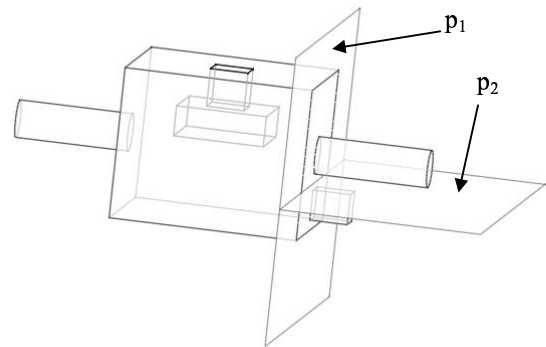
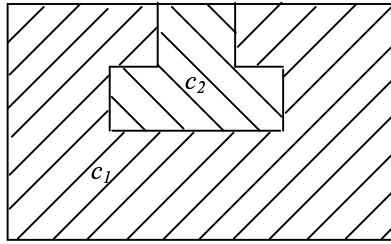


Figure 5. A set of parting planes consisting of two planes used to separate a component from assembly

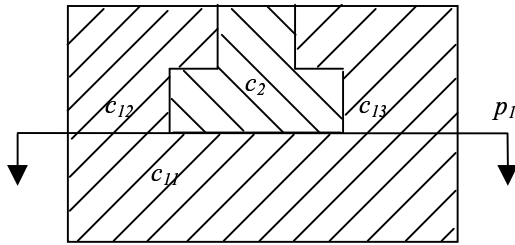
##### 4.1 Finding a Single Partitioning Plane to Separate a Component From the Assembly

To check if a component  $c_i$  can be separated from the object assembly  $O_i$  using a single partitioning plane we use the following steps:

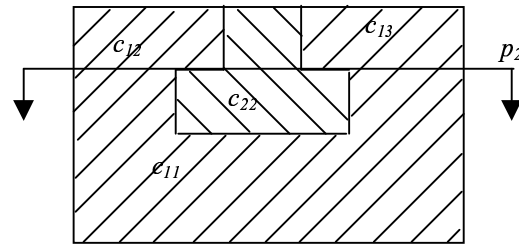
1. Make an object  $O'$  which is union of all the components except  $c_i$ .
2.  $c_i$  can be separated from the object  $O_i$  using a single partitioning plane if it shares only one face with  $O'$ . Find the set of all the faces of  $c_i$  shared with  $O'$ .
3. If only one face of  $c_i$  is shared with  $O'$  then plane  $p$  represented by shared face is a candidate-partitioning plane.
4. The plane  $p$  will be a valid partitioning plane if its intersection with the interior of  $O'$  and the interior of  $c_i$  is null. If  $p$  is a valid partitioning plane, then  $c_i$  can be separated from rest of the assembly using partitioning plane  $p$ . Figure 3



(a) Neither Component  $c_1$  nor component  $c_2$  can be removed



(b) Decomposition Alternative 1



(c) Decomposition Alternative 2

Figure 6. Examples of component decomposition to facilitate mold stage generation

shows an example where a single partitioning plane is used to separate a component from the object assembly. Figure 4 shows an example of an invalid partitioning plane. An invalid partitioning plane may become a valid partitioning plane at a later stage of object decomposition.

## 4.2 Finding Multiple Partitioning Planes to Separate a Component From the Assembly

If partitioning the assembly using a single partitioning plane cannot separate a component from the rest of the assembly, then we need to try more than one partitioning plane to separate the component  $c_i$  from the assembly. Here we have to find a set  $S_i$  of partitioning planes such that the  $c_i$  lies in the region formed by intersecting half spaces associated with planes in  $S_i$  and that region does not contain any portion of rest of the object assembly. The following steps determines set  $S_i$  of partitioning planes to perform partitioning:

1. Create a set  $L$  of all the components from which  $c_i$  needs to be separated. Initialize this set with all the components of  $O_i$  except  $c_i$ .
2. Create a set of candidate partitioning planes  $P$ , which consist of all the faces of  $c_i$ , which are shared with the components in  $L$ .
3. Select a candidate valid plane  $p$  from  $P$ .  $p$  is a valid partitioning plane if its intersection with interior of  $c_i$  is null. If no valid partitioning can be found, then stop. No solution is possible in this case.
4. Take the intersection of  $p$  with each component in  $L$ . If  $p$  intersects with a component  $X$  of  $L$  then decompose  $X$  along the plane  $p$  into two or more components.
5. Remove  $X$  from  $L$  and add the decomposed components of  $X$  to  $L$ .

