

Geometric Algorithms for Rapidly Reconfigurable Mold Manufacturing of Free-form Objects

Aditya Kelkar, Rakesh Nagi, Bahattin Koc*

*Department of Industrial Engineering, 342 Bell Hall,
University at Buffalo (SUNY), Buffalo, NY 14260, USA*

Abstract

This paper presents geometric algorithms for developing a re-configurable tooling system for fabrication of freeform objects. The proposed method involves a mold block, with n faces, in which the mold cavity is formed by moving a set of discrete pins on each face of the block. The part surfaces are approximated in the mold cavity using the pins from the suitable mold block faces. The geometric algorithms detailed in this paper analyze the part and determine the face of mold block from which the part model is approximated best. Further, the algorithms detect possible interference between pins from different faces, and suitably alter the approximating face to alleviate interferences. By moving these pins in and out of the mold block, the shape of the mold cavity is reconfigured rapidly to suit the changes in part geometry. Since the proposed method approximates free-form objects with discrete pins, a surface-error calculation method is also developed to control the accuracy. Computer implementation and examples are also presented in this paper.

Key words: Re-configurable molds, discrete molds, free-form objects, NURBS surface discretization, error analysis.

PACS:

1 Introduction

In the past, production was ruled by economies of scale, and the best way to earn profits was to fully utilize resources, and produce in large volumes. However, during recent years, a lot of attention has been paid to flexibility, and

* To whom correspondence should be addressed. *Email address:* bkoc@buffalo.edu

adaptability of the manufacturing systems. The current market trend is moving towards mass customization, in which every product can be customized by the customers to adjust to their needs. This has led to the development of “Agile Manufacturing Systems” [1], which enables a high degree of mass customization to meet the uncertain and ever-changing demands of the customers.

The manufacturing processes involved in such a dynamic manufacturing environment must handle a variable product design pattern. A large class of industry such as injection molding, casting, stamping, and forging rely on the long and costly precursor of design and fabrication of dies and molds. However, a mold made for a particular product design can be used only for that design. Any design changes deliver the mold unsuitable for use, and a new mold must be made. Hence, such processes may not prove suitable for mass-customization. Layered manufacturing (LM) and similar technologies have developed very fast in the last decade, and have shown the capability to produce complex geometries. However, those processes are limited by the choice of materials and may not be feasible for mass-production. Hence, a process capable of quickly adapting to design changes, and at the same time, requiring less lead time, is much needed. In this paper, reconfigurable tooling system is proposed for mass customization.

The proposed method uses a set of reconfigurable pin matrices to produce reconfigurable molds for 3-dimensional free-form parts. By adjusting the pins on each mold face to approximate a given area of the part surface, a mold cavity for the desired part is obtained. Figure 1(a) shows an example part, a gear shift knob for a car. The original design is changed to meet user requirements as in Figure 1(b). Using the proposed tooling system, the mold cavity can be rapidly configured as shown in Figure 1(c) and 1(d) for the original and the changed model respectively. A conventional mold would require a totally new mold set, in order to incorporate the design changes. In the proposed re-configurable system, by moving the suitable pins inside or out of the mold cavity, the new shape of the knob can be easily configured in the mold cavity.

The proposed re-configurable tooling system can be used in several fabrication processes such as injection molding, transfer molding, and thermoforming. Reconfigurable injection molds can be designed to incorporate sprue, and cooling system. Ejection system will not be necessary in such a re-configurable system, as the molded part can be easily removed by retracting the pins. For transfer molding processes, the pins themselves can be used as rams to push the molten material into the cavity. The system can also be configured for thermoforming applications, where the mold can be formed using the pins.

This paper presents geometric algorithms to enable such rapidly re-configurable tooling system. Section 2 presents a review of previous research on re-configurable

systems. Section 3 describes the proposed process. Section 4 details the geometric algorithms for Re-configurable Mold Manufacturing. Section 5 presents examples used for testing the algorithm, while Section 6 discusses further improvements to the proposed system. Section 7 concludes the paper.

2 Literature Review

Several researchers have studied re-configurable tooling systems in forming, and part fixturing. Walczyk and Hardt [2, 3] developed discrete reconfigurable dies for forming sheet metal. The system consists of small, individually positioned pins, mounted in two rectangular matrices. The die in this system is an opposed die, which is formed by the pins on each of the matrices. These discrete pins are positioned to approximate the continuous surface of the matched dies with the help of numerical control. The pins may be either placed at intervals in the matrix, or densely packed. In both cases, the pins are positioned to approximate the desired shape of the die, and are then clamped with a tool. These methods are limited to create a single surface geometry and can be used only for formed parts with limited geometric complexity because of the nature of the forming process.

Papazian et al. [4, 5] used the same principle of reconfigurable dies to build a system for sheet stretch forming. It differs from the system in Hardt [2, 3] in that it uses stretch forming rather than opposed die forming. This system involves only a single matrix of pins, and a flexible rubber sheet. This system is used to shape large sheets of metal for applications in the aerospace industry. This method specifically addresses forming of large surfaces for aerospace applications, and is not designed to fabricate three dimensional parts.

Crawford and Klesspies [6] present a method for producing large curved surfaces using a "variable configuration vacuum forming mold". The mold consists of a number of discrete pins, and a flexible rubber sheet. The system is similar to Hardt [2, 3], but uses a different method for deformation of the sheet. Also, the system uses thermoplastic sheets rather than sheet metal. Again, this process is limited to only formed parts (i.e. cannot produce three dimensional free-form objects), and the interpolation sheet can cause waviness on the formed surface as the authors state in their paper.

