Computer-Aided Design and Applications

http://www.cadanda.com

Machining Free-Form Surface Cavities Using a Combination of Traditional and Non-Traditional Multi-Axis Machining Methods

C. G. Jensen[1], W. E. Red[2] and C. Ernst[3]

[1]Brigham Young University, USA, cjensen@byu.edu
[2]Brigham Young University, USA, ered@et.byu.edu
[3]Elemental Technologies, Inc, USA

ABSTRACT

Ball-end mill (3-axis) machining is a proven, yet inefficient, material removal process for machining surfaces defined within a cavity. Curvature matched (5-axis) machining (CM^2) is a proven efficient material removal process for large free-form surfaces, but has primarily been applied to open surfaces. This paper presents a robust algorithm that blends CM^2 machining with traditional three-axis ball-end machining to accurately mill free-form surfaces enclosed by cavity walls. Surfaces once inaccessible to CM^2 methods can now be effectively machined, while inaccessible regions are prepared for automatic ball-end mill machining. Results of time saving and the percentage of the surface machined using each method are presented.

Keywords: five-axis machining, global gouging, tool path generation.
DOI: 10.3722/cadaps.2008.241-253

1. INTRODUCTION

Free-form surfaces, because of their aesthetic appeal and superior performance, commonly find their way into the form and function of automobiles, airplanes, appliances, etc. From a manufacturing perspective, this is the virtual or digital design phase of engineering. Creating the manufacturing-tooling (molds and dies) that produce these designs is a much more daunting task if one attempts to use methods more efficient than the traditional 3-axis, ten passes per mm ball-end mill. Curvature Matched Machining (CM^2) is a 5-axis machining technique that uses flat and filleted-end mills to machine free-form surfaces efficiently and accurately [13-14],[21]. Others have referred to this method as curvature-catering [24]. The basic premise of these methods is to match the curvature of the swept silhouette of the cutting tool to the curvature of the surface orthogonal to the direction of tool travel. This matching is done at discrete steps along a tool's trajectory, incorporating both tool tilt and tool inclination angles.

CM^2 has demonstrated numerous benefits, e.g., reducing scallop height and machining time, and producing an almost grind-free surface. In the current release of Dassault Systemes CATIA V5, numerical part programmers can develop tool paths based on a CM^2 like algorithm. It is assumed that other CAD/CAM vendors will likewise follow suit and provide curvature-catering algorithms; however, to date, these methods have not seen widespread adoption. The authors believe this is due in part to the limited functionality that surrounds this new path planning method, and to the limited tool orientation control methods found in modern controllers. For example, free-form surfaces not enclosed by vertical walls; see Fig. 1(a), or surfaces enclosed by other draft or parting surfaces; see Fig. 1(b), would typically be machined using an end milling approach, as opposed to flank milling. These have heretofore proven themselves as excellent candidates for CM^2 methods. An extensive benchmark has been completed that compares curvature milling methods on open surfaces [1]. Results showed that for large open automotive surfaces, curvature milling methods were significantly more (2-3 times) efficient than other 5-axis flat end mill techniques.

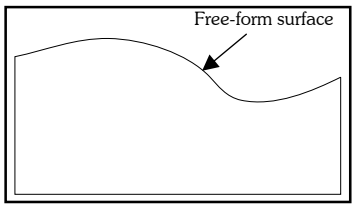

Fig. 1(a): Free-form surface no enclosed.

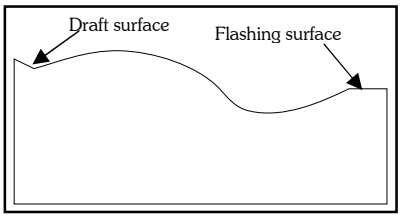

Fig. 1(b): Free-form surface enclosed by draft or flashing surfaces.

Many mold and die applications have their free-form surface cavities enclosed by vertical or near vertical walls (meaning a 5-7 degree draft angles) similar to Fig. 2. This paper focuses on the development and generalization of CM^2 methods that combine traditional and non-traditional machining algorithms to automate the path planning of free-form surfaces within vertical wall cavities (VWC). The CM^2 algorithms are used to determine and path plan accessible regions while detecting and correcting local and global gouging within the cavity. Next, our algorithms use a ball-end mill to automatically machine the inaccessible regions and flank mill the VWC walls leaving a fillet blend between the walls and the free-form surface. This VWC problem and its solution have yet to be addressed by the community of five-axis curvature matched machining researchers.

2. BACKGROUND

The topic of gouging has commonly been divided into two categories – local and global gouging. For a filleted-end mill, local gouging occurs when the bottom toroidal portion of the cutter (see Fig. 3) removes material below an allowable profile tolerance specified on the design surface; see Fig. 4. Global gouging occurs when any other part of the cutter, tool holder or spindle interferes with any surface region; see Fig. 4.

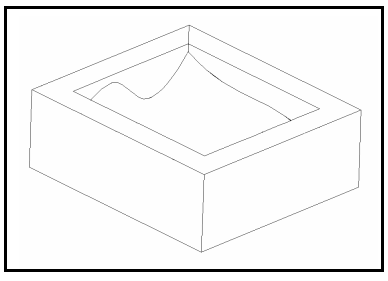

Fig. 2: Free-form surface in a cavity.