6. Find those components in  $L$ , which are not on the same side of  $p$  as  $c_i$  and remove them from  $L$ . Add  $p$  to set  $S_i$ .
7. Repeat Steps 3 through 6 until either  $L$  or  $P$  is empty.
8. If  $L$  is not empty and  $P$  is empty, then we need to find additional planes that will separate component  $c_i$  from all the remaining component in  $L$  and add them to  $S_i$ . The following steps describe how to find such additional planes:
  - a. Construct a set  $C_h$  that contains the convex hull of all the components in  $L$ . For every pair of convex hulls  $(c_h, c_h')$  where  $c_h$  is the convex hull of component  $c_i$ , and  $c_h' \in C_h$ , compute  $I = c_h \cap c_h'$ . If  $I \neq \emptyset$  for any such pair then component  $c_i$  cannot be separated from rest of the components using this algorithm.
  - b. Find a partitioning plane such that a pair of convex hulls  $(c_h, c_h')$  lie on opposite sides of this plane and add that plane into the set  $S_i$ . Megiddo [15] describes a linear time algorithm for computing such a plane using the approach of linear programming.
  - c. Repeat Steps (a) and (b) until  $L$  is empty.

Figure 5 shows an example when a set of partitioning planes can be used to separate a component from the assembly.

The above algorithm does not necessarily produce all the planes needed for partitioning. This is a limitation of the current algorithm. This algorithm cannot handle those cases when the candidate partitioning planes in Step 3 intersect with the component  $c_i$  itself. But these cases can be handled by decomposing the component along the material interfaces and doing so generates mold-pieces of simpler shape. Section 5 describes how to decompose a component.

## 5 PARTITIONING OF HOMOGENEOUS COMPONENTS OF THE OBJECT

This step is needed if more than one component is left in the object assembly  $O_i$  and none of the components can be separated