Walczyk et al. [7] use computer-controlled re-configurable fixturing devices (RFDs) to fixture compliant parts. They use a matrix of pins fixed on a rigid platen. Each pin can be individually moved. The authors have presented their research on developing mechanisms for pin actuation using a combination of gas springs and pneumatic clamps. This system can be configured for any arbitrary part shape and can work as a universal fixturing device.

All the systems mentioned in the literature use a single re-configurable matrix of pins to form sheet metal, or a thermoplastic sheet, or to fixture a compliant part. These methods cannot be used to fabricate complex three dimensional (3D) free-form objects. In this paper, geometric algorithms are presented to enable 3D re-configurable tooling system for fabrication of 3D free-form objects.

3 Reconfigurable Mold Manufacturing (RMM)

The basic idea behind a reconfigurable tooling system is to be able to reconfigure the tooling setup to incorporate geometric changes in the design of a component to the tooling system [8]. With such a setup, it can be possible to reflect any changes in the product design directly from the CAD system to the customized part. In the proposed method, any change in the part design can be easily incorporated in the process by rapidly readjusting the location of the pins inside the mold block. By readjusting the suitable pins inside the mold cavity, the new part surface can be easily replicated. This would otherwise require a total change in the mold geometry, and a new mold would be required to be fabricated. By controlling the depth through which the pins are pushed into the mold block, the shape of the surface of the part can be achieved. Thus, changing the design will simply mean changing the positions of the pins inside the mold as shown in Figure 1. The surfaces of the mold cavity are formed by approximation of the model surfaces using a set of discrete pins. An example of a free-form surface, $S(u, v)$ is shown in Figure 2(a). The surface is approximated by a set of discrete pins by changing the height of each pin as shown in Figure 2(b).

The set of pins in Figure 2(b) is part of a reconfigurable mold block in which the mold cavity is formed. Each pin can be moved in and out of the mold block using suitable screw-based or similar mechanisms. Figure 3 shows an example of such system for controlling the position of the pins. By properly positioning the pins inside the mold block, the geometry of the part model can be approximated, and the mold cavity can be formed. The process generating a design cavity in a reconfigurable tooling system is summarized as follows (also shown in Figure 4): Given a free form object O with a set of parametric surfaces $S_i(u, v)$ as shown in Figure 4(a).

- (1) Determine the faces of the mold cavity from which each point of the surface grid will be best approximated by pins and assign face attributes to the surface points. Trace the contours of point sets on the surface grid having same face attributes. Eliminate any interference between pins, and the mold cavity. And re-assign interfering point sets to alternate mold faces (shown in Figure 4(b)).

- (2) As shown in Figure 4(c), the contours are projected onto the respective faces of the mold block and the pins on the mold faces lying inside the projected contours are determined. These pins which form the cavity are called active pins.
- (3) The exact location of the pins is calculated by projecting the active pins onto the surface model as shown in Figure 4(d).
- (4) Gaps are detected and extra pins are located to close the gaps between pins as shown in Figure 4(e).

According to the geometry of the part, the algorithms presented in the following sections generate the mold cavity of a given design model. Possible interferences between pins and the mold cavity are analyzed, and suitable point sets are re-assigned to alternate mold faces to provide an interference-free part cavity. The active pins which form the actual mold surfaces are determined, and positioned inside the mold block to generate the mold cavity from the CAD model of the part directly. The accuracy and finish of the molded part depends on the resolution of the pin size. Smaller the resolution, better is the fabricated part obtained. Since the presented method depends on surface approximation, a method of measuring the errors on the fabricated surfaces has been developed to control the accuracy. In the following sections, details of the geometric algorithms to generate the reconfigurable mold cavity directly from free-form CAD models are presented.

4 Geometric Algorithms for Reconfigurable Mold Manufacturing

The presented methods use a boundary representation of an object O . In this paper, NURBS parametric representation of the surfaces $S_i(u, v)$ are used to have better coverage of different surfaces as [9]:

$$S(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) Q_{ij} w_{ij}}{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) w_{ij}}, \quad (1)$$

where Q_{ij} denotes the control point net for surface $S(u, v)$ and w_{ij} denotes the weights of the control points. p and q are the degrees, and n and m are the orders of the surface along the parameters u and v . In order to determine what portion of the object should be assigned to each faces of the mold block, a set of points P_s on the parametric surfaces $S_i(u, v)$ are sampled. Any point P_s on a surface $S(u, v)$ is best approximated from the mold block face F_j that gives the least angle between block face normal vector \vec{N}_{f_j} and surface normal

at point P_s , \vec{N}_{P_s} . The normal vector \vec{N}_{P_s} of a surface point P_s is calculated as follows [9]:

$$\vec{N}_{P_s} = \frac{\frac{\partial S}{\partial u} \times \frac{\partial S}{\partial v}}{\left| \frac{\partial S}{\partial u} \times \frac{\partial S}{\partial v} \right|}, \quad (2)$$

where $\partial S/\partial u$ and $\partial S/\partial v$ are the first derivatives at a point P_s on a surface $S(u, v)$.

Each mold face F_j is referenced by its face number j ($j = 1, 2, \dots, n$). Superscript indices are used to represent the attribute in this paper. Points approximated from a particular face F_j are given that face number j as a classification attribute. This yields k point sets, $\mathcal{P}_{s_k}^j$ of face attribute j (superscript j refers to the attribute of point sets). To determine the contours of the region for the point sets $\mathcal{P}_{s_k}^j$ with the same face attribute j , a boundary tracing algorithm has been used (also shown in Figure 5). Instead of tracing the boundaries in Euclidean space, which would be computationally expensive, the boundary tracing is performed in parametric space, where only $u - v$ values need to be traced. After the boundaries of the regions are obtained, they are projected onto the respective mold block faces. This algorithm yields the contour of point sets that are to be approximated from a particular mold face. These contours are then project onto the respective mold faces. The pins lying inside these contours are the active pins A_n^j as shown in Figure 5, which actually form the mold cavity when suitably positioned inside the mold block j .