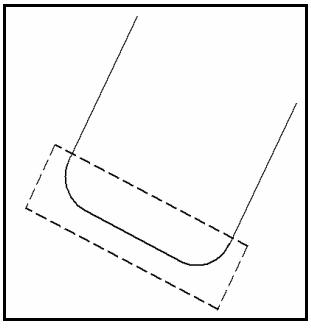

Fig. 3: Bottom portion of tool.

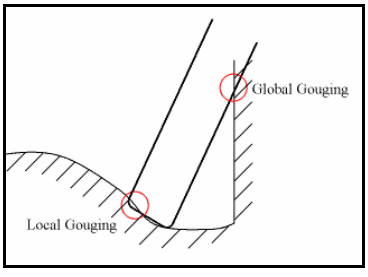

Fig. 4: Local and global gouging.

Solutions to 3-axis gouging have been widely researched and developed [4],[6-8],[10],[17],[23],[26]. In 3-axis machining, the tool orientation is fixed. Elber et al. and Hornung et al. approaches use hidden surface-removal techniques from a direction collinear with the tool axis [6],[10]. Many CAM systems use Z-buffering methods to verify and correct global gouging within 3-axis tool paths [7]. Tseng and Joshi use subdivision of the control polygon to avoid tool interference of Bézier surfaces [23]. This method employs a subdivided control polygon to approximate the surface. If the control polygon is interference-free, then the surface it approximates is interference-free as well.

While local and global three-axis gouging are well understood both pose barriers to accurate, efficient 5-axis cavity milling of free-form surfaces. Solutions to 5-axis gouging have historically followed three paradigms. The first is a "detect and correct" approach [2-3],[15],[18],[20],[25]. In this approach the gouging is detected and subsequently eliminated by reorienting the cutting tool. Many researchers as well as commercial CAD developers have developed efficient methods for sensing global gouging. One method [15] uses a two-phase approach to finding global tool interference. In the first stage the convex hull property of NURBS is exploited to quickly find a conservative tool orientation. If interference is detected, a second phase is implemented which uses exact surface normals and surface curvatures. Another method [20] tessellates the surface being machined. A bounding box is then calculated around

the cutter. Distance calculations that fall inside the bounding box are made between each triangle of the tessellated surface and the tool axis. If any distance is less than the tool radius, gouging occurs and is corrected.
The "detect and correct" approach works best for ball-end cutters because they are always constrained to be tangent to the machined surface. Hence at a single point on the surface, the cutter normally has a full range of orientations that can be considered in an attempt to eliminate global gouging. This approach does not work as well for a flat or filleted-end cutter. These cutters are typically oriented very precisely, based on surface curvature, and have limited or restricted orientations in which to eliminate local and global gouging. Attempts to reorient flat and filleted cutters to avoid global gouging can result in local gouging. This is a common problem yet to be solved by the major CAD/CAM vendors.
The second approach is to create accessible boundaries [5]. This approach has captured the attention of far fewer researchers. Elber and Cohen create boundary regions of check surfaces to identify regions of the free-form surface that can be machined without gouging. In their method, vectors are defined and computed between a discrete point on the drive (to be machined) surface and all points of the check surface. A boundary point is determined when any computed vector is normal to two vectors that span the plane orthogonal to the tool axis at this discrete point and are subsequently perpendicular to a normal vector of the check surface. The algorithm is versatile, working for any type of tool orientation; however, computational efficiency is a major concern.
A third approach to the local gouging problem is by Yoon et al. [27]. They define local machinability of smooth surfaces in terms of second order directional derivative cases. Their geometric method of optimal tool positioning is based on principal curvatures and directions of both the part surface and the cutter. Their method guarantees gouge-free tool positioning in a differential area around the cutter contact point. This method requires further development to guarantee the absence of local gouging with the entire bottom portion of the cutter. The previous machining strip evaluation work of Lee and Ji [16] was also incorporated in Yoon's work. A natural extension to Yoon's work would include surface profile tolerances so that tool positions below the design surface, but above the offset profile tolerance surface, would not be flagged as a gouge but would be an acceptable tool position and orientation.

