Proceedings of IMECE04
2004 ASME International Mechanical Engineering Congress and Exposition
November 13-20, 2004, Anaheim, California USA

IMECE2004-61845

MULTIPLE PARTING SURFACES FOR SAND CASTING

Hemant Ramaswami
Computer Aided Manufacturing Laboratory
Industrial and Manufacturing Engineering Program,
University of Cincinnati,
Cincinnati, OH 45221-0072.

Sam Anand
Computer Aided Manufacturing Laboratory
Industrial and Manufacturing Engineering Program,
University of Cincinnati,
Cincinnati, OH 45221-0072.
Sam.Anand@uc.edu

ABSTRACT

Multiple parting surfaces are frequently used in sand casting, die casting and injection molding processes. However, most research in this area has focused on die casting and injection molding. Parting surfaces for die casting and injection molding are relatively easier to compute compared to sand casting because their orientations and shapes are less restricted. In sand casting, the parting surfaces have to be parallel to each other and perfectly flat to permit the use of flasks with more than two pieces. The concepts of visibility and object illumination can be used to divide an object into two parts using a single parting surface. These methods, however, cannot be directly used for multiple parting surfaces.

In this paper, a methodology to generate multiple parting surfaces for sand casting is described. The method uses Gauss maps to identify potential casting directions, and global accessibility cones to determine which faces can be cast in the same part of the pattern. The pattern is sliced using parallel planes such that each slice can be withdrawn from the mold in at least one direction. After the object is sliced, the number of parting surfaces is reduced by combining adjacent middle sections depending on their accessible directions.

Keywords: Parting line, parting surface, sand casting, accessibility, Gauss maps, global accessibility cones.

INTRODUCTION

Expendable mold casting is among the most widely used methods in manufacturing. Among expendable mold castings, sand casting is extensively used due to its versatility. Virtually any shape can be formed in the sand to produce molds that can be used for casting. This process is slow, but is very economical for low quantities, intricate designs, and for large sized castings [1, 2]. One of the initial stages in sand casting is patternmaking, and generation of the parting line or parting surface is a very critical part of this stage.

Conventional sand casting uses two flasks, viz. a cope and a drag, to enclose the mold. However, the complexity of the parts that can be produced in this way is limited. Parts with higher complexity can be cast using two-piece molds with the use of loose-piece cores. However, the cost and set-up time required can be prohibitive. Also, some very complex parts cannot be cast using a two-piece mold. By using additional flasks between the cope and the drag, parts with increased complexity can be produced, with a substantial reduction in effort and time. Most research on castability and parting line generation considers only a single parting line [3-9]. Other research dealing with parting line generation for die casting and injection molding considers multiple parting lines [10-13]. However, since the focus is on injection molding and die casting, the orientation of the parting surfaces is not constrained. Also, since the entire part has to be ejected without intersections with the die, the accessibility considerations differ from those for sand casting.

PROBLEM FORMULATION:

Given a polyhedral object, design the multi piece pattern with the following inputs and associated outputs.

Input:

- Solid geometric model of the polyhedral part
- Machining volumes
- Core and coreprint size and locations

Output:

- Assembly of solid geometric models of pattern parts and the associated parting lines or parting surfaces and the casting direction such that:
  - o The union of all the pattern pieces is exactly the union of the part, the machining volumes and the cores and coreprints, i.e.

$$\sum_i (B_{pattern})_i = B_{final} + B_{machined} + B_{shrinkage} + B_{core} + B_{coreprint} \quad (1)$$

where:

$(B_{pattern})_i$ = solid model of slice i of the pattern
B_{final} = solid model of the final component
$B_{machined}$ = solid model of the machined volumes
$B_{shrinkage}$ = shrinkage allowance

B_{core} = solid model of areas to be cored

$B_{coreprint}$ = solid model of coreprint volumes

- Each part is accessible in at least one direction, i.e. each part can be translated in at least one direction without causing any damage to the sand mold.

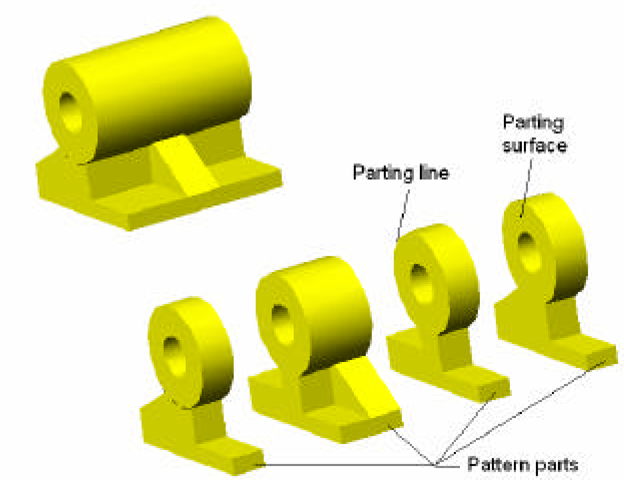


Figure 1: A solid and its pattern pieces

LITERATURE REVIEW

Considerable research has been carried out in the area of casting automation. Parting line generation and development of heuristics to assist in selection of parting surfaces when more than one option is available have been the focus of many of these efforts.