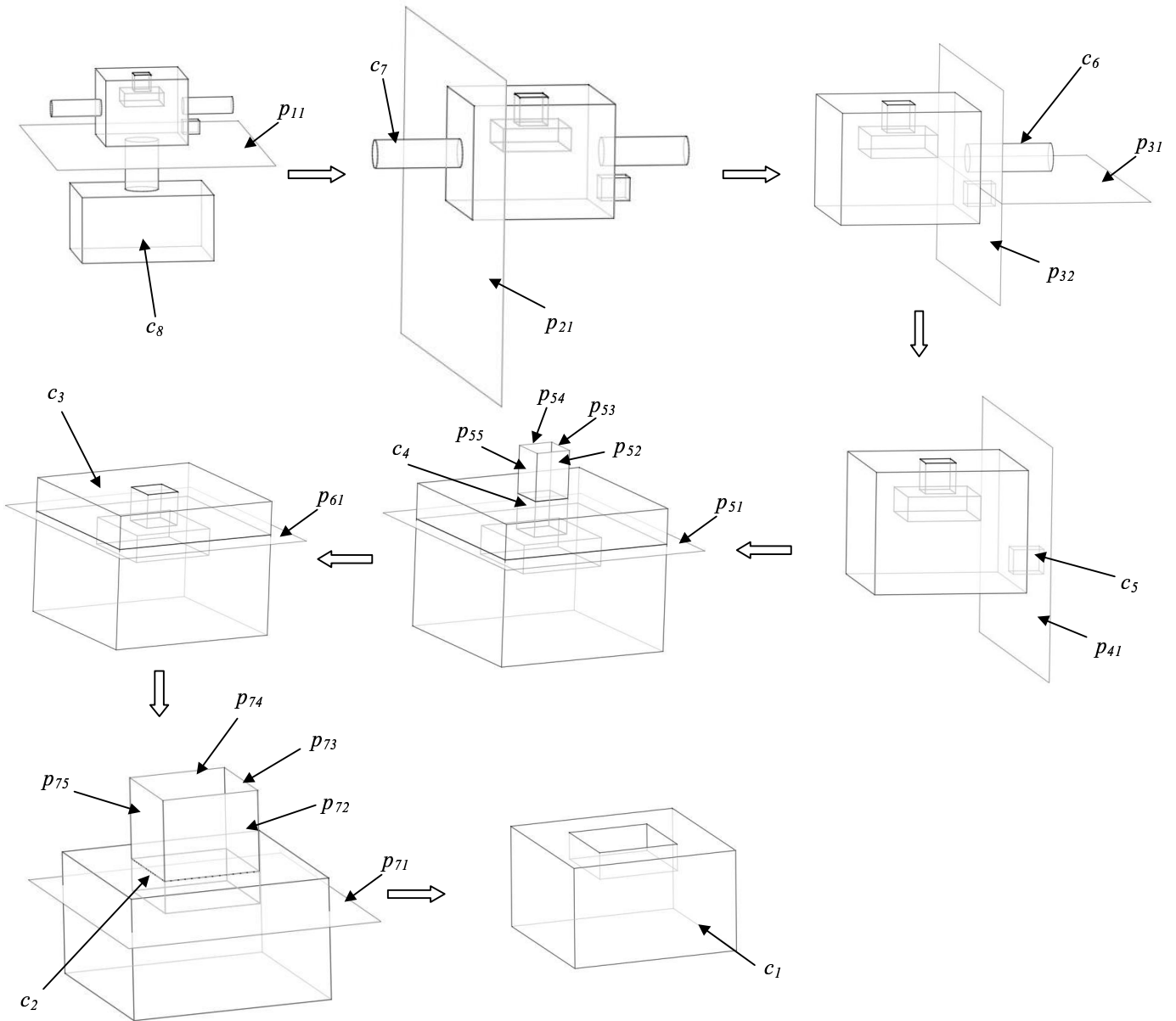


Figure 7. A Feasible Object Partitioning Sequence

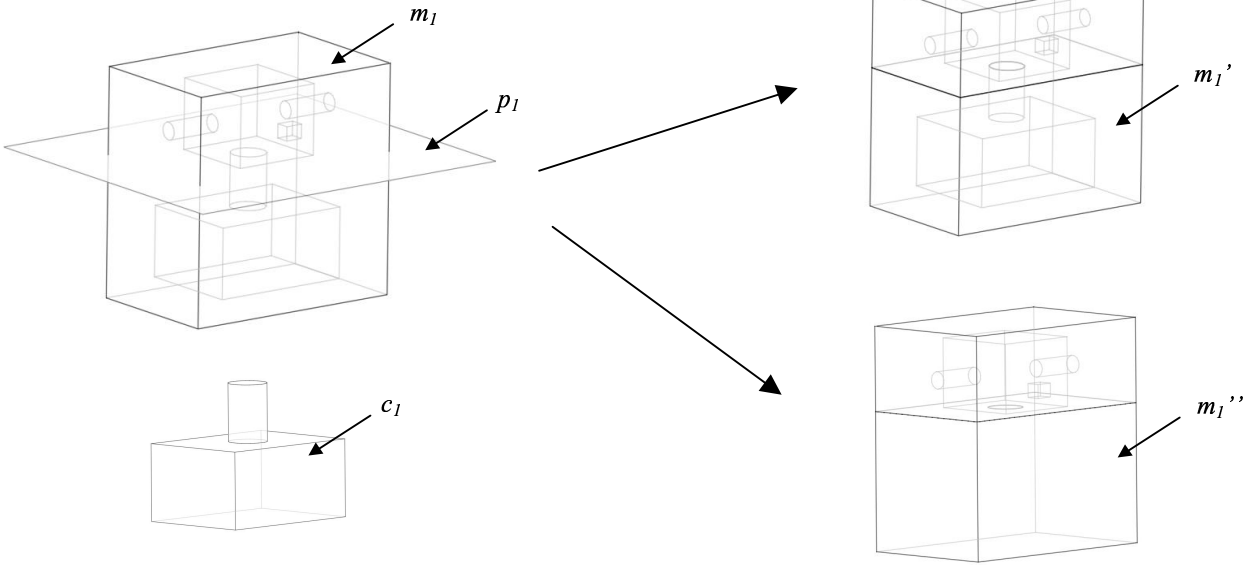
from the assembly using a set of partitioning planes. This situation arises in following two cases:

- If the removable component set  $R_i$  computed in Section 4 is empty, i.e. none of the component is disassemblable from the object assembly  $O_i$
- Algorithms described in Sections 4.1 and 4.2 fail to find a component that can be separated using a set of partitioning planes.

In this case components in the object assembly  $O_i$  need to be decomposed along one or more planes to make the removal of a component feasible. Candidate partitioning planes are along the faces, which are shared by two or more components. This would result in the components to be decomposed along material interfaces. Components can be decomposed along one or more candidate partitioning planes. The number of components and

thus number of mold-stages as well as shapes of mold-pieces depend on choice of partitioning planes. The following steps describe our algorithm for decomposition of components to facilitates further stage generation:

1. Create a set of candidate partitioning plane  $P$  from the faces of remaining components in the component set which are shared by more than two components.
2. Select a plane  $p$  in  $P$  and decompose the components in the object assembly  $O_i$  along  $p$ . After decomposition update the component set and try finding components that can be removed from the object assembly  $O_i$  using the algorithm described in Section 4. If a valid object partitioning is enabled then stop.



(a) Gross Mold Shape, parting plane and component added in the final stage of the Final Stage

(b) Gross Mold Shape of the last Stage and the previous stage after that include partition for interstage mold component change

Figure 8. Generation of last mold-stage for the object shown in Figure 2.

3. If the decomposition does not enable object partitioning then combine the components back. Remove  $p$  from  $P$ . If  $P$  is empty, then stop. Otherwise, go back to Step 2.

Figure 6(a) shows a case where component decomposition is needed because none of the components of  $c_1$  and  $c_2$  can be removed from the current state of the object assembly  $\{c_1, c_2\}$ . Figures 6(b) and 6(c), show two possible decompositions of assembly  $\{c_1, c_2\}$  to facilitate mold-stage generation. As shown in Figure 6(b), only one decomposition along  $p_2$  is sufficient to generate all mold-stages necessary to produce object assembly  $\{c_1, c_2\}$ . In the second decomposition, as shown in figure 6(c), the components are decomposed along  $p_1$  that facilitates removal of component  $c_{11}$ . In this case the assembly  $\{c_{12}, c_{13}, c_2\}$  will need to be decomposed further in subsequent steps to generate all mold-stages necessary to produce the original object assembly  $\{c_1, c_2\}$ . The object assembly can also be decomposed along other candidate partitioning planes from the set of candidate partitioning planes. Thus, there are many possible solutions of the component decomposition, which depends on the order of partitioning planes in set of candidate partitioning planes. This can be treated as an optimization problem over the number of stages or cost of mold-piece fabrication. However, currently we only try to find a feasible solution.

## 6 GENERATION OF MOLD-STAGE

The algorithm described in Section 3 generates an ordered set of homogenous components that corresponds to mold-stage-sequence. It also finds the sets of partitioning planes used in each step of the object decomposition. Figure 7 shows a feasible sequence of object partitioning for the example part. Once we find set  $S_i$  of partitioning planes, that can be used to separate the

component  $c_i$  from the object assembly  $O_i$ , we perform partitioning of mold-pieces using the partitioning planes in  $S_i$  to define mold-stages. For each mold-stage we define the mold-pieces that are to be removed from the previous stages and the mold-pieces that are to be added to the current stage. We start from the final mold-stage and find the mold-pieces involved in mold-stage transformation recursively. Since the previous mold-stage should not contain the cavity for the component  $c_i$  added in the current stage the previous mold-stage should contain a mold-piece  $m_i$  of the same geometry as the component  $c_i$ . The removal of component  $c_i$  requires partitioning of mold-pieces. Mold-pieces are partitioned along the planes in set  $S_i$  so that mold-piece  $m_i$  can be assembled and disassembled. The mold-piece  $m_i$  can be combined with some other mold-piece of the previous stage to produce mold-piece of simpler shape, if doing so does not pose any assembly/disassembly problem in mold-stage transformation. The following algorithm is used to identify mold pieces in every mold-stage:

1. Initialize a set  $M$  of mold-pieces with the gross mold of final multi-material object  $O_f$ . The solid model of the gross mold for final mold-stage is created by subtracting the solid model of the multi-material object from the solid model of a large rectangular block. The rectangular block should completely enclose the object.
2. For each  $c_i$ :
  - Partition mold-pieces in  $M$  along planes in  $S_i$  and find the set  $M'$  of mold-pieces which are on the same side as  $c_i$ .
  - Add a mold-piece  $m_i$  of the same geometry as  $c$  to  $M$ .

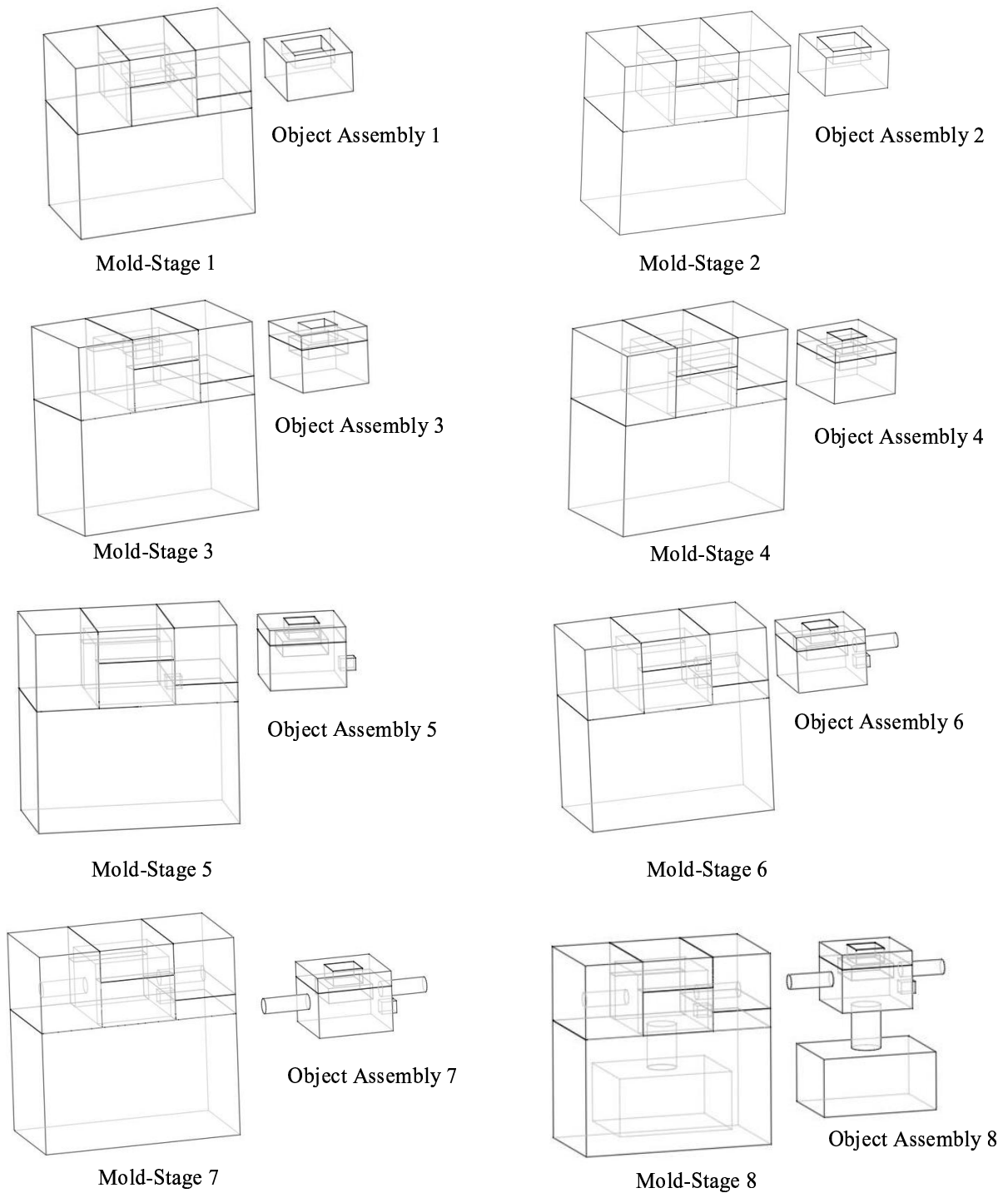


Figure 9. Mold stages for the example part in Figure 2 and the state of object assembly after each stage

- For every pair of mold pieces  $(m_s, m_b)$  where  $m_s \in M$  and  $m_b \in M'$ , subtract  $m_b$  from  $m_s$ .
- Add all mold pieces of  $M'$  to  $M$ .
- Unite the mold-piece  $m_i$  with some other mold-piece  $m_i'$  in  $M$  which share one or more face(s) with it to avoid assembly features on the shared face and to produce a mold-piece with a simpler shape. This step produces a mold-piece  $m_i''$ . If  $m_i$  share face(s) with more than one components, then  $m_i$  is united with a mold-piece  $m_i'$  that shares the maximum number of faces with  $m_i$  and the minimum number of faces with other components.