3. METHOD

As describe in the introduction, our research plans CM^2 tool paths across the largest accessible area of a NURBS surface set within a VWC. The regions that are inaccessible to this five-axis flat or filleted-end mill machining method will automatically be planned using a ball-end cutter. Because of the vertical or near vertical (5-7 degree draft angle) conditions of the containment walls and the typical existence of a prescribed fillet radius that blends the cavity walls to the free-form surface bottom, our first step is to decompose (remove) the inaccessible portion of the perimeter from the free-form surface.
The next step involves generating the tool-axis vector and the cutter-location (CL) points for CM^2 machining of this trimmed surface. This step contains a number of sub-tasks. The initial tool-axis vector is subjected to a local gouge detection test and is corrected, if necessary. A subsequent global gouge detection test is then performed. Where no global gouging occurs, a CL point is calculated and output to a file for later ordering and sequencing into an efficient curvature matched tool path. Where global gouging does occur, a cutter contact (CC) point is stored for later use in creating inaccessible surface regions. The final stage is the ordering and efficient machining of ball-end mill tool paths over the inaccessible and border regions of the free-form surface. During this stage, a final flank milling past of the ball-end mill is taken around the perimeter of the pocket to both accurately mill the planar walls and leave the specified fillet. Both the CM^2 and ball-end mill tools paths are combined (via a tool change operation) into a single path plan capable of milling free-form surface cavities.
The main advantage of this method is that it extends the benefits of CM^2 five-axis machining formerly inaccessible regions. A surface previously non-machinable in 5-axis because of vertical walls or prescribed fillet blends adjacent to the cavity walls, us now amenable to a new machining algorithm. This method automatically path plans five-axis CM^2 trajectories on accessible regions and then path plans three-axis ball-end mill trajectories on inaccessible regions.

3.1 Surface Trimming

As discussed above, the first step in generating five-axis CM^2 NURBS cavity machining is to trim off the portion of the drive surface that is within a tool radius (flat-end mill) of the cavity walls; see Fig. 5. Generating a five-axis CM^2 tool path along the perimeter is extremely gouge prone (both local and global), especially as the bounding walls become more vertical. Attempting to find the accessible locations along this perimeter would cost more in computation time than in the savings generated from the tool paths. It is also true that the easiest and most efficient method for generating a prescribe fillet blend between the free-form surface and the cavity walls is via ball-end milling, not curvature matching.

Trimming away the border portion of the free-form surface insures that regardless of the cutter orientation, local gouging will not occur with the cavity walls. This is important because currently our local gouge-detection method only detects the local gouging on the drive surface. The authors are currently extending their local gouge detection to all (bordering) check surfaces. Trimming the surface is accomplished by first projecting the edges of the free-form surface onto a plane above the cavity surface; see Fig. 5. Next, the projected trimming curves are offset inward a tool diameter. The offset curves are then projected down to the surface(s) that define the cavity floor. These new projected edge curves are used to trim the original surface; see Fig. 5(c). The machining of this trimmed perimeter surface will be discussed later.

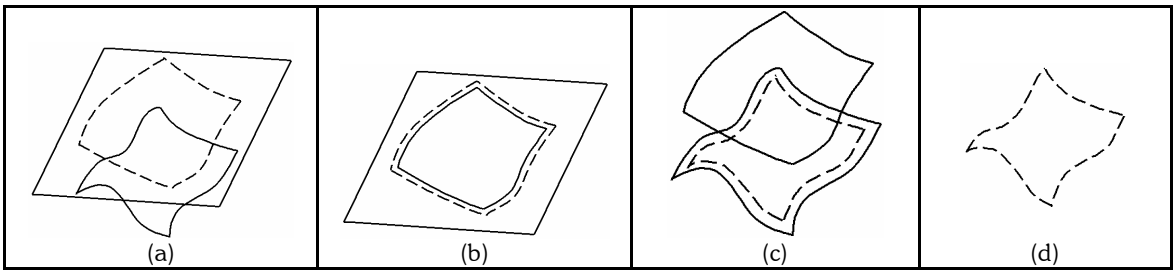

Fig. 5: Surface trimming examples, (a) project edges, (b) offset inward, (c) project onto surface, (d) trimmed surface.

3.2 Five-Axis Orientation and Positioning

Using the trimmed cavity surface, our method employs CM[2] to plan the flat- or filleted-end mill tool paths across the trimmed surface. In calculating tool path trajectories we plan along planar-cut or iso-curves of the trimmed surface. The final composite five-axis path is created by connecting these trajectories into a continuous tool motion where the overall strategy is either unidirectional cuts, lace cuts or a traditional pocketing (from inside out) routine.

The scheme for defining the inaccessible regions is slightly different, depending on the method used for surface curve generation. Let $S_T(u,v)$ denote the portion of the free-form surface that remains after the border surface has been trimmed. Let $Q(s)$ denote either a planar-cut parametric surface curve or iso-curve where s is not necessarily the arc length. A planar-cut parametric surface curve is generated on $S_T(u,v)$ by the intersection of $S_T(u,v)$ with parallel planes. A series of n parallel planes equally spaced by a distance d are used to create a series of curves to evenly machine $S_T(u,v)$; see Fig. 6.

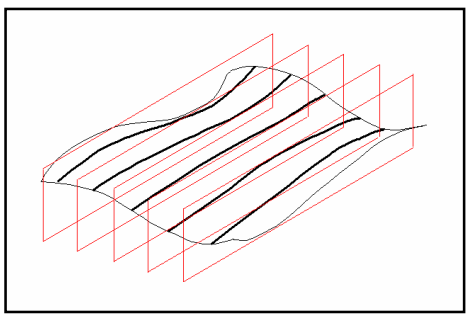

Fig. 6: Planar cuts.