Single Parting Line Generation:

Ahn et al. [3] consider the case where given an object and a direction, *d*; the cast can be divided into two parts such that the parts can be moved along the directions *d* and *–d* without colliding with each other or the mold. They also describe an algorithm to find a set of directions which permit such a removal. Ahn et al. [4, 5] address the problem of casting a part in two pieces such that one piece of the mold retracts carrying the part with it. The object is then ejected from the part in another direction. They use the concept of visibility and illumination under lights from two different directions to propose an O (n^3 log n) algorithm for polyhedral objects in [4]. Subsequently, in [5], they improve the running time of the algorithm to O (n^2 log n). Fu et al. [6] extensively discuss about the visibility of different types of surfaces found in solids, and the relation between visibility and moldability. The visibility information is used to classify the surfaces into core-molded, cavity-molded, or local tool-molded (requiring the use of cores) surfaces. They use this classification to determine the location of the parting line. Opposite directions are considered for translating the two mold parts, and additional directions for translating the cores. Majhi et al. [7] present algorithms to compute undercut-free parting lines to divide convex polyehedra into two parts based on two alternate criteria, one on the length of the parting line, and second on the position of the vertices. Chen et al. [8] describe a deterministic procedure to compute an optimal pair of parting directions which minimizes the number of cores required. Hui and Tan [9] propose a method based on solid-sweep operations to obtain the core and cavity for die-cast or injection-molded parts. They also provided heuristics based on the undercut areas and projected area to select a parting direction.

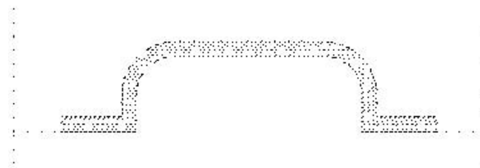


Figure 2: A two-piece (single parting line) mold (from Chen et al [8])

Multiple Parting Line Generation:

Weinstein and Manoochehri [10] use a multi-objective optimization based on the complexity of the parting line, draw depth, number of cores and other parameters to obtain an optimal mold design. They use a tree structure with leaves representing the surfaces formed by one mold half, and use this information to generate the parting lines. Priyadarshi and Gupta [11] suggest an algorithm to produce multi-piece molds for injection molding and die casting. The algorithm is based on the global accessibility of the part, and ensures complete disassembly of the mold pieces. They use heuristics to obtain the set of candidate parting directions, and use exact accessibility analysis to compute additional directions to access facets that are not accessible from the initial set of directions. Dhaliwal et al. [12] describe a feature-based algorithm for automated design of multi-piece sacrificial molds. The algorithm creates the part geometry based on a feature-based description. If the mold shape is not machinable as a single component, then the gross mold shape is decomposed into simpler geometric components to make sure that each component is machinable. Chen and Rosen [13] present a region-based approach to automated mold design that is suitable for two and multi-piece molds. They attempt to minimize the number of mold pieces by grouping different partitions of faces into regions until the smallest number of regions is found. They also use a linear programming approach for finding a satisfactory parting direction of a region.

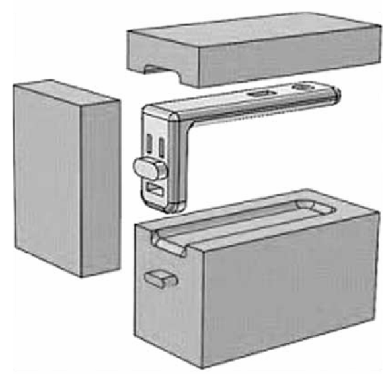


Figure 3: A multi-piece mold (from Priyadarshi and Gupta [11])

Decision Criteria:

Ravi and Srinivasan [14] propose a list of design criteria to be considered during parting line generation. These include the projected area, flatness, draw depth, draft requirements, undercut areas, dimensional stability, flash, location of surfaces requiring post-casting machining and directional solidification. They also give detailed algorithms for implementation of some of the factors. The final decision is made by using a weighted sum of all the factors, i.e. select parting surface PS_j such that for all k,

$$\sum_i w_i c_{ij} \leq \sum_i w_i c_{ik} \quad (2)$$

where:

w_i = weight associated with factor i

c_{ij}=assessment value of criterion i for parting surface j

c_{ik}= assessment value of criterion i for parting surface k

Ganter and Tuss [15] use a rule set based on “rules of thumb” collected from various sources and surveys to determine the parting direction. They use a subset of the collected rules, considering only the draw depth, projected area, draft, and irregularity of the parting line. Smith and Lee [16] illustrate an interactive procedure to obtain the proper pattern for a casting from the description of the final product. The steps involved in the procedure include hole and cavity elimination, adding of shrinkage allowance and draft and cores. A limitation of this approach is that the parting lines are constrained to be parallel to one of the co-ordinate planes.

Accessibility:

Accessibility of a solid is a property that is used in determining many manufacturability aspects, viz. CNC machining, casting, and inspection. Dhaliwal et al. [17] describe an algorithm to compute the global accessibility cones for polyhedral objects modeled using facets. They discuss the mathematical conditions for determining the exact inaccessibility of any surface. Knowing the exact inaccessibility, it is possible to compute the accessibility cone for each facet. Elber [18] uses hidden line and surface removal algorithms to solve the global interference problem for convex machined surfaces. A 5-axis global accessibility problem is reduced to a 3-axis global accessibility problem which is relatively easier to solve. Woo [19] classifies manufacturing processes based on their visibilities. The part geometry is also mapped onto a unit sphere using a visibility map. The problem of setting up a part on a machine is reduced to finding a maximal intersection between the visibility map of the object and a sphere representing the visibility map of the machine. Suh and Kang [20] use accessibility analysis to determine the setup orientation for NC machining of freeform surfaces. They compute the accessibility by faceting a unit sphere, and projecting the centroids of the facets on the surface to facets on the sphere. However, this approach is approximate and is subject to errors in the boundary cases. Spitz et al. [21] use accessibility analysis to find an interference free path for a CMM for inspection of freeform surfaces. However, in the case of inspection using CMMs, global accessibility of the entire surface is not essential. It is sufficient to ensure accessibility of only the inspection points. Lim and Menq [22] also use accessibility analysis to determine the accessibility of features and to generate paths for CMM inspection. They use ray tracing to compute the accessibility cones. The direction cone is discrete in nature representing the different orientations possible using the probe considered in the paper.

