M. Kumar and S.K. Gupta. Automated design of multi-stage molds for manufacturing multi-material objects. *ASME Journal of Mechanical Design*, 124(3):399-407, 2002.

Automated Design Of Multi-Stage Molds For Manufacturing Multi-Material Objects

Malay Kumar and *Satyandra K. Gupta*[1]

Department of Mechanical Engineering
University of Maryland
College Park, MD20742

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 add only 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 a sequence to produce the desired multi-material object. Our algorithm starts with the final object assembly and considers removing components either completely or partially from the object one-at-a-time such that it results in the previous state of the object assembly. If a 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 a step towards automating the design of multi-stage molds and therefore will help in reducing the mold design lead-time for multi-stage molds.

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, called a mold stage 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-function 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 for the object. The spatial partitioning of the gross mold is driven by a number of factors, which include material variation in the object, manufacturibilty 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.

---

[1] Corresponding Author

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 a 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. *Manufacturability-driven spatial decomposition*: 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 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 is 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 [1]. SDM is being used to fabricate multi-material parts [2]. In SDM, spatial variation in the material properties of the part can be achieved by varying the materials used in the deposition process. 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 [3]. Rajagopalan et al. [4] and Cham et al. [5] provide additional details on manufacturing of heterogeneous objects using SDM technique. 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 decomposition for SFF. Most work in this area focuses on object slicing using 2.5D layers [6-8]. 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 [9]. 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 [10-14]. 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 [15, 16]. 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 [17]. 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 [18, 19]. Another layered manufacturing process, which is capable of producing heterogeneous objects is 3D Printing [20, 21]. 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 [22]. 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.

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 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 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 can be defined by (M_i, M_i^d, M_i^a, c_i) where

1. M_i : Starting mold-stage assembly in the *i*th mold-stage.
2. M_i^d : Set of mold-pieces that need to be removed in the *i*th mold-stage.
3. M_i^a : Set of mold-pieces that need to be added in the *i*th mold-stage.
4. c_i : Component that is to be added to the already fabricated portion of the target object O_{i-1} in the *i*th stage.

In each mold-stage t_i, two types of transformations are achieved. First, the mold assembly M_{i+1} is created from the mold assembly M_i by removing unnecessary mold-pieces M_i^d and adding the required new mold-pieces M_i^a. This is called mold stage transformation. 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+1}. This is called object transformation.

We have developed an algorithm that performs the following tasks:

1. It performs preliminary 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. The complete multi-material object is fabricated by adding the decomposed parts of the object in the generated sequence. 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 the 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 generated 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.

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 produce O_i. We also need to find the set of partitioning planes S_i, 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 of associated partitioning planes S_i:

1. Find a set of removable components R_i of O_i. A component c_i is a removable component of O_i if c_i is disassemblable from O_i. A component is called disassemblable from an assembly, if the component can be translated to infinity by a single translation motion without intersecting with the rest of the assembly. We used algorithm by Woo and Dutta [23] 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 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 is 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.

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 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 (1) c_i lies in the region formed by intersecting half spaces associated with planes in S_i , (2) the intersected region of all half spaces in S_i does not contain any portion of rest of the object assembly, and (3) intersected region of all half spaces in S_i is not finite. The following steps are used to find the set S_i of partitioning planes:

1. Create a set of all the components L 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. Find a valid partitioning 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 and return failure. If a valid separating plane p is found then take the intersection of p with each component in L.
4. If p intersects with a component X of L then decompose X along p into two or more components. Remove X from L and add the decomposed components of X to L.
5. Check if adding p to S_i keeps the intersection of half spaces of planes in S_i semi-infinite or not. If adding p keeps it semi-infinite then 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 and remove p from P.
6. Repeat Steps 3 through 5 until either L or P is empty.
7. If L is not empty and P is empty, then we 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.
    b. 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 $I \neq 0$ for any such pair then component c_i cannot be separated from rest of the components using this algorithm. Find a separating plane p such that a pair of convex hulls (c_h, c_h') lie on opposite sides of this plane (Megiddo [24] describes an algorithm for computing such a plane). Check if adding p to S_i keeps the intersection of half spaces of planes in S_i semi-infinite or not. If adding p keeps it semi-infinite then find that component of L which correspond to c_h' and remove that from L and add p to set S_i.
    c. If L is empty then return S_i otherwise return failure.