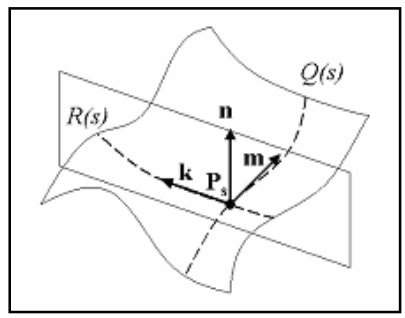

Fig. 7: Cross product of **m** and **k.**

If one desires to machine along iso-cut surface curves, our algorithm simply holds one of the parameters of $S_T(u,v)$ constant, v for example, and increment the other parameter, u. We then let the i^{th} iso-cut surface curve be represented by $Q_i(u)$, and let w represent the interval in v between adjacent curves.

To determine the orientation of the cutter at a point $\mathbf{P_s}$ on a surface curve *Q(s)*, another surface curve *R(s)* is created at $\mathbf{P_s}$ by the intersection of a plane created through $\mathbf{P_s}$ and normal to $\mathbf{m} = \frac{d}{ds}Q(s)$ where $\frac{d}{ds}Q(s)$ is the direction of cutter motion. A unit surface normal vector **n** defined by:

$$\mathbf{n} = \frac{\mathbf{m} \otimes \mathbf{k}}{|\mathbf{m} \otimes \mathbf{k}|}, \tag{1}$$

where $\mathbf{k} = \frac{d}{ds}R(s)$ and $\otimes$ is the cross-product; see Fig. 7.

To match the curvature of the swept silhouette of the cutting tool to the curvature of the surface orthogonal to the direction of tool travel, the cutter axis unit vector **t** in Fig. 8 is rotated in a clockwise direction about **b** (the negative of **k**) by an angle α,

$$\alpha = \sin^{-1}\frac{(R_C - R_F)}{\left(\frac{1}{k_n} - R_F\right)}. \tag{2}$$

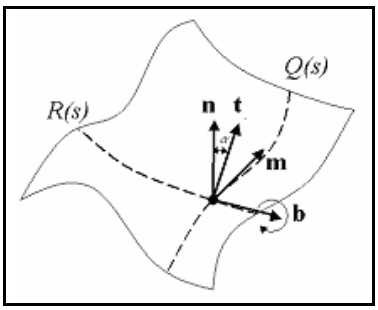

Fig. 8: **n** is rotated about **b** by α to produce **t.**

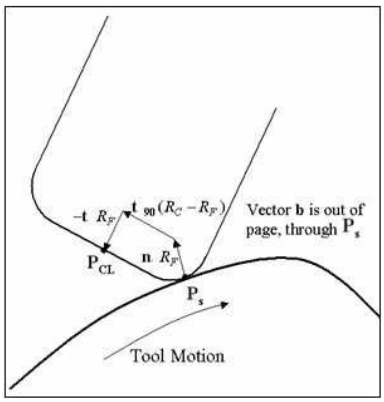

Fig. 9: Calculation of $\mathbf{P_{CL}}$.

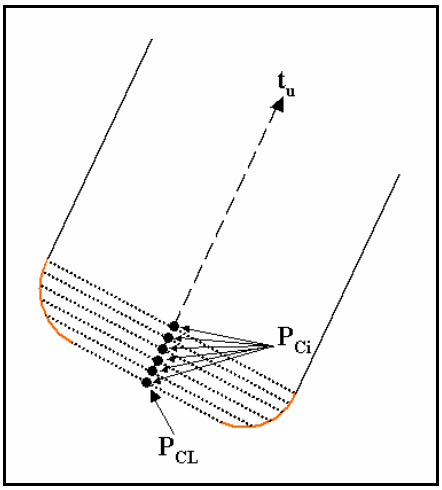

Fig. 10: Circles approximating bottom of cutter.

Where R_C and R_F are the cutter radius and fillet radius respectively and k_n, called the normal curvature, is the curvature of *R(s)* at $\mathbf{P_s}$.

Determining the cutter location point $\mathbf{P_{CL}}$ at $\mathbf{P_s}$ is straightforward, that is

$$\mathbf{P_{CL}} = \mathbf{P_s} + \mathbf{n}R_F + \mathbf{t_{90}}(R_C - R_F) - \mathbf{t}\ R_F, \tag{3}$$

where $\mathbf{t_{90}}$ is **t** rotated 90° counterclockwise about **b**; see Fig.9.

3.3 Local Gouge Detection and Correction

The purpose of local gouge detection is to monitor any unacceptable interference (penetration of the cutter below the surface profile tolerance) between the toroidal portion of the filleted-end mill and the surface being machined; as previously shown in Fig. 4. The local gouge detection method discussed in this section is derived from a method first presented by Petrizze [19]. Since we have already removed the perimeter of the cavity surface, we need only check for the local gouging on the trimmed surface. In addition, our method assumes that the radii of curvatures found on the cavity surface are enclosed by the tool radius and infinity. This means that we used the toroidal (or flat portion) of the cutter to machine convex regions.