4.1 Interference detection and elimination

In the previous section, a set of points is assigned to a particular mold face, from which that set is best approximated, i.e. the face that will give the least errors in the mold surface. However, this does not take into consideration the accessibility of that point set from the assigned face. This can lead to two types of interferences:

- Cavity - pin intersection: If a pin approximating a surface point on the part surface passes through the mold cavity of the part, then the cavity-pin intersection occurs.
- Pin-pin Interference: This happens when pins from one face intersect pins from another face. There can be two types of such interferences:
 - Local interference: At the interfaces between two adjacent curves, boundary pins for the two sets from the respective faces may intersect each other. This is called local Interference.
 - Global Interference: If the non-boundary pins for two sets intersect each

other, then the intersection is called as global interference.

Interface checking process is not feasible at the face assignment stage itself, as each pin would have to be checked with a number of pins in order to detect interference. However, using the point set approach instead simplifies the problem, as the intersection of point sets can yield possible interferences with less computation.

4.1.1 Cavity-pin intersection detection and elimination

The cavity-pin interference detection and removal is formulated as an accessibility problem. If a point set is not accessible from a particular mold face, then it would lead to the pins passing through the mold cavity. Hence, such intersection must be detected and eliminated. To determine if a surface point $\mathcal{P}_{s_k}^j$ is visible or can be accessed from a viewing direction \vec{N}_{f_j} (normal vector of a mold face j), silhouette curves are used. A silhouette curve is defined as a curve of points on the surface, at which the surface normals are at a right angle with the viewing direction vector as:

$$\vec{N}_{p_s} \cdot \vec{N}_{f_j} = 0, \quad (3)$$

where \vec{N}_{f_j} denotes the viewing direction vector.

For a particular mold face j with normal vector \vec{N}_{f_j} , the silhouette curves $C_{s_m}^j$ ($m = 1, 2, \dots$) for the part when viewed from mold face j are determined. Let there be t sets of surface points assigned to mold face j . (best approximated from face j). Figure 6(a) shows the point sets $\mathcal{P}_{s_k}^j$, $\mathcal{P}_{s_{k+1}}^j$, $\mathcal{P}_{s_{k+2}}^j$ assigned to face j . A point set or a curve is said to lie below another point set or curve, if the former lies closer to the face of approximation than the latter. Similarly, a point set or curve is said to lie above another point set or curve, if the latter lies closer to the face of approximation than the former. Thus as shown in Figure 6(a), point set $\mathcal{P}_{s_{k+1}}^j$ lies above silhouette curve $C_{s_{m+4}}^j$, while point set $\mathcal{P}_{s_{k+2}}^j$ lies below silhouette curve $C_{s_{m+4}}^j$.

Viewing from the given direction \vec{N}_{f_j} of a mold face, if a set $\mathcal{P}_{s_k}^j$ ($k \in t$) lies below a silhouette curve which means visible from face j , and projection of the point set is completely inside the silhouette curve, then it is completely visible or completely accessible from the face j . Hence, this set of points can be approximated by pins from the face j . As shown in Figure 6(b), silhouette curve $C_{s_{m+4}}^j$ is visible from face j while silhouettes $C_{s_{m+1}}^j$ through $C_{s_{m+3}}^j$ are not. Point set $\mathcal{P}_{s_{k+2}}^j$ lies completely inside the silhouette curve $C_{s_{m+4}}^j$, and below it. Hence, the point set $\mathcal{P}_{s_{k+1}}^j$ is completely accessible from the face j .

If a set $\mathcal{P}_{s_k}^j$ lies completely inside a silhouette curve visible from face j , but above it when viewed along direction \vec{N}_{f_j} , then that set is completely obscured by the part surface lying below the silhouette curve, and is not visible from the mold face j . Hence, if this set of points is approximated by pins from face j , the pins will intersect the part surface lying below the silhouette curve, and would have to pass through the mold cavity to result in cavity-pin interference. Therefore this set cannot be approximated from the face j . In Figure 6(c), the point set $\mathcal{P}_{s_{k+1}}^j$ is assigned to face j , and it lies completely inside the silhouette curve $C_{s_{m+4}}^j$, which is visible from face j . However, point set $\mathcal{P}_{s_{k+1}}^j$ lies above curve $C_{s_{m+4}}^j$, and is obscured by the part surface below silhouette curve $C_{s_{m+4}}^j$. While $\mathcal{P}_{s_{k+1}}^j$ lies below the silhouette curve $C_{s_{m+2}}^j$, and completely inside it, the silhouette curve $C_{s_{m+2}}^j$ itself is obscured by the part surface below it. Therefore, $\mathcal{P}_{s_{k+1}}^j$ is not visible from the face j , and is not accessible from it. This set cannot be approximated from face j , and must be re-assigned to an alternate face.

If a set $\mathcal{P}_{s_k}^j$ lies partially inside the silhouette curve and above it when viewed along \vec{N}_{f_j} , then the part inside the silhouette curve cannot be accessed from the mold face j . The part lying outside the curve can be accessed from the face j , provided it is not obscured by another part of the surface. In Figure 6(d), silhouettes $C_{s_m}^j$ and $C_{s_{m+4}}^j$ are visible from the mold face j . Point set $\mathcal{P}_{s_k}^j$ is partially inside silhouette curve $C_{s_{m+4}}^j$, but completely inside $C_{s_m}^j$. The part of set $\mathcal{P}_{s_k}^j$ inside the silhouette curve $C_{s_{m+4}}^j$ is not visible from mold face j , and hence is not accessible from it. However, the part of $\mathcal{P}_{s_k}^j$ completely inside silhouette curve $C_{s_m}^j$, but not lying inside the silhouette curve $C_{s_{m+4}}^j$ is visible from the face j . This part of the set $\mathcal{P}_{s_k}^j$ can thus be accessed from the mold face j . Therefore, the point set $\mathcal{P}_{s_k}^j$ yields two different point sets, one accessible from the face j , and one not accessible from face j . When pin-cavity interference is found, the interfering point set is reassigned to another mold face by checking the second best candidate face. Interference checking and elimination continues until there's no interference and all the point set are assigned to their corresponding mold face. Due to the complex geometries of parts, it is possible that some of the point sets cannot be assigned to any mold face j without interferences.