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. The above algorithm cannot handle those cases when the candidate partitioning planes in Step 2 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 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 Step 1 described 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. First we decompose a component along all the candidate partitioning planes to get the maximal decomposition set of the component. A maximal decomposition set $A(c)$ of a component c of a multi-material object assembly A is defined as the set of all the components of c generated by decomposing c along all the mating faces with other components in A of different materials. If this decomposition facilitates removal of more than one component then we try to combine those decomposed components to see if a subset of maximal decomposition set can also be removed from the assembly. The following steps describe our algorithm for decomposition of components to facilitate further stage generation:

1. For each c_i in O_i, do the following:
    a. Decompose c_i along all the planes, that are shared by components of other materials to create $A(c_i)$ and make a set D_i from all the elements of $A(c_i)$. (This will result in the maximal decomposition set $A(c_i)=\{c_{i1}, c_{i2}, c_{i3}, \ldots\ldots, c_{im}\}$ of c_i.)
    b. Find a subset R' from the elements of D_i that are removable component of O_i. If R' is empty then go to Step 1 to evaluate next $\underline{c_i}$ in O_i.
    c. Find a pair (c_{ij}, c_{ik}) in R' such that the union of c_{ij} and c_{ik} is also removable from O_i and unite c_{ij} and c_{ik} in D_i. Repeat this step until no such pair is found.

    d. Remove $\underline{c_i}$ from O_i, add elements of D_i in O_i and return success.

2. Return failure. (At this step, decomposition of any of the c_i does not facilitate any component removal so failure is returned.)

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 object assembly $\{c_1, c_2\}$. Figure 6(b) shows that one of the component c_1 can be decomposed into eleven components by partitioning along all the material interface planes. Figure 6(c) shows the final object assembly $\{c_{11}, c_{12}, c_2\}$ after combining some of the components of decomposed c_1.

6 GENERATION OF MOLD-STAGE

The algorithm described in Section 4 generates an ordered set $CS = ((c_1, S_1), (c_2,S_2), (c_3, S_3), \ldots\ldots\ldots\ldots, (c_f, S_f))$ of homogeneous component and set of partitioning planes pair that corresponds to a feasible mold-stage-sequence. Figure 7 shows a feasible sequence of object partitioning for the example part. Once we find the set of partitioning planes S_i, that can be used to separate the object component c_i from the object assembly O_i, we perform partitioning of mold-pieces using the 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 d_i of the same geometry as the component c_i. The removal of mold-piece d_i requires partitioning of mold-pieces. Mold-pieces are partitioned along the planes in set S_i so that mold-piece d_i can be assembled and disassembled. The mold-piece d_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 generate mold-pieces in various mold-stages:

1. Initialize a set of mold-pieces J with the gross mold of the final multi-material object O_f. The solid model of the gross mold for the 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 object c_i in CS for $i= f$ to 2, where f is the total number of stages needed to produce the object and is equal to the number of elements in CS (this would consider the c_i's in the order produced by object decomposition algorithm discussed in Section 4), do the following:
    a. Copy set J to set K. For each plane p in each set of partitioning planes S_j (for $j = f$ to i), find the mold-pieces in K, which are completely on the side of p that does not contain c_i and remove them from K. (These mold pieces do not pose any problem for removal of c_i. So we do not need to consider those mold-pieces for decomposition in subsequent steps and remove them from K.)
    b. Subtract K from J.
    c. For each plane p in S_i, decompose the mold-pieces in K along p. Find the set of mold-pieces in K, which are not on the same side of p as c_i and remove them from K and add them to J.
    d. Make a mold piece m_i'' which is union of all the mold-pieces in K and d_i.
    e. Add all the mold-pieces of K to M_i^a and m_i'' to M_i^d.
    f. Create a mold-stage t_i as M_i^d being the set of mold-pieces that has to be removed from the previous stage and elements in M_i^a being the mold-pieces that has to be added to produce the mold-assembly M_{i+1}. Add the mold-stage t_i at the beginning of the mold-stage list T.
    g. Add m_i'' to J.
3. Create the first mold-stage as all the mold-pieces in J being the set of mold pieces M_1^a that has to be added to produce the first stage mold-assembly and set M_1^d to be null set for the first mold-stage.
4. Set starting mold assembly M_1 for the first stage as to null. Compute the starting mold assembly for each stage for i=1 to f-1 as $M_{i+1} = (M_i - M_i^d) \cup M_i^a$.