To represent the toroidal portion of the cutter, a series of n concentric circles are used; see Fig. 10. This simply reduces the computational time by performing curve/surface intersections instead of surface/surface intersections. The center point P_{Ci} of the i^{th} circle is defined by the following equation:

$$\mathbf{P_{Ci}} = \mathbf{P_{CL}} + \mathbf{t}\left[\frac{R_F}{n} i\right] \tag{4}$$

The radius of the i^{th} circle, r_i, is defined by the equation:

$$r_i = (R_C - R_F) + \sqrt{R_F{}^2 - \left(R_F(\frac{i}{n})\right)^2} \tag{5}$$

where $i = 0,\ldots,n$. In our test cases the fillet radius was 0.03 inches and n was set to 10. However, if the fillet radius were to increase, more circles would be required. If local gouging is detected, **t** is rotated counterclockwise about **b** by some small positive value δ (generally 2 degrees).
To detect and correct for local gouging at a point $\mathbf{P_s}$ on $S(u,v)$, the following algorithm was developed and implemented:

Algorithm 1. Local gouge detection and correction.
Input: Point $\mathbf{P_s}$
Surface $S_T(u,v)$
Cutter axis unit vector **t**
vector **b**
Cutter location point $\mathbf{P_{CL}}$

Output: A CL point that produces no local gouging.

Begin

for the i^{th} circle in the series **do**

create i^{th} circle with a radius of r_i at $\mathbf{P_{Ci}}$

while(the i^{th} circle intersects $S_T(u,v)$)
rotate **t**, $\mathbf{P_{CL}}$ and the i^{th}circle by angle δ clockwise about **b**
output $\mathbf{P_{CL}}$ and **t**

End

3.4 Global Gouge Detection

The purpose of global gouge detection is to calculate any collisions between the cutter and adjacent surfaces. This method is designed to detect collisions between the upper portion of a filleted-end mill and adjacent surfaces. The ACIS solid modeling kernel was used to implement global gouge detection. The discussion below summarizes its use in computing bounding boxes between the tool and the cavity being machined.
Using $\mathbf{P_{CL}}$ and **t,** three cylinders representing the upper portion of the cutter, the tool holder and spindle are created with axes collinear to **t**; see Fig. 11. Bounding boxes are then computed around each of the three cylinders and collision surfaces. The collision surfaces chosen in this research are the drive surface, cavity walls and the top surface of the solid. While these do not represent all potential collision surfaces (e.g. clamps, trunions, etc.) our algorithm can be easily extend to include these other bounding boxes. To see if there is overlap, the cylinder bounding boxes are compared to the collision surface bounding boxes. If no bounding boxes overlap, $\mathbf{P_{CL}}$ and **t** are output to a file. Using the bounding box scheme is an efficient and inexpensive way to do a preliminary global gouge check.
If any two bounding boxes are found to overlap, the global gouge detection scheme enters a refinement phase, where the individual surfaces within the two overlapping bounding boxes are compared to see if they intersect, i.e. a surface/surface intersection calculation made. When the tool body is being checked during the gouge refinement phase, only its cylindrical surface is used, which significantly speeds up the intersection calculation. If no intersections are found, $\mathbf{P_{CL}}$ and **t** are output to a file. If an intersection is found, then CL data are output commanding the cutter to

raise to the retract plane, to avoid the collision. No other five-axis CL data is output until the path planner finds a location where no global gouging occurs. Subsequently CL data are again output to lower and position the cutter back on the drive surface, thus resuming the five-axis machining.

3.5 Creating the Inaccessible Regions

The global gouge detection algorithm is coupled to the creation of inaccessible regions. Inaccessible regions are those regions of $S_T(u,v)$ that, if CM2 machined, would cause the cutting tool to interfere with another surface. The inaccessible regions on $S_T(u,v)$ are defined on a plane above the surface. The regions are then projected down onto the surface; see Fig. 12. The manner in which these inaccessible regions are defined depends on whether the tooling paths are generated from iso-cut or planar-cut surface curves and guarantees that the free-form surface will be completely machined, leaving no small portion or sliver of the cavity not machined.

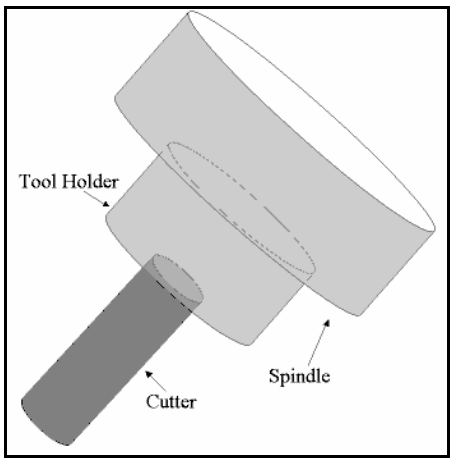

Fig. 11: Cylinder for global gouge detection.

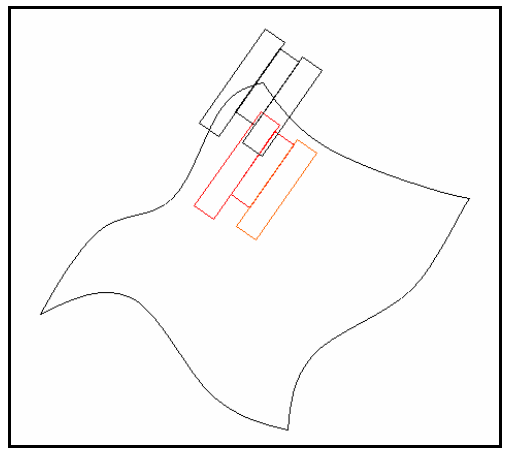

Fig. 12: Inaccessible regions projected onto surface.