Computational geometry methods have previously been adapted and used in the fields of automation and CAD/CAM integration. This paper employs the concepts of transformations across different spaces using Gauss maps and global accessibility cones of surfaces to compute the parting lines for sand casting. The selection of the optimum parting surfaces is based on a weighted sum of heuristic measures, which are derived from the geometry of the pattern pieces.

METHODOLOGY

The algorithm described in this paper accepts the solid model of the component, the post-casting machined volumes and the expected core and coreprint volumes as input. The pattern model obtained is then analyzed to determine potential casting directions. The global accessibility of each surface is also computed to determine the castability of the part. The part is then analyzed for casting in each potential direction and the optimal solution is determined by using a weighted sum of performance measures defined later in the paper.

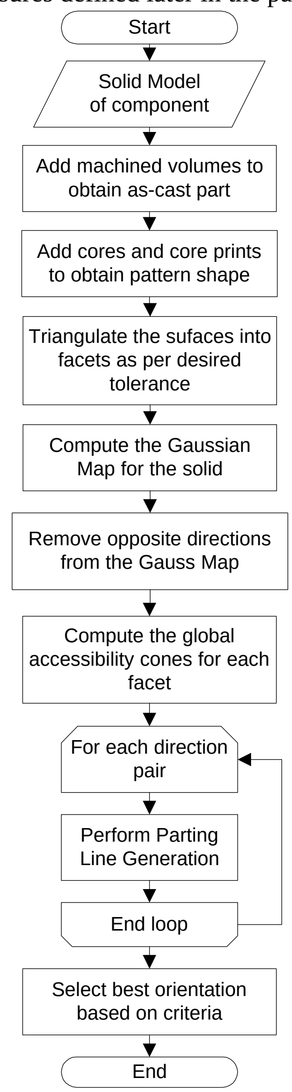


Figure 4: Flowchart of entire procedure

The steps followed by the algorithm are shown in Fig. 4. The details of each stage are explained using a sample part shown in Fig. 5.

Generate pattern shape:

The input to the algorithm is the model of the component. The algorithm also accepts as input, the features that are created by machining the cast part. The user also provides information regarding any cores that may be required to produce cavities. The final pattern shape is obtained as the union of the component model, the machined volumes and the core and coreprint volumes. The faces of the final pattern shape are then triangulated into facets for use in the following steps of the algorithm. Curved surfaces are also triangulated based on a pre-specified tolerance value. This approximation reduces the computation in later stages. If higher accuracies are desired, the tolerance value can be kept very small, and a fine triangulation can be obtained.

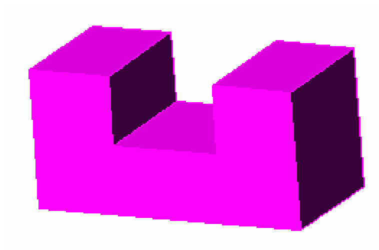


Figure 5: A sample part

Identify potential casting directions:

Any surface in Euclidean space can be mapped onto a unit sphere by translating the unit normal at every point of the surface to the origin. Weisstein [23] defines a Gauss map (GMap) as a function from an orientable surface in Euclidean space to a sphere. A GMap associates to every point on the surface its normal vector. Thus, the GMap of a flat surface is nothing but a point on the sphere, that of a polyhedron is a set of points, and that of a curved surface is a region on the sphere. If, in a casting, a parting surface is parallel to a face, the face in question will never be split between pieces. This is therefore a good starting point to decide on the casting directions. All the directions in the GMap are considered as potential casting directions. The GMap for the solid shown in Fig. 5 is shown in Fig. 6.

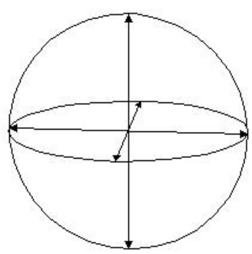


Figure 6: Gauss Map of solid in Fig. 5

Analyze the potential casting directions:

The potential casting directions are used to define the slicing plane which will be used to partition the pattern. The slicing plane is taken normal to the casting direction. A plane normal to a direction *d* will also be normal to the direction opposite to *d*, i.e. –*d*. Thus, any direction *d*, and its opposite, -*d*, will produce the same combination of pattern pieces. This result is very useful because it can be used to eliminate one direction from the set if a direction and its opposite are present simultaneously. This considerably reduces the number of directions that need to be checked. The reduced GMap for the GMap in Fig. 6 is shown in Fig. 7.

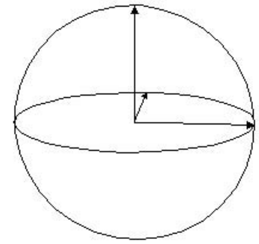


Figure 7: Reduced Gauss Map for Fig. 6

Compute the Global Accessibility Cones:

The Accessibility Map of any surface is a set of directions from which the surface is globally accessible. Global accessibility in a direction implies that the surface can be translated to infinity in that direction without being obstructed by any other part of the mold or any other surface in the interior of the part. This is an important requirement for the object to be castable. If each face of the part is globally accessible from at least one direction, then the part is castable, because the face can be translated along that direction to infinity without interfering with any other face. In this research, a requirement of global accessibility in at least one direction is imposed. It is further assumed that cores will be required to cast any face that does not satisfy this criterion. Some additional properties of polyhedra that prove to be useful and mentioned in Chen et al [8] are:

- The global accessibility cone of a face which lies on the convex hull of the part is a hemisphere with the direction normal as the pole.
- For a facet that does not lie on the convex hull of the part, the accessibility cone is less than the hemisphere. Additionally, the facet will be blocked only by facets in the same concave region. This property also simplifies the computation of the global accessibility cones for non-convex hull facets.