Details of cavity-pin intersection determination and elimination are given in Algorithm 1. In Algorithm 1, the intersection of two point sets or a point set and a curve is interpreted as the intersection of their projected boundaries on a mold face (shown with the symbol \cap in the algorithm. For instance, the intersection of a point set and a silhouette curve means the intersection of the boundary curve of the point set, and the silhouette curve projected on a mold face.

Algorithm 1: Detection and elimination of cavity-pin intersection

INPUT: Free-form object O with set of surfaces $S_i(u, v)$. Point sets $\mathcal{P}_{s_k}^j$ ($k = 1, \dots, t$) assigned to mold face j .

OUTPUT: Set A_j of point sets accessible from mold face j , Set I_j of point sets inaccessible from mold face j .

START

Initialize $A_j = \{\}$ & $I_j = \{\}$

For viewing vector \vec{N}_{f_j} , determine silhouette curves $C_{s_m}^j$ ($m = 1, \dots$, number of silhouettes) using Equation (3).

$\forall \mathcal{P}_{s_k}^j :$

$\forall C_{s_m}^j :$

Calculate $\mathcal{P}_{s_k}^j \cap C_{s_m}^j$;

If ($\mathcal{P}_{s_k}^j \cap C_{s_m}^j \neq \text{NULL}$)

If ($\mathcal{P}_{s_k}^j \cap C_{s_m}^j = \mathcal{P}_{s_k}^j$ and $\mathcal{P}_{s_k}^j$ lies below $C_{s_m}^j$ and $C_{s_m}^j$ is visible from face j)

Then $A_j \leftarrow A_j \cup \mathcal{P}_{s_k}^j$;

/**Add $\mathcal{P}_{s_k}^j$ to set A_j of point sets accessible from face j .**/

If ($\mathcal{P}_{s_k}^j \cap C_{s_m}^j = \mathcal{P}_{s_k}^j$ and $\mathcal{P}_{s_k}^j$ lies above $C_{s_m}^j$ and $C_{s_m}^j$ is visible from face j)

Then $I_j \leftarrow I_j \cup \mathcal{P}_{s_k}^j$;

/**Add $\mathcal{P}_{s_k}^j$ to set I_j of point sets inaccessible from face j .**/

If ($\mathcal{P}_{s_k}^j \cap C_{s_m}^j \supset \mathcal{P}_{s_k}^j$ and $\mathcal{P}_{s_k}^j$ lies above $C_{s_m}^j$ and $\mathcal{P}_{s_k}^j$ lies below any $C_{s_n}^j$)

/**Point set partially visible from face j **/

Then $A_j \leftarrow A_j \cup (\mathcal{P}_{s_k}^j - \mathcal{P}_{s_k}^j \cap C_{s_m}^j)$;

$I_j \leftarrow I_j \cup (\mathcal{P}_{s_k}^j \cap C_{s_m}^j)$;

Find face i ($i \neq j$) for set I_j such that angle between \vec{N}_{P_s} and \vec{N}_{f_i} is minimum;

Reassign the set face attribute of I_j to i ;

END*4.1.2 Global pin-pin interference*

The pin-pin interference occurs when two pins from different mold faces intersect (or collide) each other. The surface point sets $\mathcal{P}_{s_k}^j$ will not cause any pin-pin interference when the point sets can be swept to infinity along their respective mold face normals without intersecting swept volume of any other point set. Since a point set with the same face attribute will not have any pin-pin interference problem, pin-pin interference will involve checking each point set with a face attribute j , with all other points sets having a different face attribute i ($i \neq j$). Consider a point set $\mathcal{P}_{s_k}^j$ ($k = 1, \dots, t$) for which possible pin-pin interference needs to be checked. Let j be the mold face from which point set $\mathcal{P}_{s_k}^j$ will be approximated. If the set $\mathcal{P}_{s_k}^j$ can be swept to infinity along the outward normal of the mold face j without intersecting any other point set $\mathcal{P}_{s_k}^i$ ($i \neq j$) with a different face attribute, then the set $\mathcal{P}_{s_k}^j$ does not cause any

pin-pin interference. However, if the sweeping of the point set $\mathcal{P}_{s_k}^j$ contains any other point set, then the point set $\mathcal{P}_{s_k}^j$ causes interference. Instead of sweeping the points, only the boundaries of the point sets are swept to check the interference. Figure 7 shows an example part which is divided into 11 point sets and assigned to each individual mold faces j . The sets are named according to the face attribute of the points. Arrow and plus and minus signs determines the direction of the swept. As shown in Figure 7, all the point sets except point set z_2^- can be swept to infinity without intersecting any other point set. Set z_2^- interferes with sets $x_1^-, y_1^-, x_1^+, y_1^+$. Hence, point set z_2^- must be re-assigned to other mold faces to eliminate possible pin-pin interference. When pin-pin interference is found, the interfering point set is reassigned to another mold face by checking the second best candidate face. Interference checking and elimination continues until there's no interference and all the point set are assigned to their corresponding mold face. Details of the pin-pin interference detection are given in Algorithm 2. In Algorithm 2, $sweep(\mathcal{P}_{s_k}^j, \vec{N}_{f_j})$ signifies the swept volume obtained when set $\mathcal{P}_{s_k}^j$ is swept along direction \vec{N}_{f_j} . Also, $size(\mathcal{P}_{s_k}^j)$ signifies the number of points included in point set $\mathcal{P}_{s_k}^j$. Intersection (\cap) of two point sets means the same as in Algorithm 1.