The above steps produce all the mold-stages required to produce the multi-material object. Figure 8 shows the generation of the last mold-stage for the example part. Set J of mold-pieces is initialized with the gross mold of the final multi-material object in Step (1). Figure 8(a) shows the mold-piece j_1 of J, component c_8, mold-piece d_8 (of the same geometry as c_8) and the partitioning plane p_{81} used to create the last mold-stage. The mold-piece in J is partitioned along the plane p_{81} as shown in Figure 8(b) and mold-piece m_8' is identified which can be combined with d_8. It then finds the mold-piece m_8'' that is to be removed from the previous stage. m_8' is then added to M_8^a and m_8'' to M_8^d. Figure 9 shows the mold-stage assemblies of all the stages produced using our algorithm for the example part and the state of object assembly after each mold-stage.

7 IMPLEMENTATION

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. It also has animation capability to show assembly and disassembly sequences.

We currently use Polyurethane to fabricate multi-material objects. Polyurethane (PURs) are a group of polymers with highly versatile properties and a broad range of commercial applications. PURs can be manufactured in an extremely wide range of grades, in polymer stiffness from very flexible elastomers to rigid hard plastics. Most of the PURs cure at room temp so no special curing facility is needed for these parts. Multi-material polymer objects are easy to manufacture by casting PUR into multi-stage sacrificial molds (i.e., molds are destroyed to remove the object) made of wax. The multi-stage molds are first fabricated from machinable wax. Molds are machined using CNC milling. We use a 3-axis machining center and a turning center to machine mold. After the mold-pieces are machined at the machining facility, they are cleaned using acetone to remove dust particles that may get trapped in the mold cavities. Cleaning of dust particles is required to get a good surface finish. Then PUR components are mixed in the specified ratio by weight or volume. First mold stage assembly is assembled and mixed PUR components are poured into the mold-stages. The partially cast part is then left to set for the required demolding time. The demolding time typically varies from 2 hours to 10 hours depending on the grade of PUR used. Current mold-stage assembly is then transformed to form the mold assembly for the next stage by adding and removing the required mold-pieces and other portion of the object is cast. This is repeated to carry out all the mold-stages to produce the desired multi-material object. The cast object is demolded by removing the wax mold-pieces and the fabricated object is then cured for the ultimate cure time. The ultimate cure time varies from 2 days to 10 days. Figure 10 shows the picture of the part that was produced using the mold-stage sequence shown in Figure 9.

8 CONCLUSIONS

8.1 Summary

This paper describes a geometric algorithm for generating mold-stages for multi-stage molding. Novel features of this algorithm are:

- It finds multiple partitioning planes to perform partitioning on the mold-pieces.
- It performs object and mold decomposition 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 the mold-pieces to be added and to be removed from the previous stage to produce the mold assembly at each stage.

Because of the page restriction the correctness proofs of the algorithms described in this paper are not included in this paper. These proofs can be found in [25].

8.2 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 large or the objects are of large size. Therefore, development of automation technology that can reduce the lead-time for mold manufacturing will help in making molding of multi-material objects commercially viable for medium to large 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.

8.3 Current Limitations

Our 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 object decomposition algorithm does not always find a feasible object partitioning sequence because it only decomposes components along the material interfaces.
- 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 backtracking in case a feasible object decomposition sequence does not give a feasible mold sequence.

Acknowledgement. This research has been supported by NSF grant DMI0093142 and ONR grant N000140010416. Opinions expressed in this paper are those of authors and do not necessarily reflect opinion of the sponsors.

9 REFERENCES

[1] R. Merz, F. B. Prinz, K. Ramaswami, M. Terk, and L. Weiss. Shape Deposition Manufacturing. *Proceedings of the 1994 Solid Freeform Fabrication Symposium*, 1994.

[2] 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.

[3] K. Ramaswami, Y. Yamaguchi, 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.

[4] 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.

[5] 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.

[6] V. Kumar, and D. Dutta. An Approach to Modeling Multi-Material Objects. *Proceeding of 4th ACM Symposium on Solid Modeling*, Atlanta, GA, May 1997.

[7] 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.

[8] Y. S. Suh and M. J. Wozny. Adaptive slicing of solid freeform fabrication processes. In p*roceedings of Solid Freeform Fabrication Symposiu*m, Austin, Texas, August 1994.

[9] 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 Conferenc*e, Las Vegas, Nevada, September, 1999.