The Global Accessibility Cones are computed using the algorithm proposed by Dhaliwal et al. [17]. The approach uses an *"initialize-update"* scheme. The unit sphere is faceted using spherical triangles, and the accessibility information is stored as a matrix with the indices identifying the facet and the spherical triangle. If a facet is accessible from a triangle, the corresponding entry in the matrix is one, else it is zero. All the convex-hull facets on the part are accessible from all the orientations contained within a hemisphere created by the facet's outward normal as the pole. For non-convex-hull facets, the algorithm initially assumes accessibility from all the orientations in the hemisphere. The directions are subsequently trimmed after checking the influence of other facets. The dimensions of this matrix are dynamic and finer triangles are added to the sphere on an as-needed basis. The addition of finer triangles does not affect the accessibility of the facets that have already been analyzed because, if a facet on the part is accessible from a spherical triangle, it will be accessible from all spherical triangles that are formed by decomposing the original triangle. This repeated division of facets on an as-needed basis permits the exact representation of the global accessibility cones. The advantage of representing the accessibility cones as a matrix is realized in the later steps when the accessibility for multiple facets needs to be compared. Figure 8 shows the accessibility cones for all the faces of the sample part. In this case, the surfaces have not been faceted so that the illustration can be made clear.

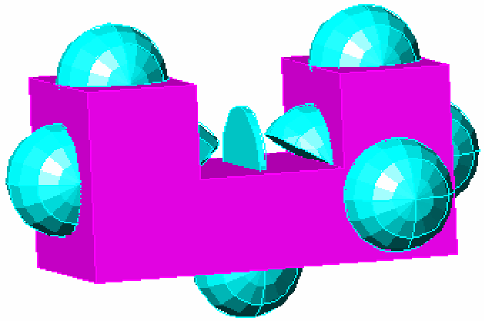


Figure 8: Accessibility cones for sample part

Generate the parting surfaces:

This stage is the most computationally intensive stage of the algorithm. Once the potential casting directions have been determined, and the accessibility cones computed, the pattern has to be analyzed along each direction to identify where the parting lines need to be located.

The analysis is started by defining a plane normal to the desired casting direction. This plane is swept across the solid to identify the sequence in which the facets are traversed. An array of length equal to the number of spherical triangles used for the accessibility cones is defined and initialized to one. As the sweep plane intersects or grazes the first facet, the array elements corresponding to the accessibility direction for the facet are updated with a value set to one.

$$new\_array = old\_array \cap new\_face\_dirns \qquad (3)$$

where:

new_array = array of accessible directions after including current face

old_array = array of accessible directions before including current face

new_face_dirns = row of accessibility cones for current face.

Figure 9(a) illustrates the sweep plane for a particular orientation. A coarsely faceted sphere used for illustration of accessibility is shown in Fig. 9(b). Table 1 shows the global accessibility cones matrix for the six faces indicated in Fig. 9(a) using the sphere in Fig 9(b).

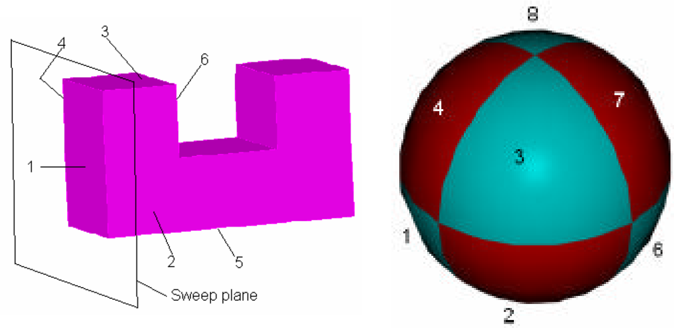
When the plane sweep starts, the array will be initialized with one in all cells. When the plane intersects with face 1, only the cells corresponding to the accessible directions for face 1 will be left unchanged, and the rest will be changed to zero. The array will be [1 1 1 1 0 0 0 0] at this stage.

As the sweep continues, new facets will intersect the plane. Whenever a new facet intersects the plane, the array is updated to represent the intersection of the accessibility directions of the new facet and the current contents of the array.

If multiple facets intersect the plane simultaneously, the intersection of accessibility directions of both the facets is used to update the array.

$$new\_array = old\_array \cap (\bigcap_i new\_face_i\_dirns) \qquad (4)$$

where $new\_face_i\_dirns$ is the row of the accessibility cones matrix corresponding to i^{th} face intersected.



(a) (b)

Figure 9: Casting direction and determination of accessible directions

In Fig. 9, when the sweep continues, the next faces to intersect the plane are 2, 3, 4 and 5 simultaneously. The facet with the lowest index is selected first to compute the intersection. After computing the intersection of the array with the directions for face 2, the array will be reduced to [0 1 1 0 0 0 0 0]. This process is repeated for all the other surfaces. The array now represents the directions from which the solid traversed so far is accessible.

Table 1: Simplified Accessibility cone matrix for simplified sample part shown in Fig. 9

		Spherical Triangles							
		1	2	3	4	5	6	7	8
Face	1	1	1	1	1	0	0	0	0
	2	0	1	1	0	0	1	1	0
	3	0	0	1	1	0	0	1	1
	4	1	0	0	1	1	0	0	1
	5	1	1	0	0	1	1	0	0
	6	0	0	0	0	1	1	1	1

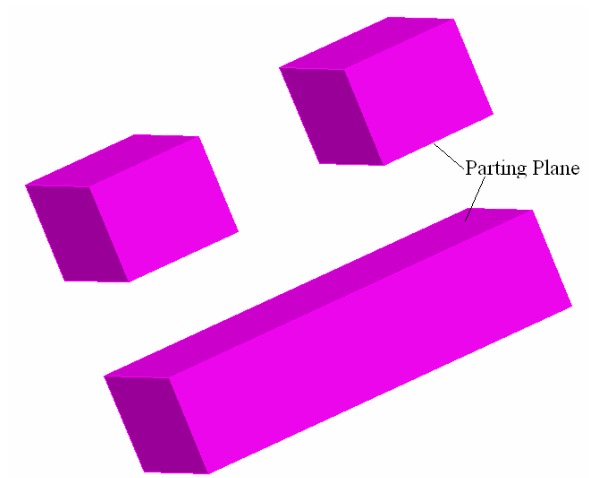
This process is continued until the array becomes empty. At this point, any further addition to the traversed solid will prevent the ejection of the pattern from the mold. When the array becomes empty, the traversal is reversed one step backward until there is at least one direction in the array. The location of the sweep plane determines the position where the pattern needs to be sliced.