Algorithm 2: Pin-Pin Interference

INPUT: Free-form object O with set of surfaces $S_i(u, v)$. Point sets $\mathcal{P}_{s_k}^j$ assigned to face j .

OUTPUT: Point sets $\mathcal{P}_{s_k}^j$ ($k = 1, \dots, t$) without pin-pin interference.

START

$\forall \mathcal{P}_{s_k}^j :$

Sweep $\mathcal{P}_{s_k}^j$ to infinity along \vec{N}_{f_j} to get $sweep(\mathcal{P}_{s_k}^j, \vec{N}_{f_j})$;

$\forall \mathcal{P}_{s_{k'}}^i$ ($i \neq j$):

Calculate $sweep(\mathcal{P}_{s_{k'}}^i, \vec{N}_{f_i})$;

If ($sweep(\mathcal{P}_{s_k}^j, \vec{N}_{f_j}) \cap sweep(\mathcal{P}_{s_{k'}}^i, \vec{N}_{f_i}) \neq \text{NULL}$)

If ($size(\mathcal{P}_{s_k}^j) > size(\mathcal{P}_{s_{k'}}^i)$)

Then

Find face m ($m \neq i$) s.t. angle between \vec{N}_{P_s} and \vec{N}_{f_m} is minimum;
Reassign the face attribute of $\mathcal{P}_{s_{k'}}^i$ to face m ;

Else

Find face m ($m \neq j$) s.t. angle between \vec{N}_{P_s} and \vec{N}_{f_m} is minimum;
Reassign the face attribute of $\mathcal{P}_{s_k}^j$ to face m ;

END

4.1.3 Algorithm for finding active pins

Algorithms 1 and 2 determine possible interferences, and eliminate them. As mentioned in Section 4, the initial steps to find the point sets are to calculate the surface grid, finding the best face of approximation for the points on the surface grid, and then tracing the boundaries of the point sets, with different face attributes. Algorithm 3 details the procedure to find active pins. Algorithms 1 and 2 are used in this algorithm to eliminate possible interferences.

Algorithm 3: Finding active pins

INPUT: Free-form object O with set of surfaces $S_i(u, v)$ ($i = 0, \dots, s$); Mold block with faces F_j ($j = 1, \dots, n$) with pins R_p^j on face F_j ; normal vectors \vec{N}_{f_j} for each face F_j ; parametric interval Δu and Δv .

OUTPUT: Active pins A_n^j on each face j .

START

Initialize $i = 0, j = 0$;

$\forall S_i$:

Sample points P_s on each surface S_i , using Equation (1), and a user-defined parametric interval Δu , and Δv ;

Calculate surface normal \vec{N}_{P_s} at every point P_s using Equation (2);

$\forall P_s$:

$\forall F_j$:

Find face $j \in \text{Min}\{\vec{N}_{P_s} \cdot \vec{N}_{f_j}\}$;

Set face attribute of point P_s to j ;

Determine point sets $\mathcal{P}_{s_k}^j$, such that all points $P_s \in \mathcal{P}_{s_k}^j$ have same face attribute j ;

$\forall \mathcal{P}_{s_k}^j$:

Project boundary of point sets $\mathcal{P}_{s_k}^j$ on face j to obtain projected boundary B_k^j ;

Determine pins that lie inside the boundary B_k^j to obtain active pins A_n^j ;

Determine and reassign cavity-pin interference using Algorithm 1

Determine and reassign pin-pin interference using Algorithm 2.

END

4.2 Determining the positions of the active pins inside the mold block

After all the active pins A_n^j have been determined, and possible interferences have been eliminated, their positions inside the mold block must be set to obtain the geometry of the part model in the mold cavity. This means that the distance through which the pin must be moved inside, or out of the mold block must be determined. Any pin R_p^j on face j can be defined by a point at the center of the top surface, i.e. the surface that forms the mold cavity. The problem is thus to determine an intersection of a line perpendicular to face F_j at a point P_j , with a parametric surface $S(u, v)$. The intersection is calculated

in terms of parameters (u^*, v^*) of the point of intersection on the surface. To determine the parameters, the Jacobian Inversion method [10, 11] is used. Using the Newton Iteration, the parameters of the point are approximated until a convergence value is reached. The parameter values u^* and v^* at the final stage give the intersection point between the line and the surface $S(u, v)$. The Jacobian Inversion enables evaluation of the parameters of the pin point R_p^j on a surface $S(u, v)$, given two of its Cartesian coordinates. Let the pin R_p^j be on the face lying on the $X - Y$ plane. Thus its x and y coordinates are known and a parametric z -value on the surface needs to be calculated. The method estimates the parameter values for the pin from an initial guess point $P_s^o(u^o, v^o)$. The following equations have been used for the evaluation [10]:

$$\begin{aligned} x_u(u^o, v^o)\delta u + x_v(u^o, v^o)\delta v &= x^* - x(u^o, v^o), \\ y_u(u^o, v^o)\delta u + y_v(u^o, v^o)\delta v &= y^* - y(u^o, v^o), \end{aligned} \quad (4)$$

where x_u and x_v denote partial derivatives of x component of surface S with respect to u and v respectively. Similarly, y_u and y_v denote partial derivatives of y component of surface S with respect to u and v respectively. δu and δv denote small changes in u and v . u^o and v^o denote the parameter values of the initial guess point P_s^o . The projected distance on the face F_j between calculated point and the actual point is calculated using the distance formula:

$$d(P_s^o, A_n^j) = \sqrt{(x_a - x^o)^2 + (y_a - y^o)^2}. \quad (5)$$

where x_a and y_a are the x - and y -values of center points of active pin A_n^j . Algorithm 4 details the calculations to obtain the positions of the pins.