- Define a mold-stage  $t_i$  as  $m_i''$  being the component that has to be removed from the previous stage and  $m_i'$  being the mold-piece that has to be added to produce the



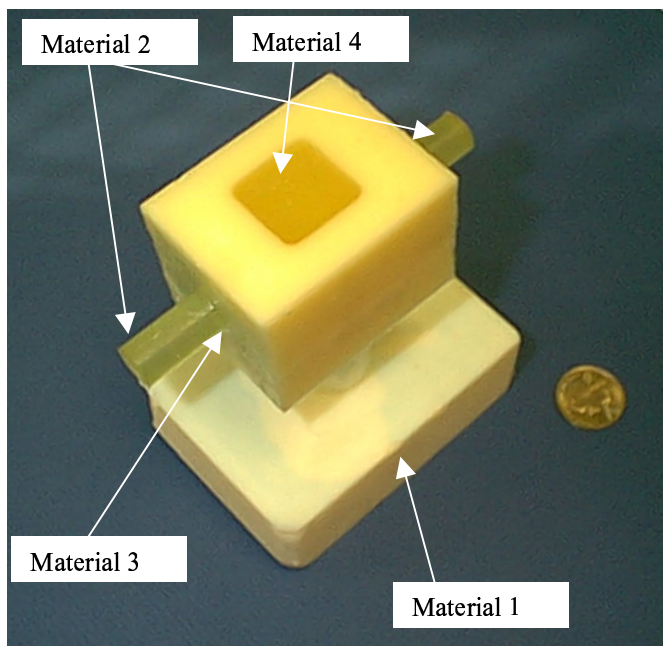


Figure 10. Fabricated Multi-material part using the stage sequence shown in Figure 9

mold-stage  $t_i$ . Add the mold-stage  $t_i$  at the beginning of the mold-stage list  $T$ .

3. Define the first mold-stage as all the mold-pieces in  $M$  being the components that has to be added and add it at the beginning of the mold-stage sequence  $T$ .

The above steps produce all the mold-stages required to produce the multi-material part. Figure 8 shows the generation of the last mold-stage for the example part. Figure 8(a) shows the mold-piece  $m_j$  of  $M$ , component  $c_j$  and the partitioning plane  $p_j$  used to create the last mold-stage. The mold-pieces are partitioned along the plane  $p_j$  as shown in figure 8(b), which produces mold-piece  $m_j'$  of set  $M'$ . It then finds the mold-piece  $m_j''$  that is to be removed from the previous stage and  $m_j'$  that has to be added to define the last mold-stage. Figure 9 shows the complete sequence of mold-stages generated using our algorithm for the example part and the state of object assembly after each mold-stage.