In this example, because the triangles are very coarse, the array will be completely empty when all the four faces have been considered. However, with a finer triangulation on the sphere, in this case, there will be at least one non-zero element corresponding to the casting direction. This will be the set of accessibility directions for the solid traversed till now.

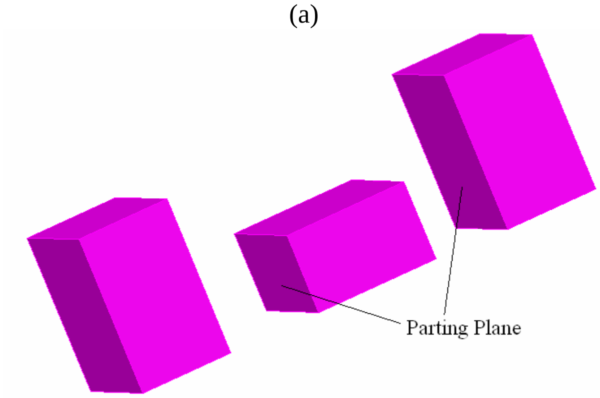
As the sweep continues in Fig. 9, the plane will intersect face 6. It can be observed that the intersection of the accessibility directions of faces 1-5 is only a single direction along the normal direction. The intersection of this set and the accessibility directions of face 6 will be a null set. This indicates that the direction array has no elements with the value of 1. Now, the plane is moved back to a position where it has not intersected face 6, and the solid is sliced at that point.

Once the pattern is sliced at this position, the remainder of the pattern is used for further computation. The array is reset and the process is repeated until the entire pattern is divided into slices.

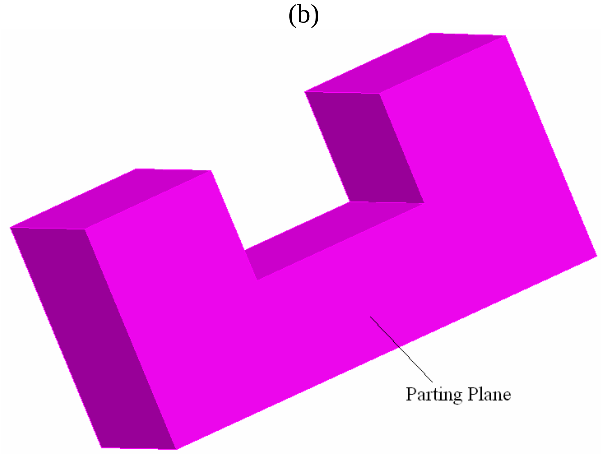
In the above example, if the normal direction is not represented explicitly in the accessibility array, the algorithm will incorrectly report that this direction is infeasible. To avoid such erroneous results, the accessibility in the normal direction is checked independently.



(a)



(b)



(c)

Figure 10: Parting lines for sample part

Figure 10 illustrates the parting planes for the sample part shown in Fig. 5.

Reduction of parting surfaces:

After the pattern has been sliced completely, the next step is to analyze the slices to check if any intermediate pieces can be molded in a single flask. This is important, because this will reduce the number of parting surfaces required. To determine this, first, outward normals are drawn from the common surface in both directions, as shown in Figs. 11 and 12.