Algorithm 4: Determining position of active pins in the mold cavity

INPUT: Parametric surface $S(u, v)$; Active pins A_n^j on block face F_j ; convergence value for numerical solution *conv*.

OUTPUT: z -values of each active pin A_n^j inside the mold cavity.

START

Initialize $P_s^o(u^o, v^o)$ as initial guess point;

$\forall A_n^j$:

 Evaluate x^o, y^o using Equation (1);

 Calculate $d(P_s^o, A_n^j)$ using Equation (5);

While($d(P_s^o, A_n^j) < conv$)

 Solve Equation(4) for δu and δv ;

$u^* = u^o + \delta u$;

$v^* = v^o + \delta v$;

 evaluate x^*, y^* using u^* and v^* in Equation (1);

Calculate $d(P_s^*, A_n^j)$ using P_s^* in Equation (5);
 $u_a = u^*$;
 $v_a = v^*$;
 Evaluate z_n^* using u_a and v_a in Equation (1);
EndWhile END

4.3 Closing the gaps in the mold

The method presented in the preceding sections projects points lying inside a contour on the mold face onto the surface model. This point is taken as a point on the top face of its corresponding pin, to generate the pin. Two adjacent points on the surface may be approximated from two different mold faces. Due to this, gaps may be introduced in the mold. To close the gaps, extra pins are used. The contours on each face are expanded on all sides to create overlaps between the point sets on the surface model. These extra pins are then positioned inside the mold block to close the gaps without any interference. Consider a surface $S(u, v)$, partitioned into m point sets, $\mathcal{P}_{s_k}^j$, ($k = 1, \dots, m$). Then, the Algorithm 3 ensures that each point on the surface $S(u, v)$ is assigned a face attribute, and hence is contained in some set $\mathcal{P}_{s_k}^j$.

$$\begin{aligned}
 \mathcal{P}_{s_1}^j \cup \mathcal{P}_{s_2}^j \cup \dots \cup \mathcal{P}_{s_m}^j &= S \\
 \mathcal{P}_{s_1}^j \cap \mathcal{P}_{s_2}^j \cap \dots \cap \mathcal{P}_{s_m}^j &= \phi
 \end{aligned} \tag{6}$$

The sets $\mathcal{P}_{s_k}^j$ when projected onto the faces F_j and expanded to yield the expanded boundary. Then, these expanded point sets are projected onto surface S to yield sets $\mathcal{P}_{s_k}^{j+}$,

$$\begin{aligned}
 \mathcal{P}_{s_1}^{j+} \cup \mathcal{P}_{s_2}^{j+} \cup \dots \cup \mathcal{P}_{s_m}^{j+} &\subset S \\
 \mathcal{P}_{s_1}^{j+} \cap \mathcal{P}_{s_2}^{j+} \cap \dots \cap \mathcal{P}_{s_m}^{j+} &\neq \phi
 \end{aligned} \tag{7}$$

Thus we can say that the set $\mathcal{P}_{s_a}^{j+} \cap \mathcal{P}_{s_b}^{j+}$ ($a \neq b$) is being approximated by pins from two faces. This region is the overlap region between the two sets, and hence is the region where the pins will interfere. By removing this interference, we can ensure that the mold does not have any gaps, as well as no pins interfere with each other. Thus, inducing the overlaps can eliminate gaps in the mold. Figure 8 shows the process of closing the gaps.

4.4 Estimation of errors between the actual surface and the mold surface

Since the above algorithms approximate the parametric surfaces with discrete surfaces, there will be an error introduced on the mold surface. To control the accuracy of the fabricated parts, the surface errors need to be calculated. Similar error calculation methods [12, 13, 14] have been used in layered manufacturing to calculate layer thickness for adaptive slicing. Using the same principle, a method has been developed to estimate errors in the reconfigurable molds. For a given width w of the pins and using the methods presented in previous sections, the height difference h between the pins can be calculated as shown in Figure 9. The method uses the corners of the pins for calculations. The curve between the two points is approximated as a circle with radius equal to radius of curvature of surface along plane containing the normal at any point P_s as shown in Figure 9. There are two possible cases that may be found, concave surfaces, and convex surfaces. Figure 9 shows pins approximating a concave surface. There may be a case where the pins approximate a convex surface. Formulae for both cases have been derived as follows:

Case (i): Concave surface

$$Error \quad \epsilon_{concave} = R \pm \sqrt{R^2 - h^2 - 2Rh \sin(\alpha)} \quad (8)$$

Case (ii): Convex surface

$$Error \quad \epsilon_{convex} = -R \pm \sqrt{R^2 - h^2 + 2Rh \sin(\alpha)} \quad (9)$$

Where R is the radius of curvature measured at point P_s , h is the height difference between the active pins A_n^j and A_{n+1}^j and w is the width of the pins. By calculating the surface errors, the accuracy of the fabricated part can be controlled. To achieve accurate parts, smaller pins should be used. But smaller the size of pins, higher is the number of pins required. Positioning a large number of pins can be time-consuming. Hence, a compromise must be reached between the number of pins, and desired accuracy.

5 Implementation and Examples

The developed algorithms have been implemented in C using OpenGL for visualization. The OpenNURBS toolkit is used to generate the output in standard .3dm file format to view and modify in Rhinoceros 3D NURBS modeling tool. The initial NURBS based models are also generated using Rhinoceros 3D software. The surface information is stored in a text file, and is used in the

implementation to generate the NURBS model. In these examples, the mold block is taken to be a cube, with 6 sides.