Inaccessible regions formed from both types of tool path curves are defined by monitoring the first and last points in a series of gouge points on a surface curve. Let $\mathbf{P_{GF}}$ and $\mathbf{P_{GL}}$ represent the first and last points respectively in a series of inaccessible points on a surface curve.

3.5.1 Planar-cut Inaccessible Regions

Let $z_{retract}$ be equal to the z-value of a plane that is completely above $S_T(u,v)$ and parallel to the xy-plane. Without loss of generality, it is assumed that the path generation plane is parallel to the xz-plane of a chosen local coordinate system.

Let $\mathbf{P_L}$ be a point offset from, $\mathbf{P_{GF}}$ such that

$$\mathbf{P_L} = \mathbf{P_{GF}} - \begin{bmatrix}1\\0\\0\end{bmatrix} R_C - \begin{bmatrix}0\\1\\0\end{bmatrix} \frac{d}{2} + \begin{bmatrix}0\\0\\1\end{bmatrix} z_{retract} \tag{6}$$

As noted earlier, d represents the distance between the planar-cut surface curves, $Q(s)$. Let the z-coordinate of $\mathbf{P_L}$ be equal to the z-value of some retract plane above surface $S_T(u,v)$ and parallel to the xy-plane. Now let $\mathbf{P_H}$ be a point offset from $\mathbf{P_{GL}}$, where

$$\mathbf{P_H} = \mathbf{P_{GL}} + \begin{bmatrix}1\\0\\0\end{bmatrix} R_C + \begin{bmatrix}0\\1\\0\end{bmatrix} \frac{d}{2} + \begin{bmatrix}0\\0\\1\end{bmatrix} z_{retract} \tag{7}$$

Four lines forming a closed loop or rectangle are defined from the two points $\mathbf{P_L}$ and $\mathbf{P_H}$; see Fig. 13. This inaccessible region is projected down onto the trimmed drive surface and trimmed from $S_T(u,v)$.

3.5.2 Iso-cut Inaccessible Regions

Forming the inaccessible regions from iso-cut tool path curves is accomplished in a slightly different manner. Let four points $\mathbf{P_1}$, $\mathbf{P_2}$, $\mathbf{P_3}$, and $\mathbf{P_4}$ define an inaccessible region, where

$$\mathbf{P_1} = Q_{i+\frac{w}{2}}(u_f - c_1) \tag{8}$$

$$\mathbf{P_2} = Q_{i-\frac{w}{2}}(u_f - c_2) \tag{9}$$

$$\mathbf{P_3} = Q_{i-\frac{w}{2}}(u_l + c_3) \tag{10}$$

$$\mathbf{P_4} = Q_{i+\frac{w}{2}}(u_l + c_4)\,. \tag{11}$$

$Q_{i-\frac{w}{2}}(u)$ and $Q_{i+\frac{w}{2}}(u)$ are iso-cut surface curves that are midway between the i^{th} curve and the i^{th}+1 and i^{th}-1 curves respectively. u_f and u_l represent the parameters on $Q_i(u)$, defining the first and last gouge points in a series of inaccessible points. C_j represents parameter increments along $Q_i(u)$ equal in magnitude to the cutter radius. Let the z-coordinate of $\mathbf{P_1}$, $\mathbf{P_2}$, $\mathbf{P_3}$, $\mathbf{P_4}$ be equal to $z_{retract}$.

A closed region is formed in Fig. 14 by straight lines from $\mathbf{P_1}$ to $\mathbf{P_2}$ and $\mathbf{P_3}$ to $\mathbf{P_4}$, and portions of iso-curves $Q_{i-\frac{w}{2}}(u)$ and $Q_{i+\frac{w}{2}}(u)$ from $\mathbf{P_1}$ to $\mathbf{P_4}$ and $\mathbf{P_2}$ to $\mathbf{P_3}$, respectively.

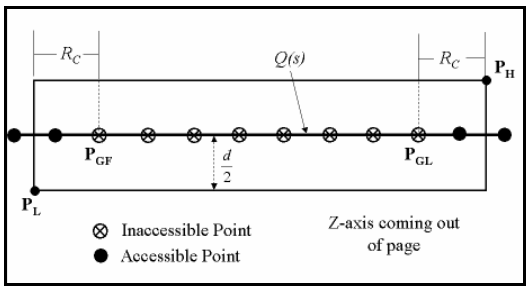

Fig. 13: Defining planar-cut inaccessible region.

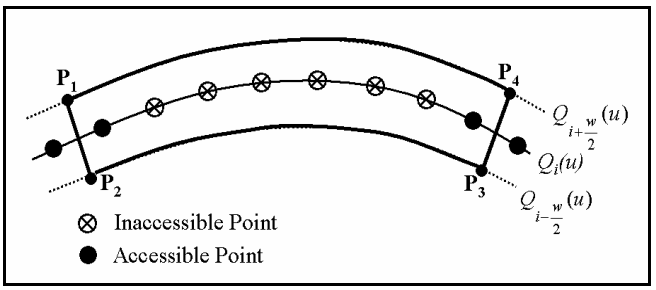

Fig. 14: Defining iso-cut inaccessible region.