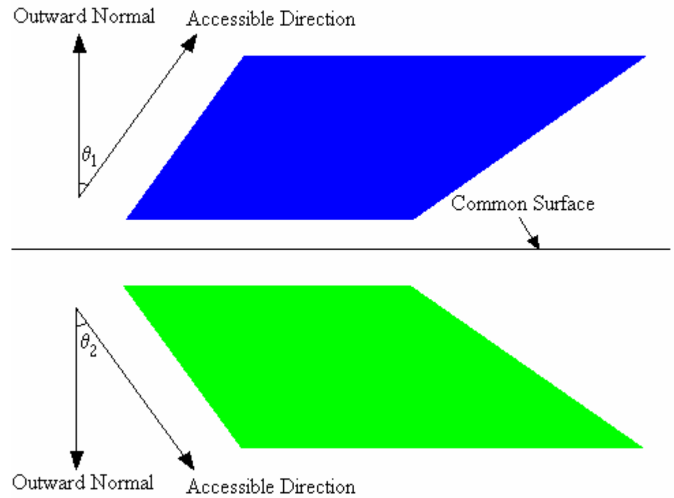


Figure 11: Condition for feasibility of parting surface reduction

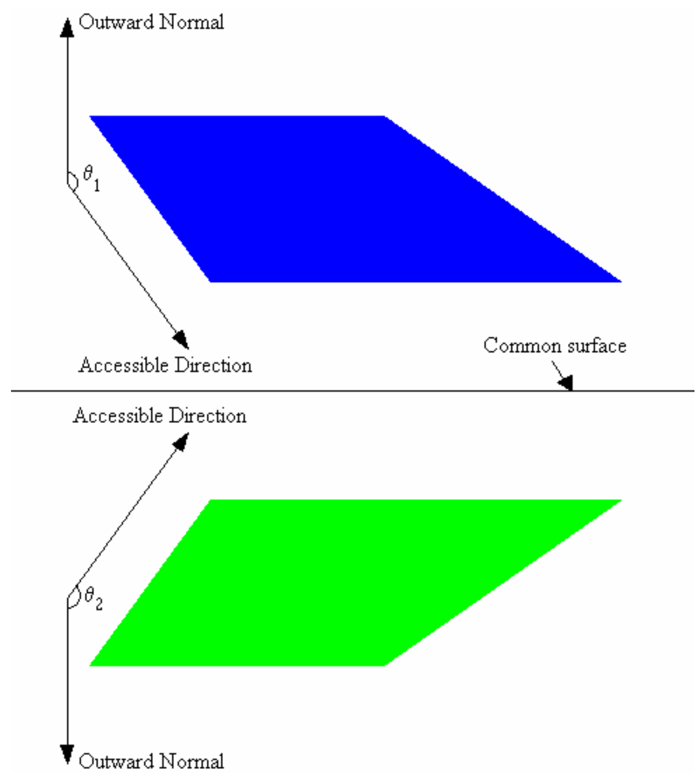


Figure 12: Condition for infeasibility of parting surface reduction

The angle between the outward normals and the accessible directions are then computed. These angles are referred as q_1 and q_2 in Figs. 11 and 12. If both pieces give results where the angle between the vectors is acute, as in Fig. 11, it can be observed that the pieces do not interfere with each other when they are removed from the flask. Hence, they can be patterned in the same flask without causing any damage to the mold during removal. Every such combination reduces the number of parting lines by one. However, if the pieces are as shown in Fig. 12, then the angles of the accessible direction with the outward normal are obtuse, and the two slices will interfere with each other when they are drawn out. Hence, they have to be molded in different flasks.

Solution Selection:

Once the parting lines are generated for each direction in the set of potential casting directions, the solutions have to be ranked based on some criteria. The criteria used for selecting the best orientation depends on the geometry of the part as well as the costs associated with the other processes related to casting, such as core making, pattern making; and the rate at which parts can be manufactured. In this research, only the geometric aspects are considered and the decision is based on a composite function of the number of cores required, the number of parting lines, the projected area of each piece, and the draw depth.

Number of cores required

The number of cores should be as small as possible. The performance measure (P_{core_i}) for the number of cores can be defined as the ratio of the number of cores required in a particular set-up (N_{core_i}) to the maximum number of cores required ($N_{core_{\max}}$). This will assign a value between zero and one for the measure. A smaller value of this performance measure indicates a better design.

$$P_{core_i} = \frac{N_{core_i}}{N_{core_{\max}}} \tag{5}$$

Presently, the number of cores is decided a priori by the designer. The coring provided could be insufficient in some orientations, requiring additional cores to obtain a valid solution. In such situations, this performance measure will have different values for different orientations. Another way of implementing this performance measure is to use the volume of cores required instead of the number of cores required. Then, the performance measure can be computed as:

$$P_{core_i} = \frac{V_{core_i}}{V_{core_{\max}}} \tag{6}$$

where:

V_{core_i} is the volume of cores required in orientation i

$V_{core_{\max}}$ is the maximum volume of cores required

Number of parting lines:

The number or parting lines should also be as small as possible. A performance measure (P_{part_i}) similar to the number of cores, varying from zero to one, is defined for the parting lines criterion, as the ratio of the number of parting lines in a particular orientation (N_{part_i})to the maximum number of parting lines over all orientations($N_{part_{\max}}$). A small value of this measure is desired.

$$P_{part_i} = \frac{N_{part_i}}{N_{part_{\max}}} \tag{7}$$

Projected area:

The projected area for each slice (A_{ij}) should be as large as possible. Also the range in areas for each slice should be as low as possible. To compute the performance measure, two parameters are computed for each orientation viz. the average projected area and the ratio of the smallest projected area to the largest projected area among the slices. The preferred values for these measures are a large number for the average projected area, and a ratio that is as close to 1.0 as possible.

$$P1_{area_i} = \frac{\sum_j A_{ij}}{n_i} \qquad P2_{area_i} = \frac{\min(A_{ij})}{\max(A_{ij})} \tag{8}$$

where n_i is the number of slices in orientation *i*.

Draw depth:

The draw depth for each slice ($draw\_depth_{ij}$) should neither be too large, as it will require a large amount of draft, nor too small, as that will lead to requiring a large number of slices. The performance measure for draw depth can be computed as the average of a ratio of the draw depth for each slice to the maximum draw depth among all the slices. The closer the value of this ratio is to 1.0, the better the design is.

$$P_{d\_depth} = \frac{\sum_j \frac{draw\_depth_{ij}}{\max(draw\_depth_{ij})}}{n_i} \tag{9}$$

where n_i is the number of slices in orientation i.

IMPLEMENTATION

The algorithm has been implemented using the ACIS geometric modeling kernel using the Scheme and C++ API interfaces under a Windows environment. The input file can be modeled in any solid modeling software and exported in Standard ACIS Text (.sat) format. This file is provided as input to the algorithm. The algorithm creates output files with different configurations for each of the different parting directions.