A sphere with a radius of 7.0 units is used as the first example. The actual NURBS model of the example part is shown in Figure 10(a). Discrete pins with square cross sections (size of 0.5 units) are used for generating the mold cavity. After determining the active pins, the cavity of the example part is configured as shown in Figure 10(b). Only five sides of the mold have been shown for clarity. The errors are calculated using the method presented in Section 4.4 and maximum errors are found to be 0.007756 units. To increase the accuracy, smaller pins can be used.

Figure 11(a) shows the NURBS model of Example part II, a vase. is taken as another example. The vase is 26.5 units in height, and maximum diameter is 12.5 units. The mold generated for a square pin size of 0.5 units in cross-section is shown in Figure 11(b). Maximum error obtained with 0.5 units pin size was 0.150749 units. Figure 11(c) shows the vase obtained from the mold in Figure 11(b). Figures 11(d) and 11(e) show the mold and vase generated, for a pin cross section of 0.3 units. The maximum error for a pin size of 0.3 units was 0.095188 units. As shown in Figure 11, smaller the resolution better is the finish obtained from the mold, as observed from the errors obtained in each model.

To illustrate the interference removal algorithms, a human head model is taken as an example. The bounding box of the head is 15 by 11 by 17 units. Figure 12 shows the steps that were required in generating the mold for the model. Figure 12(a) shows the model divided into the initial point sets. Twelve point sets were derived from the model at this stage, based on the methodology described in Section 4. Each of the point sets is assigned to a mold face, as shown in the Figure 12(a). After the silhouette curve analysis, as described in Algorithm 1, it was found out that point sets 4 and 5 are not visible from the mold face with the direction in z^+ , and hence, cannot be approximated from that face. Hence, point sets 4 and 5 were re-assigned to face with the direction in x^+ of the mold block (second best candidate). This resulted in the number of point sets being reduced to 10, as shown in Figure 12(b). Figure 12(c) shows the boundaries of the point sets being swept along the normal of their assigned mold face. Using Algorithm 2, interference between point set 9 and point sets 2,3,6 and 8 is detected. The decision at this point would be to decide if set 9 is to be re-assigned to an alternative mold face, or to re-assign the interfering point sets to alternate faces. This decision is based on the size of each point set. If set 9 is re-assigned to other faces, then effectively, a large number of points would be approximated from an alternate mold face, rather than the first choice face. This would lead to an increase in the average error in the model. On the other hand, if point sets 2,3,6 and 8 were assigned to an alternate face, i.e. face with the direction in x^+ , then a lesser number of

point would be re-assigned, and hence, the average error will not increase as much as in the first case. Hence, in this implementation, the decision was made based on average error, and the point sets 2,3,6,8 were assigned to mold face with the direction in $x+$. Figure 12(d) shows the final assignment. Based on these point sets, the mold for the head was generated. Figure 13(a) shows the NURBS model of Example part III. The model generate from the mold cavity is shown in Figure 13(b). Figure 13(c) shows a cut section of the mold with a pin size of 0.3 units.

To show the accuracy of the surface error calculation method presented in Section 4.5, a free-form surface $S(u, v)$ is designed as shown in Figure 14(a). To show the error distribution, a surface curve $C_s(u^*, v)$ are calculated as shown Figure 14(b) Using pins sizes 0.4 and 0.8 units, the surface is approximated as shown in Figure 14(c) and 14(d) respectively. The error analysis results are shown in Figure 14(e) and 14(f) for grid sizes 0.4 and 0.8 units. As shown in the Figure 14(e) and 14(f), the error depends on the absolute value of curvature of the surface. As the absolute value of curvature increases, the error in the mold surface increases. The maximum error using a grid size of 0.4 units was obtained as 0.100179 units. The least error was 0.000096 units, in the region where the surface was almost flat. With a grid size of 0.8 units, the maximum error obtained was 0.147232 units, while the least error 0.000195 units.

To test the feasibility of the mechanism of reconfigurable tooling system, we have developed a prototype mechanism as shown in Figure 15. The prototype system includes a set of reconfigurable pins, a CNC controller, a positioning mechanism for locating the pins as shown in Figure 15.

6 Discussion

As presented in the earlier sections, the proposed reconfigurable mold manufacturing process consists of positioning each active pin on each of the mold faces in its desired position to approximate the shape of the part to be molded. As seen from the analysis of surface errors, smaller the pins, better is the accuracy of the produced mold. Also, by using pins with inclined top surfaces to approximate the surface better, the accuracy of the mold can be further improved.

A better approximation would be if the top surfaces of the pins are oriented along a plane normal to the surface normal of the center point of the pin. The four corner of the pin's top surface would then be found out by using Coplanarity condition. In other words, the four corner points and the center point should lie on the same plane, and the plane should be perpendicular to the normal vector at the center point. The center point of the pin, R_p is

known completely, and the point lies on the surface $S_i(u, v)$. The aim is to find the top surface T_p of the pin R_p such that the normal vector of this plane T_p is parallel to the normal vector \vec{N}_{p_s} at the point P_s . This means that the four corner points need to be calculated such that they are coplanar. The coplanarity condition states that the dot product of the vector \vec{N}_j between the two coplanar points P_{c_t} ($t = 1, 2, 3, 4$) and $P_j(u, v)$, and the normal $\vec{\rho}$ should be perpendicular to each other, i.e.:

$$\vec{\rho} \cdot \vec{N}_j = 0. \quad (10)$$

For a pin R_j lying on $X - Y$ plane, since the size of the pin R_j is known, the X and Y coordinates of the four corners can be easily found out as:

$$\begin{aligned} x_{p_t} &= x_j(u, v) \pm w/2, \\ y_{p_t} &= y_j(u, v) \pm w/2, \end{aligned} \quad (11)$$

where w is the width of the pin.