As before this closed loop is projected down onto the $S_T(u,v)$ surface and trimmed away. The inaccessible regions are machined using a ball-end milling approach and the inaccessible region algorithm implemented for this research follows.

Algorithm 2. Inaccessible region creation

Input: Surface $S_T(u,v)$

Surface point $\mathbf{P_s}$

Unit cutter axis vector $\mathbf{t}$

Surrounding collision surfaces

CL point $\mathbf{P_{CL}}$

Cutter radius R_C

Fillet radius R_F

Output:

A CL point that produces no global gouging

Inaccessible regions

Begin

At $\mathbf{P_s}$, using $\mathbf{P_{CL}}$ create cylinders representing the top portion of cutter, tool holder and spindle.

Create bounding boxes around cylinders and collision surfaces.

Compare bounding boxes for overlapping.

if (overlapping exists)

Check if surfaces overlap with surface/surface intersection calculation.

if (surfaces intersect)

Define $\mathbf{P_L}$ [or $\mathbf{P_1}$ and $\mathbf{P_2}$].

else if(point is first accessible after a series of gouges)

Define $\mathbf{P_H}$ [or $\mathbf{P_3}$ and $\mathbf{P_4}$].

Output CL point

Define $\mathbf{P_H}$ [or $\mathbf{P_3}$ and $\mathbf{P_4}$].

Output CL point

else

Output CL point

Create inaccessible region.

Project region onto $S_T(u,v)$.

Generate three-axis tooling paths on inaccessible region.

Delete $\mathbf{P_L}$ and $\mathbf{P_H}$ [or $\mathbf{P_1}$, $\mathbf{P_2}$, $\mathbf{P_3}$, $\mathbf{P_4}$]

End

4. RESULTS

To determine the robustness of our algorithms and the validity of using traditional and non-traditional milling methods to machine free-form cavity surfaces, six generalized surfaces were defined; see Fig. 15. These surfaces were placed in a cavity sufficiently deep to allow a minimum of 0.125 inches of raw stock to be removed. While these test surfaces do not cover all mold and die possibilities, they do represent a wide variation of surfaces. Surfaces A and B have multiple saddle regions with varying changes in curvature. Surface C is defined such that half of its area is concave with the other half convex. Surface D is completely convex; surface E has large planar regions and surface F is completely concave.

All surfaces were machined on a BostonDigital 505 five-axis mill. A 0.250 inch diameter filleted end mill, with a 0.030 inch corner radius, was used to path plan the CM^2 paths. A 0.500 inch diameter ball-end mill was used to path plan the inaccessible regions. In an attempt to minimize any handwork in the inaccessible or interface regions of these surfaces, the following machining best practices were employed: (1) When using multiple tools to machine a surface one must determine tool length offsets via a highly accurate and repeatable laser tool length measurement system; (2) Programmed scallop height of the ball-end mill must be set to ½ the programmed value used on the filleted-end mill.; and (3) Where possible avoid positioning the ball-end mill such that its tool tip is in contact with the drive surface.

Any theoretical or practical researcher of path planning strategies knows that these practices are not full proof and depend heavily on the rigidity of the mill, the depth and width of cut taken, as well as other machining parameters and fixturing conditions. Assuming one chooses these with some care, and considering our research is focused on finish surface machining, we visually inspected our machined surface with a MICROINCH™ FLEXBAR end-milling comparator and saw minimal pending handwork required to remove unwanted lay marks and scallops.

Fig. 16 shows the inaccessible regions that were removed from $S_T(u,v)$ during CM2 path planning for each test case. Tab. 1 contains the percentages of CM2 machining done on each surface. Surface A serves as a baseline for the other surfaces since no additional inaccessible regions were removed once the initial perimeter trim was performed; see Fig.16.

While these numbers are relative and depend both on the nature of the free-formed drive surface and the depth with which it is set down in the cavity, these results do clearly demonstrate the blending of traditional and non-traditional machining methods into an automated path planning strategy. When comparing the average of 10 tool paths per inch used by the CM^2 method to the over 80 passes per inch that were needed in the inaccessible regions to produce a 16-

32 μ inch (visually inspected) surface finish, one can clearly see the benefits this integrated strategy would have if used on the large open stamping molds and die of the aerospace and automotive industries.
In each case at least 50% if Sr(u, v) was accessible to our 5-axis CM^2 path planning algorithm. Due to the significant difference in tool path densities between the fillet-end and ball-end mill methods an average finish machining time savings of 60% was realized when base-lined against milling these surfaces with only a ball-end mill. This is comparable to the in-depth results of CM^2 when applied to various industrial surfaces [1],[9]. Fig. 17 shows all six surfaces cut in machinable wax. Fig. 18 shows surface C machined from a block of aluminum. In this example the step-over of the ball-end mill was set to 10 passes per inch so that the reader could easily visualize the location of the inaccessible regions.