[10] L. Chen, and T. C. Woo. Parting directions for mold and die design. *Computer Aided Design*, 25(12):762-768, 1993.

[11] 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.

[12] K. C. Hui, and S. T. Tan. "Mould Design with Sweep Operations – a Heuristic Search Approach". *Computer Aided Design*, Vol. 24(2), 1992.

[13] B. Ravi, and M. N. Srinivasan. Decision Criteria for Computer-Aided Parting Surface Design. *Computer Aided Design*, Vol. 22, No. 1, 1990.

[14] M. Weinstein and S. Manoochehri. Optimum Parting Line Design of Molded and Cast Parts for Manufacturability. *Journal of Manufacturing Systems*, Vol. 16(1), 1997.

[15] D. W. Rosen. Towards Automated Design of Molds and Die*s.* In *Proceedings ASME Computers in Engineering Conferenc*e, Minneapolis, September 1994.

[16] K. H. Shin, and K. Lee. Design of side cores of injection moulds from automatic detection of interference faces. *Journal of Design and Manufacturin*g, vol. 3, pp. 225--236, 1993.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

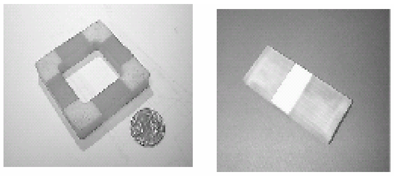
[21] H. Wu, E. M. Sachs, N. M. Patrikalakis, D. Brancazio, J. Serdy, T. R. Jackson, W. Cho, H. Liu, M. Cima, and R. Resnick. Distributed Design and Fabrication of Parts with Local Composition Control. *Proceedings of the 2000 NSF Design and Manufacturing Grantees Conference*, Jan 2000.

[22] 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.

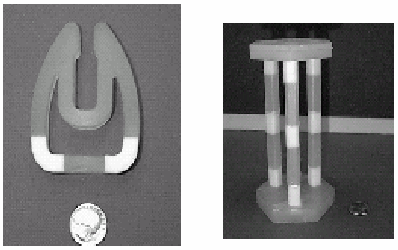
[23] D. Dutta, and T. C. Woo. Automatic Disassembly and total ordering in three dimensions. *ASME Journal of Engineering for Industry*, 1991, 113(1), 207-213.

[24] 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.

[25] M. Kumar. *Automated Design of Multi-stage Molds for Manufacturing Multi-material Objects*. M.S. Thesis, Department of Mechanical Engineering, University of Maryland, June, 2001.



(a) Four-Bar Linkage (b) Snap-fit Cap



(c) Gripper (d) Torsion Table

Figure 1. Example of Multi-Material Objects

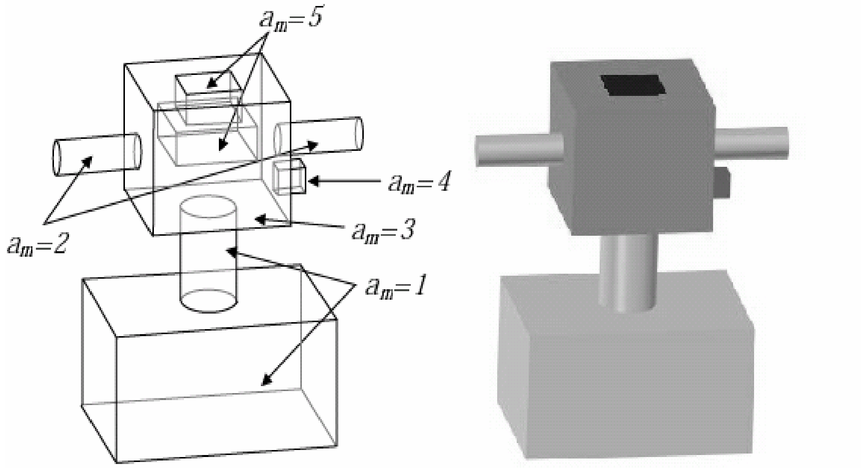


Figure 2. A multi-material object modeled as an assembly of homogeneous components