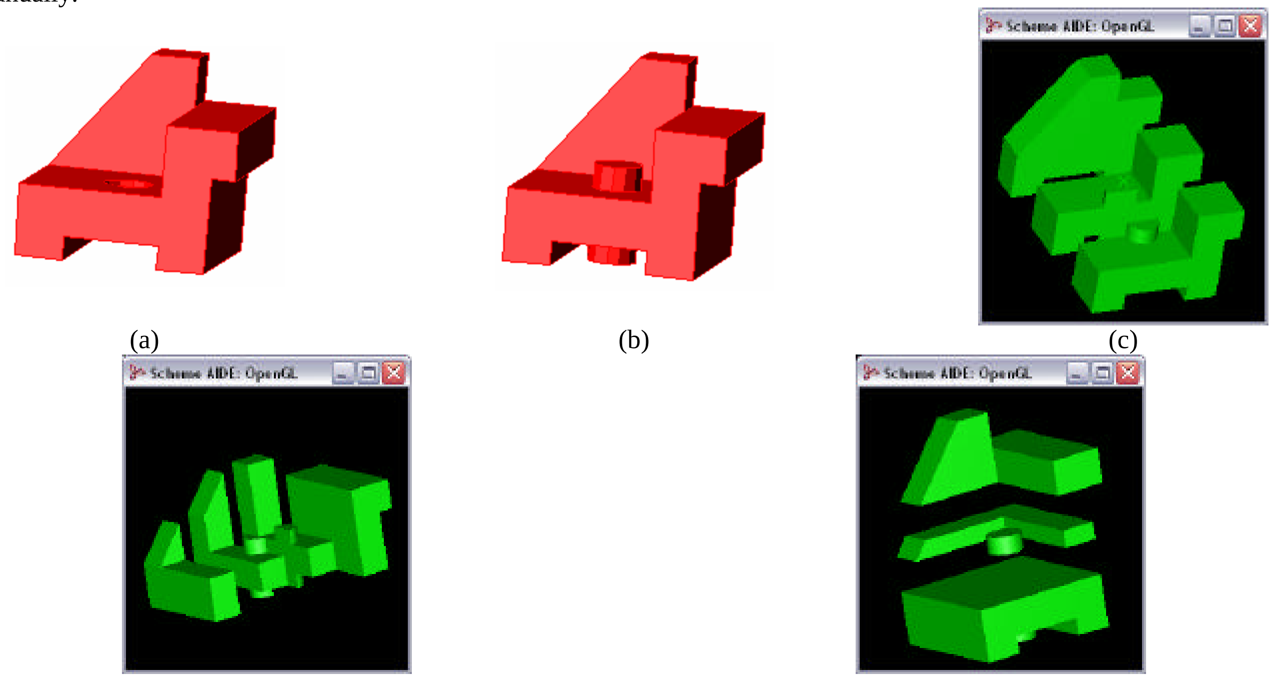
The output solutions are viewed using the Scheme ACIS Interface Driver Extension. The algorithm also returns a consolidated report of number of parting lines, projected area, draw depth, and the performance metrics defined earlier. For infeasible directions, none of these parameters are returned. For such directions, further manual intervention is required to identify the issues causing infeasibility. This report can be used to determine the composite performance metric for each orientation depending on the pre-decided weights, and the optimal orientation can be selected.

EXAMPLES

Figures 13 and 14 illustrate the solutions obtained by the algorithm for two different parts. The part in Fig. 13 has been used previously by Ravi and Srinivasan [14]. Figure 13(a) shows the final component, Fig. 13(b) shows the final pattern shape after adding the core and coreprint volumes, while Figs. 13(c-e) show the pattern slices and parting surfaces obtained for different casting directions. This example has been modeled in AutoCAD 2004 and exported in ACIS format for processing. The determination of core and coreprint volumes is done manually.

Figure 14(a) shows the final component. In Fig. 14(b), the machined holes have been filled, and in Fig. 14(c), the core and coreprint volumes have been added. Figures 14(d-f) show the pattern slices and parting surfaces for different casting directions. This example has been modeled in Solidworks 2003 and exported in ACIS format for processing. The determination of machining, core and coreprint volumes is done manually.

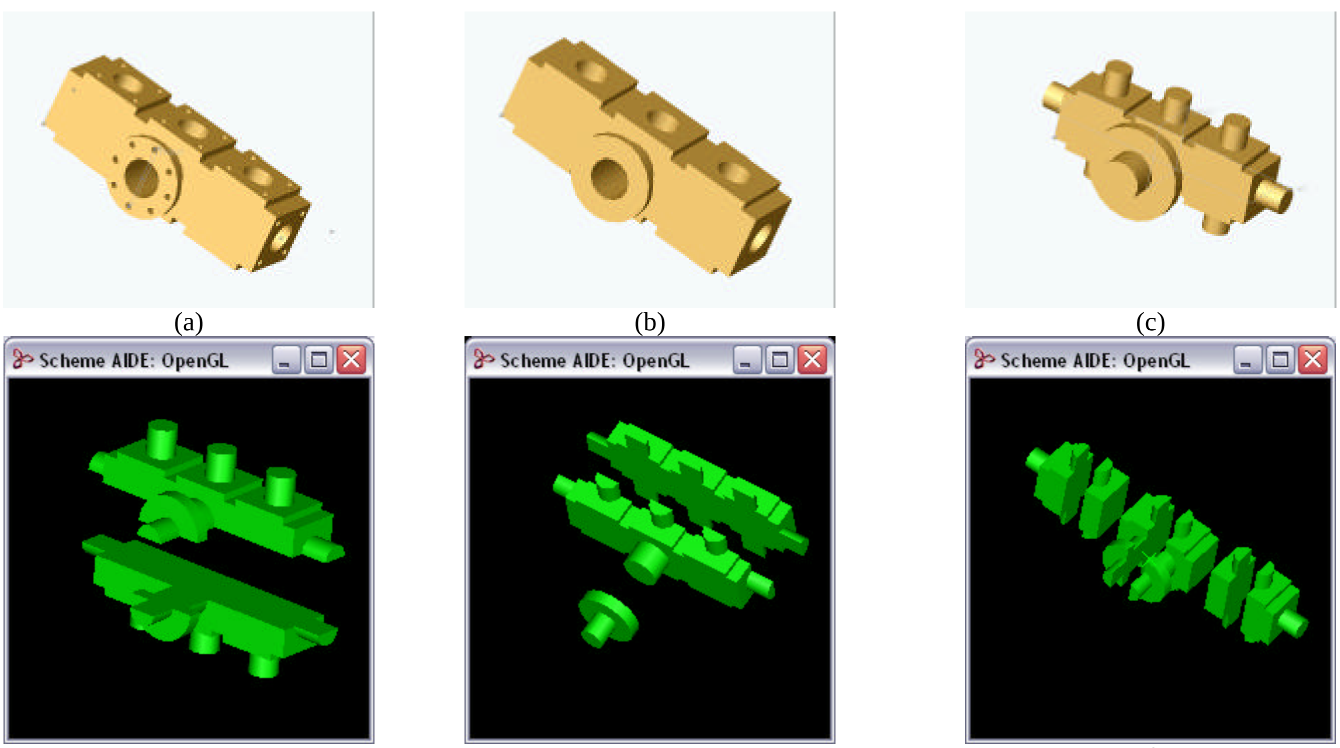
In both these figures, the results after parting are saved in ACIS format and displayed using the Scheme ACIS Interface Driver Extension. It can be observed in Fig 14 (f) that the second and third slices from either side can be molded in the same flask.