Using Equations (10) and (11), the Z coordinate of the corner point P_t can be found. Thus, four coplanar point defining the top surface T_j of the pin Φ_j can be found out. Figure 16(a) shows a surface, approximated using pins with flat top surfaces, while Figure 16(b) shows the same surface approximation using inclined top surfaces. As can be seen from the Figure 16, the inclined surfaces give a better approximation of the surface. To be able to use pins with inclined top surfaces, a new mechanism system needs to be designed to pick and place pins with pre-cut top surfaces in varying degrees. This system would be much more complex to control.

Another application of the presented algorithms is for tool path generation in 3-axis machining. Since the pins from each of the faces can be moved only along the normal of that particular face, and movement along any other direction is not possible, this is analogous to 3-axis machining in which the tool movement is restricted only along the 3 coordinate axes. By decomposing the surface into regions accessible from each of the 3 coordinate directions, the machinability of the surface can be analyzed as presented in earlier sections.

Using pins from 2 different faces to approximate a surface is analogous to using a change in setup during machining. If a part surface is not completely accessible in one particular setup, then, with a change in setup, the inaccessible region can be made accessible. All surfaces accessible from one direction can be machined in one setup. Thus, the regions of the surface can be machined one by one, using a different setup for each machining direction.

The complexity of the parts that can be manufactured using the presented process depends on the accessibility of the designed part with a set of pins. Very complex parts with internal features may not be suitable for the proposed process due to accessibility issues.

7 Conclusions

This paper presents geometric algorithms for rapidly reconfigurable mold manufacturing of free-form objects. The process presents a unique method of producing highly customized products. By reconfiguring the pins of the mold, the mold cavity can be rapidly changed to accommodate geometric changes in design. This can certainly prove time-saving for products that have a high rate of design change. To generate a 3D cavity directly from the CAD model of the designed part, geometric algorithms have been presented in this paper. By analyzing the part surfaces to be approximated, required active pins from the suitable mold block faces have been calculated to form the mold cavity. Possible interference problems have been detected and eliminated using the developed algorithms. Since the methodology is based on surface approximation, an error estimation method for measuring the accuracy between the actual model and the generated mold has been presented. The developed geometric algorithm enables rapidly reconfigurable tooling technology which can greatly reduce lead time. These processes can prove to be useful for mass customization of products. It is expected that the algorithms will find a wide range of applications in reconfigurable fixturing and forming processes, as well as in 3-axis machining.

References

- [1] Sanchez, L.M. and Nagi, R., "A review of agile manufacturing systems," *International Journal of Production Research*, 2001, 39, 3561-3600.
- [2] Walczyk, D.F. and Hardt, D.E., "A Comparison of Rapid Fabrication Methods for Sheet Metal Forming Dies," *ASME Journal of Manufacturing Science and Engineering*, 1999, 121(1), 214-224.
- [3] Walczyk, D.F. and Hardt, D.E., "Design and Analysis of Reconfigurable Discrete Dies for Sheet Metal Forming," *Journal of Manufacturing Systems*, 1998, 17(6), 436-454.
- [4] Nardiello, J., Christ, R., and Papazian, J.M., Block Set Form Die Assembly, USA Patent 6,053,026, April 2000.

- [5] Papazian, J.M., Anagnostou, E.L., Christ, R.J., Hoitsma, D., Orivile P, Schwarz, R.C., Spitzer, K., and Barkley, C., "Tooling for Rapid Sheet metal parts production," *6th Joint FAA/DoD/NASA conference on Aging Aircraft*, September 2002.
- [6] Klesspies, H.S. and Crawford, R.H., "Vacuum Forming of compound curved surfaces with a variable geometry mold," *Journal of Manufacturing Systems*, 17(5), 327-337.
- [7] Walczyk, D.F. and Longtin, R.S., "Fixturing of compliant parts using a matrix of reconfigurable parts," *ASME Journal of Manufacturing Science and Engineering*, 2000, 122(4), 766-772.
- [8] Kelkar, A., Koc, B., and Nagi, R., "Rapidly Re-configurable Mold Manufacturing," *2003 ASME Design and Manufacturing Conference*, Chicago, Illinois, September 2 - 6, 2003
- [9] Piegl, L. and Tiller, W., *The NURBS Book*, Springer Verlag, Germany, 1995.
- [10] Choi, B.K. and Jerard, R., *Sculptured Surface Machining - Theory and Applications*, Kluwer Academic Publishers, Dordrecht, Netherlands, 1998.
- [11] Faux, I.D. and Pratt, M.J., *Computational Geometry for Design and Manufacture*, Chichester, England: Horwood; New York: Halsted Press, 1981.
- [12] Kulkarni, P. and Dutta, D., "An Accurate Slicing Procedure for Layered Manufacturing," *Computer Aided Design*, 1996, 28(9), 683-697.
- [13] Koc, B. and Lee, Y.S., "Adaptive ruled layers approximation of STL models and multi-axis machining applications for rapid prototyping," *Journal of Manufacturing Systems*, 2002, 21(3), 153-166.
- [14] Koc, B., Ma, Y., and Lee, Y.S., "Smoothing STL files by Max-Fit biarc curves for rapid prototyping," *Rapid Prototyping Journal*, 2000, 6(3), 86-204.
- [15] Pavlidis, T., *Algorithms for Graphics and Image Processing*, Computer Science Press, Rockville, Maryland, 1982.
- [16] Rourke, J., *Computational Geometry in C*, Cambridge University Press, 1998.