## 7 CONCLUSIONS

### 7.1 Summary

This paper describes a geometric algorithm for generating mold-stages for multi-stage molding. These algorithms will be useful for a wide variety of molding processes that use sacrificial or permanent molds. Novel features of this algorithm are:

- It finds multiple partitioning planes to perform partitioning on the mold-pieces.
- It performs object and mold decomposition that are needed to ensure the assembly and disassembly of mold-pieces during mold-stage assembly.
- It generates the complete molding sequence of the multi-stage molds. It specifies mold-pieces that are to be added and mold-pieces that are to be removed from the previous stage to produce the mold assembly at each stage.

A prototype system has been developed based on the algorithms described in this paper for designing multi-stage molds. The system is developed in C++ and uses ACIS Geometric kernel. The system has a Java3D based visualization interface to display the mold-stages designed by the system. Figure 10 shows the picture of the part that was produced using the mold-stage sequence shown in Figure 9.

### 7.2 Anticipated Benefits

For multi-material polymer objects that are manufacturable using multi-stage molds, the processing cost associated with the multi-piece molds is significantly lower compared to the other processes for making these objects (e.g., 3D printing and selective laser sintering) when the batch size is fifty or more. Therefore, development of automation technology that can reduce the lead-time for mold manufacturing will help in making molding of multi-material ceramic/polymer objects commercially viable for small to medium batch production. The ability to manufacture geometrically complex multi-material objects economically will significantly expand the design space and will allow development of new products in many areas.

### 7.3 Current Limitations

Our current algorithm has the following limitations:

- The contact between homogenous components is assumed to be through planer faces. This limits the types of material interfaces in the multi-material object that can be currently handled by our algorithm. We are planning to extend the algorithm to handle commonly used curved interfaces by defining curved partitioning along the mating faces.
- The mold-pieces generated by our algorithm may not have the optimum shape. We plan to develop algorithms to optimize the number and shape of mold-pieces by combining mold-pieces of different stages that don't pose any problem in mold-stage transformations.
- We do not consider the feasibility of addition of assembly features and sprues in the current work. Future plans include determining feasibility of designing assembly features on the mold-pieces and back-tracking in case a feasible object decomposition sequence does not give a feasible mold sequence.

## 8 ACKNOWLEDGEMENTS

This research has been supported by the NSF grant DMI9896255, ONR grant N000140010416, and Minta Martin Research Foundation. Opinions expressed in this paper are those of authors and do not necessarily reflect opinion of the sponsors.

## 9 REFERENCES

- [1] S. A. Bailey, J. G. Cham, M. R. Cutkosky, and R. J. Full. Biomimetic Robotic Mechanisms via Shape Deposition Manufacturing. *9th International Symposium of Robotics Research*, Snowbird, Utah, October 9-12, 1999, p. 321-327.
- [2] J. Beaman, D. Bourell, B. Jackson, L. Jepson, D. McAdams, J. Perez, and K. Wood. Multi-Material Selective Laser Sintering: Empirical Studies and Hardware Development. *Proceedings of the 2000 NSF Design and Manufacturing Grantees Conference*, Jan. 2000.
- [3] J.G. Cham, B.L. Pruitt, M.R. Cutkosky, M. Binnard, L. Weiss, and G. Neplotnik. Layered Manufacturing with Embedded Components: Process Planning Issues. *Proceedings of the 1999*