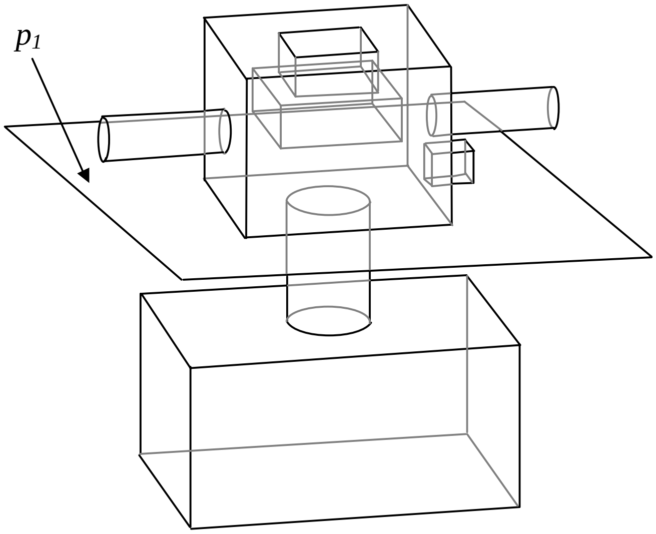


Figure 3. A valid single partitioning plane to separate a component out from rest of the assembly

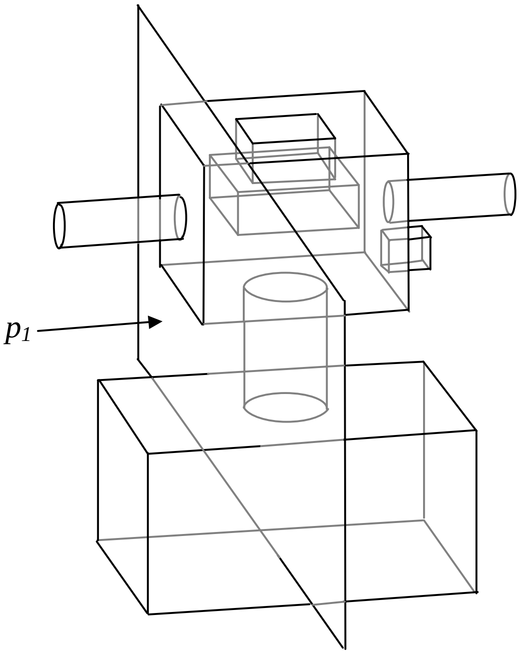


Figure 4. An invalid single partitioning plane

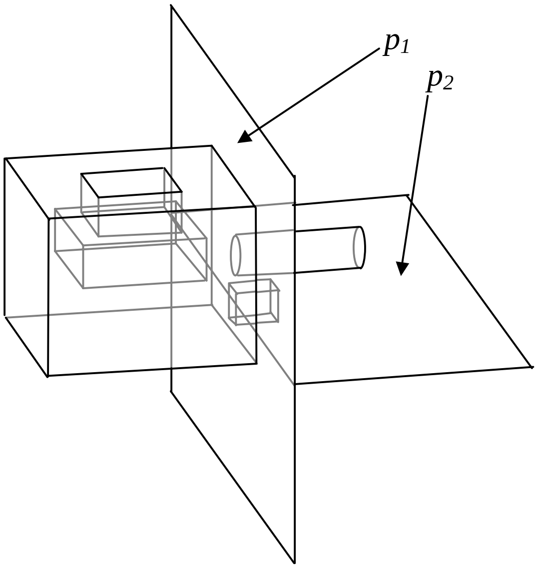
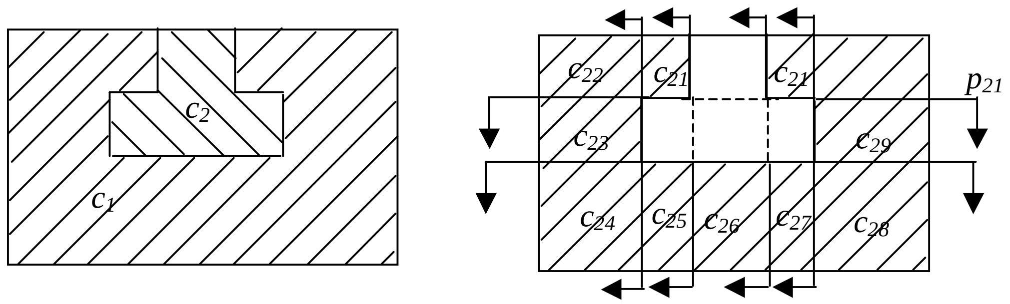
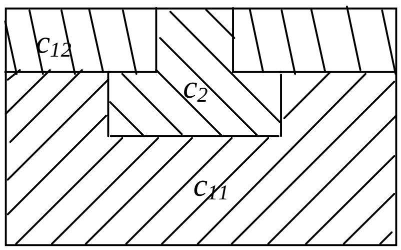