(a) (b) (c)

(d) (e)

Figure 12: Sample part from Ravi and Srinivasan [14] and parting lines along different orientations



(a) (b) (c)

(d) (e) (f)

Figure 13: Sample part and parting lines along different orientations

CONCLUSIONS

A method has been proposed and demonstrated to generate multiple parallel parting lines for sand casting. Some performance measures have also been defined to aid in selecting the best orientation. This algorithm will be beneficial to the design engineer in making a choice on casting plans. However the selection of coring requirements and assignment of weighting factors for the performance criteria are still decisions that have to be made by the designer. Research is underway to include feature recognition capabilities to the algorithm in order to aid the designer in the decision of coring requirements. Work needs to be undertaken to determine the optimal weighting factors to be used. Another open area that can be investigated is finding an efficient way to recompute the accessibility cones for the surfaces after a slice has been removed from the pattern.

REFERENCES

[1] Beeley, P., 2001, Foundry Technology, Butterworth-Heinemann, Oxford.

[2] Flinn, R.A., 1963, Fundamentals of Metal Casting, Addison-Wesley, Reading, MA.

[3] Ahn, H.K., de Berg, M., Bose, P., Cheng, S.W., Halperin, D., Matoušek, J., and Schwarzkopf, O., 2002, "Separating an object from its cast", Computer-Aided Design, 34, pp. 547-559.

[4] Ahn, H.K., Cheng, S.W., and Cheong, O., 1998, "Casting with Skewed Ejection Direction", Proc. 9th International Symposium on Algorithms and Computation, Springer-Verlag, pp. 139-148.

[5] Ahn, H.K., Cheng, S.W., and Cheong, O., 1999, "Casting with Skewed Ejection Direction Revisited", Proc. 11th Canadian Conference on Computational Geometry. http://www.cs.ubc.ca/conferences/CCCG/elec_proc/elecproc.html

[6] Fu, M.W., Nee, A.Y.C., and Fuh, J.Y.H., 2002, "The application of surface visibility and moldability to parting line generation", Computer-Aided Design, 34, pp. 469-480.

[7] Majhi, J., Gupta, P., and Janardan, R., 1999, "Computing a flattest, undercut-free parting line for a convex polyhedron, with application to mold design", Computational Geometry, 13, pp. 229-252.

[8] Chen, L.L., Chou, S.Y., and Woo, T.C., 1993, "Parting directions for mould and die design", Computer-Aided Design, 25, pp. 762-768.

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

[10] Weinstein, M., and Manoochehri, S., 1997, "Optimum parting line design of molded and cast parts for manufacturability", J. of Manufacturing Systems, 16, pp. 1-11.

[11] Priyadarshi, A.K., and Gupta S.K., 2004, "Geometric Algorithms for automated design of multi-piece permanent molds", Computer-Aided Design, 36, pp. 241-260.

[12] Dhaliwal, S., Gupta, S.K., Huang, J., and Kumar, M., 2001, "A Feature-Based Approach to Automated Design of Multi-Piece Sacrificial Molds", ASME J. of Computing and Information Science, 1, pp. 225-234

[13] Chen, Y., and Rosen, D.W., 2002, "A Region Based Method to Automated Design of Multi-Piece Molds with Application to Rapid Tooling", ASME J. of Computing and Information Science, 2, pp. 86-97.

[14] Ravi, B., and Srinivasan, M.N., 1990, "Decision criteria for Computer-Aided parting surface design", Computer-Aided Design, 22, pp. 11-18.

[15] Ganter, M.A., and Tuss, L.L., 1990, "Computer-Assisted Parting Line Development for Cast Pattern Production", Trans. American Foundry Society, pp. 795-800.

[16] Smith, C.T., and Lee, K., 1986, "Computer-Aided Pattern Design for Casting Process", Trans. American Foundry Society, pp. 21-28.

[17] Dhaliwal, S., Gupta, S.K., Huang, J. and Priyadarshi, A.K., 2003, Algorithms for Computing Global Accessibility Cones", ASME J. of Computing and Information Science, 3, pp. 200-209.

[18] Elber, G., 1994, "Accessibility in 5-axis milling environment", Computer-Aided Design, 23, pp. 796-802.

[19] Woo, T.C., 1994, "Visibility maps and spherical algorithms", Computer-Aided Design, 26, pp. 6-16.

[20] Suh, S.H., and Kang, J.K., 1995, "Process planning for multi-axis NC machining of free surfaces", International J. of Production Research, 33, pp. 2723-2738.

[21] Spitz, S.N., Spyridi, A.J., and Requicha, A.A.G., 1999, "Accessibility Analysis for Planning of Dimensional Inspection with Coordinate Measuring Machines", IEEE Trans. on Robotics and Automation, 15, pp. 714-727.

[22] Lim, C.P., and Menq, C.H., 1994, "CMM feature accessibility and path generation", International J. of Production Research, 32, pp. 597-618.

[23] Weisteinn, E.W., 1999, CRC concise encyclopedia of mathematics, CRC Press, Boca Raton.