- ASME DETC/DFM Conference, Las Vegas, NV, Sept 12-15, 1999.
- [4] L. Chen and T.C. Woo. Parting directions for mold and die design. *Computer Aided Design*, 25(12):762-768, 1993.
- [5] L. Chen, S. Chou, and T.C. Woo. Separating and intersecting spherical polygons: computing machinability on three, four, and five axis numerically controlled machines. *ACM Transaction on Graphics*, 12(4):305-326, 1993.
- [6] D. Dutta, and T. C. Woo. Automatic Disassembly and total ordering in three dimensions. *ASME Journal of Engineering for Industry*, 1991, 113(1), 207-213.
- [7] Fickenscher America (<http://www.fickenscher.com/>).
- [8] I. Horváth, J. J. Broek, Z. Rusák, G. Kuczogi, and J. S. M. Vergeest. Morphological segmentation of objects for thick-layered manufacturing. In *Proceedings of the ASME Design for Manufacturing Conference*, Las Vegas, Nevada, September, 1999.
- [9] K. C. Hui and S.T. Tan, "Mould Design with Sweep Operations – a Heuristic Search Approach". *Computer Aided Design*, Vol. 24(2), 1992.
- [10] T. P. Jackson, E. M. Sachs, and M. J. Cima. Modeling and Designing Components with Locally Controlled Composition. *Proceedings of the Solid Freeform Fabrication Symposium*, August, 1998.
- [11] L. Jepson, J. Perez, J. Beaman, D. Bourell, and K. Wood. Development of Multi-Material Selective Laser Sintering Process. *Proceedings of the 1999 NSF Design and Manufacturing Grantees Conference*, 1999.
- [12] S. Krishnan and E. B. Magrab, "A New Approach to Mold Design Using Manufacturable Entities", In *Proceedings of the Design for Manufacturability Symposium, ASME Winter Annual Meeting, Dallas, TX*, November 1997.
- [13] V. Kumar, and D. Dutta. An Approach to Modeling Multi-Material Objects. *Proceeding of 4th ACM Symposium on Solid Modeling*, Atlanta, GA, May 1997.
- [14] A. Marsan, and D. Dutta. A survey of process planning techniques for layered manufacturing. In *Proceedings of the ASME Design Technical Conference*, Sacramento, CA, Sept. 1997.
- [15] N. Megiddo, "Linear-Time Algorithm for Linear Programming in  $R^3$  and Related Problems", *Society for Industrial and Applied Mathematics*, Vol. 12, No. 4, November 1983.
- [16] R. Merz, F.B. Prinz, K. Ramaswami, M. Terk, and L. Weiss. Shape Deposition Manufacturing. *Proceedings of the 1994 Solid Freeform Fabrication Symposium*, 1994.
- [17] D. Qui, N. Langrana, S. Danforth, M. Jafari, and A. Safari, Virtual Simulation for Multi-material LM Process. *Proceedings of the Solid Freeform Fabrication Symposium*, August 1998.
- [18] S. Rajagopalan, R. Goldman, K.H. Shin, V. Kumar, M.R. Cutkosky, and D. Dutta. Representation for the Design, Processing and Freeform Fabrication of Heterogeneous Objects. accepted for publication to *Materials and Design*, 2000.
- [19] K. Ramaswami, Y. Uamaguchi, and F. F. Prinz. Spatial partitioning of solids for solid freeform fabrication. In *Proceedings of the Fourth ACM Symposium on Solid Modeling and Applications*, Atlanta, GA, May 1997.
- [20] B. Ravi and M. N. Srinivasan, Decision Criteria for Computer-Aided Parting Surface Design. *Computer Aided Design*, Vol. 22(1), 1990.
- [21] D. W. Rosen. Towards Automated Design of Molds and Dies. In *Proceedings ASME Computers in Engineering Conference*, Minneapolis, September 1994.
- [22] K. H. Shin and K. Lee. Design of side cores of injection moulds from automatic detection of interference faces. *Journal of Design and Manufacturing*, vol. 3, pp. 225--236, 1993.
- [23] Y. S. Suh and M. J. Wozny. Adaptive slicing of solid freeform fabrication processes. In *proceedings of Solid Freeform Fabrication Symposium*, Austin, Texas, August 1994.
- [24] M. Weinstein and S. Manoochchri, "Optimum Parting Line Design of Molded and Cast Parts for Manufacturability", *Journal of Manufacturing Systems*, Vol. 16(1), 1997.
- [25] H. Wu, E. M. Sachs, N. M. Patrikalakis, D. Brancazio, J. Serdy, T. R. Jackson, W. Cho, H. Liu, M. Cima, and R. Resnick. Distributed Design and Fabrication of Parts with Local Composition Control. *Proceedings of the 2000 NSF Design and Manufacturing Grantees Conference*. Jan 2000.