Figure 5. A set of partitioning planes consisting of two planes used to separate a component from assembly



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

(b) Component c_1 decomposed along all the candidate partitioning plane



(c) Final component assembly after decomposition of c_1 and some components of c_1 has been united back

Figure 6: Example of Component Decomposition to Facilitate Mold Stage generation

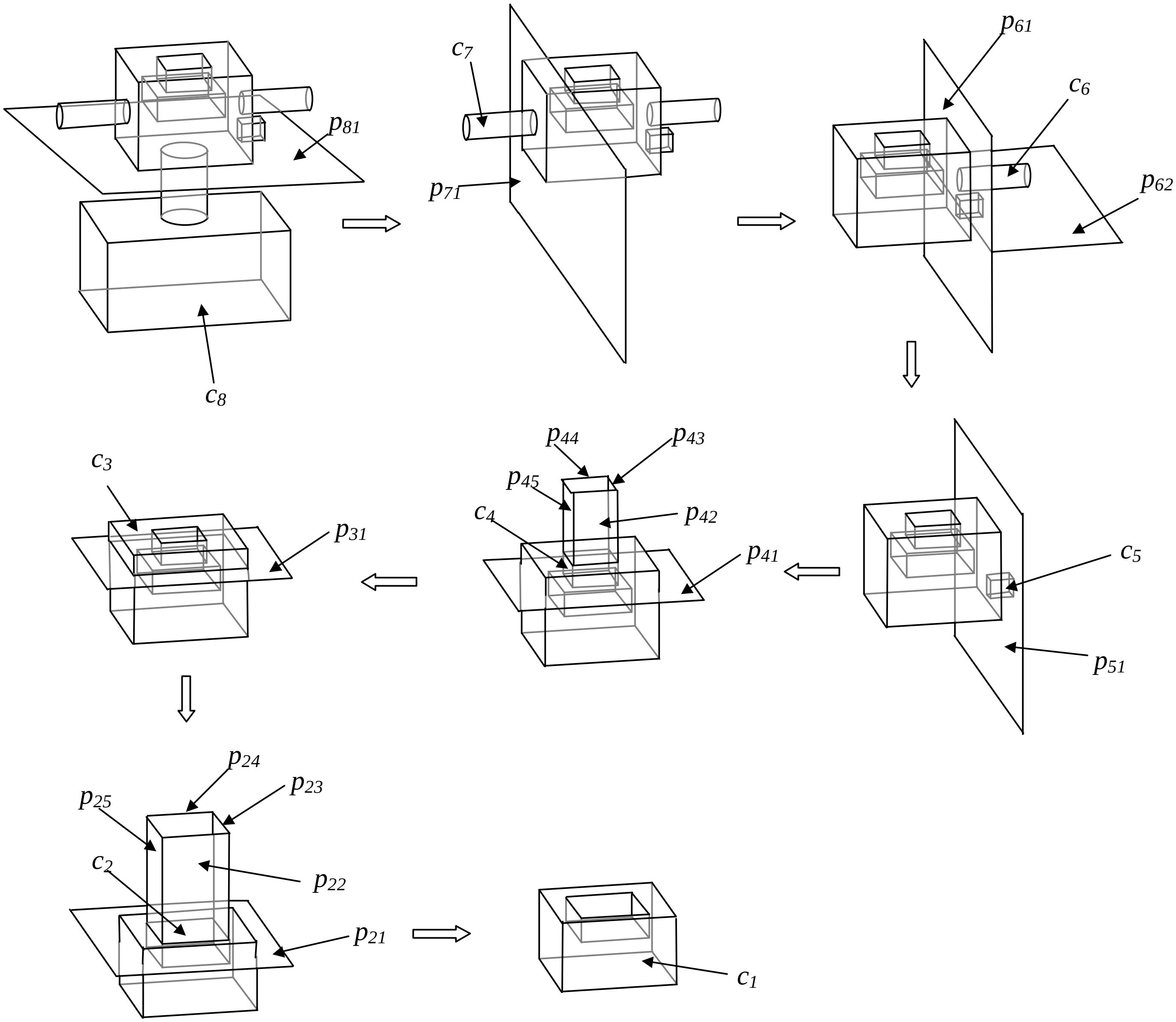
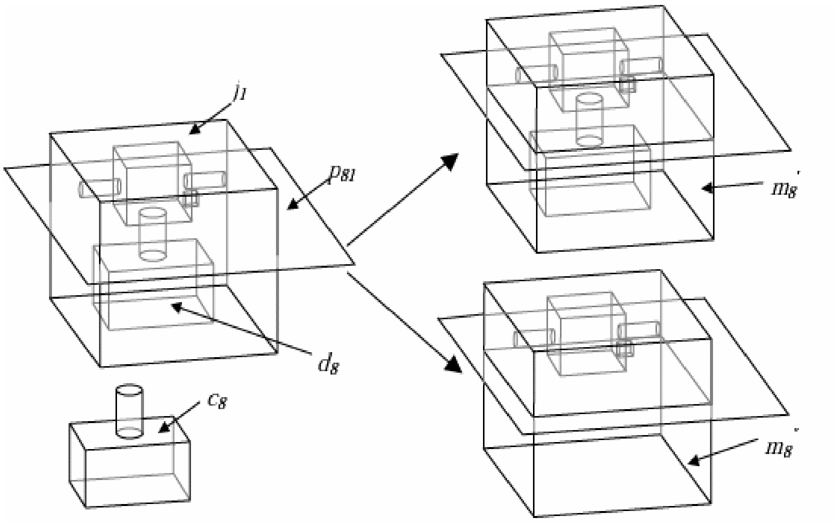


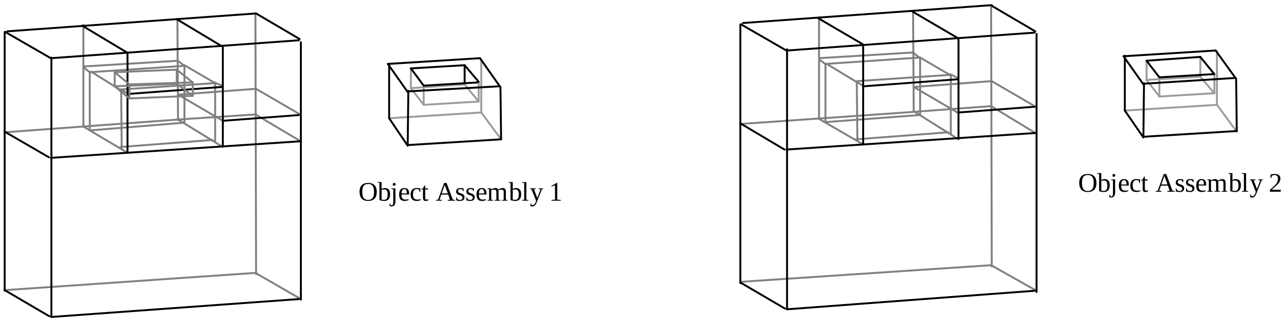
Figure 7. A Feasible Object Partitioning Sequence



(a) Partitioning plane and component added in the last mold-stage

(b) Mold-pieces involved in last mold-stage transformation

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

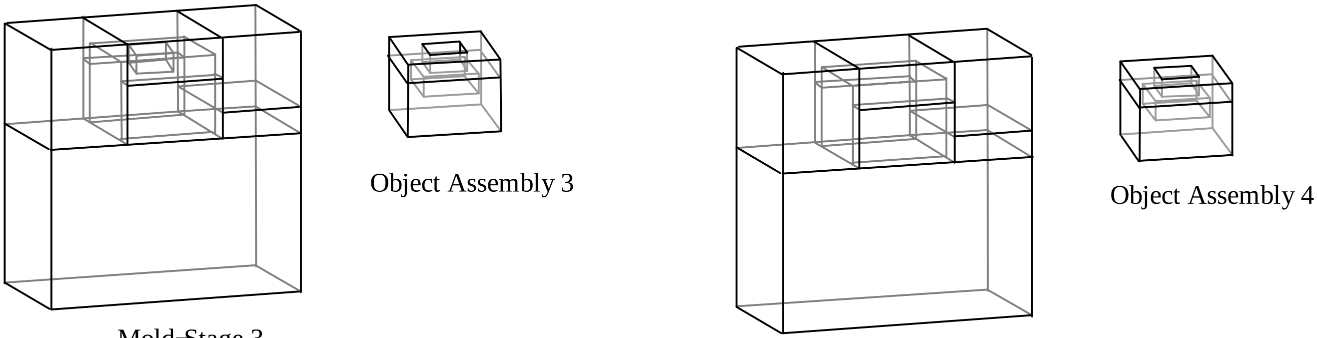


Mold-Stage 1

Object Assembly 1

Mold-Stage 2

Object Assembly 2

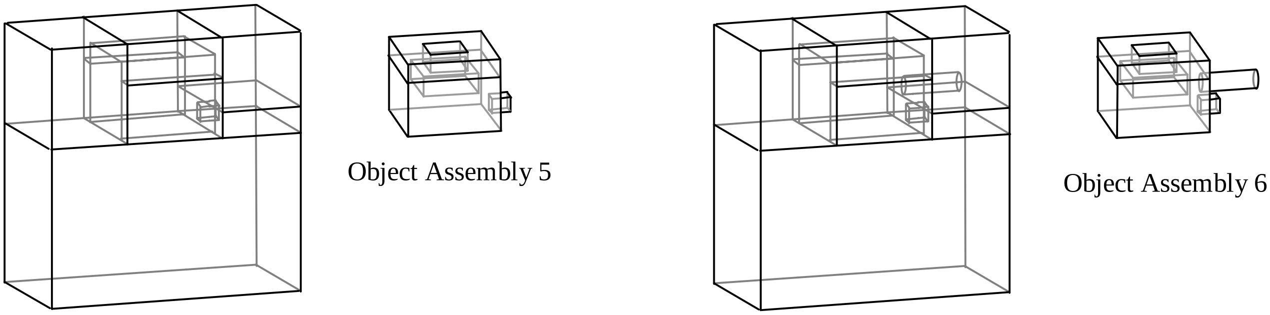


Mold-Stage 3

Object Assembly 3

Mold-Stage 4

Object Assembly 4

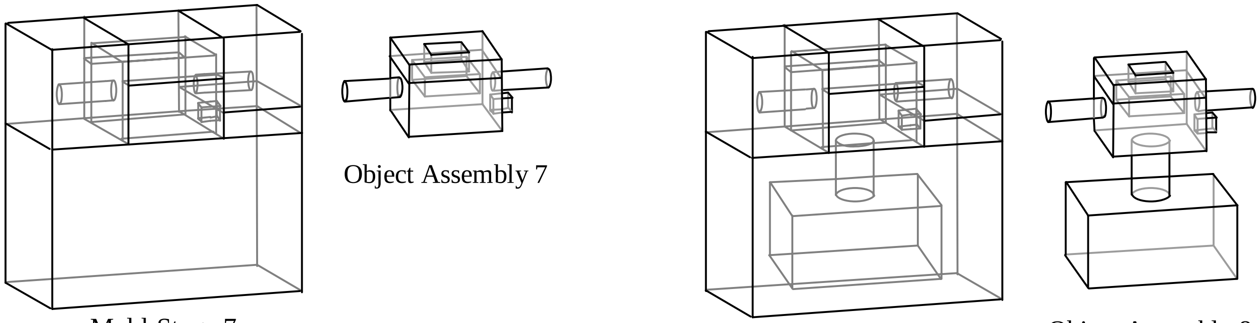


Mold-Stage 5

Object Assembly 5

Mold-Stage 6

Object Assembly 6



Mold-Stage 7

Object Assembly 7

Mold-Stage 8

Object Assembly 8

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

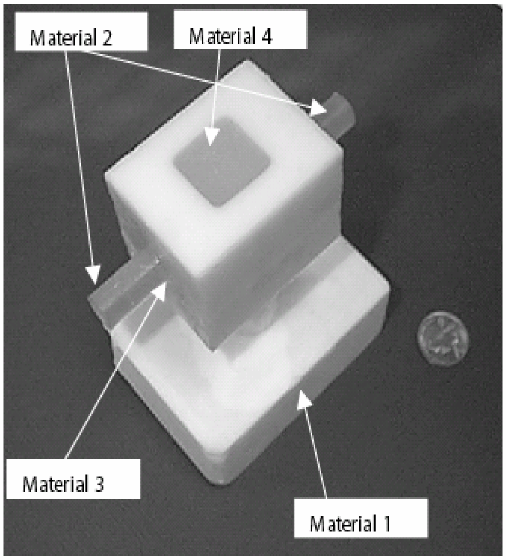


Figure 10. Fabricated Multi-material part using the stage sequence shown in Figure 9