Surface	*%CM^2*	*% Marked Inaccuracy (during CM2 planning)*
A	86	0
B	50	36
C	61	25
D	58	28
E	68	18
F	59	27

Tab. 1: Percent of surfaces accessible to CM.

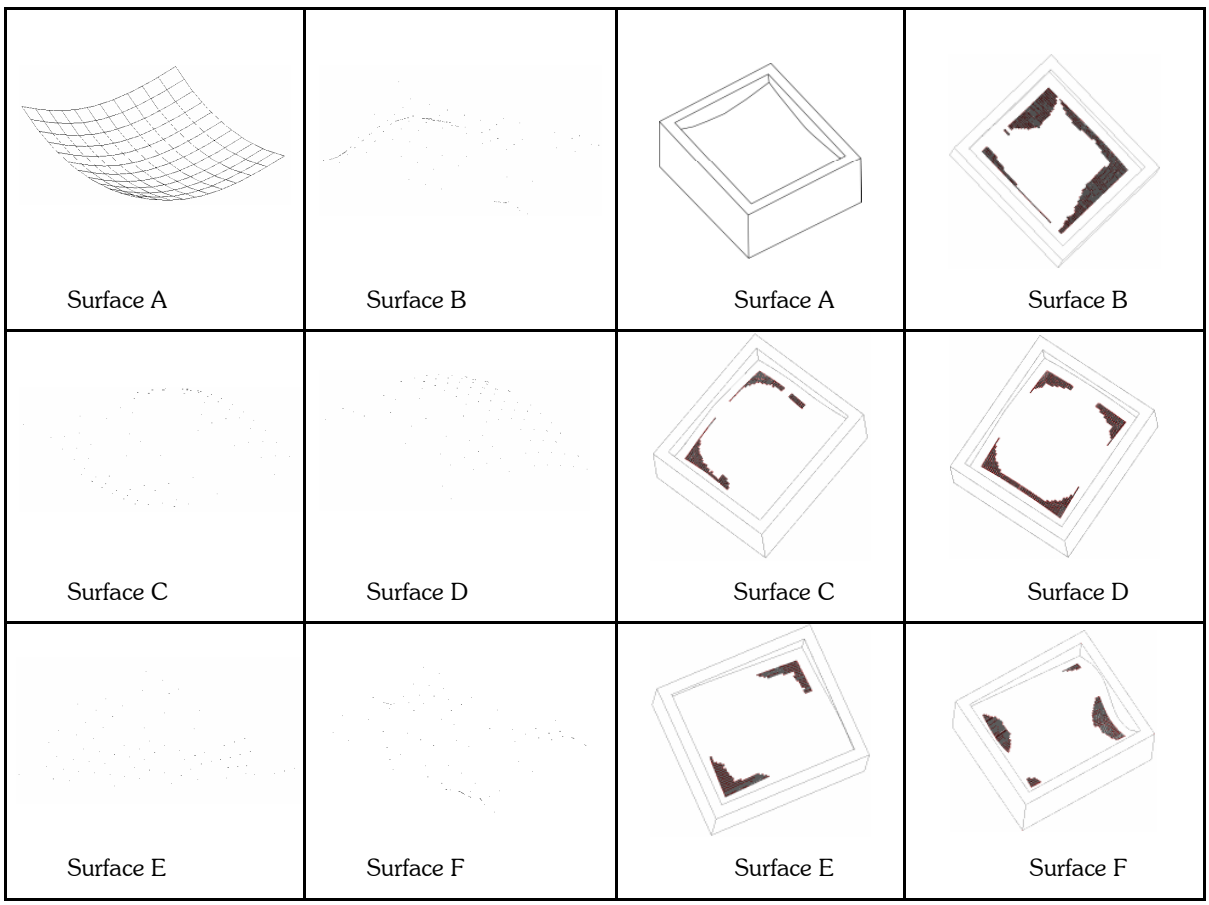

Fig. 15: Images of the machined surfaces.

Fig. 16: Images of the inaccessible regions.

In each case at least 50% of $S_T(u,v)$ was accessible to our five-axis CM2 path planning algorithm. Due to the significant difference in tool path densities between the fillet-end and ball-end mill methods an average finish machining time savings of 60% was realized when base-lined against milling these surfaces with only a ball-end mill. This is comparable to the in-depth results of CM2 when applied to various industrial surfaces [1,9]. Fig. 17 shows all six surfaces cut in machineable wax. Fig. 18 shows surface C machined from a block of aluminum. In this example the step-over of the ball-end mill was set to 10 pass per inch so that the reader could easily visualize the location of the inaccessible regions.

Surface A

Surface B

Surface C

Surface D

Surface E

Surface F

Fig. 17: Machined surfaces.

Fig. 18: Block C milled in aluminum.

5. CONCLUSIONS AND FUTURE WORK

This research clearly shows the union of traditional and non-traditional multi-axis machining methods applied to free-form surfaces embedded within VWC's. This was made possible by applying both local and global gouge detection and corrections method to the five-axis tool orientation and position vectors. When correction was not possible, inaccessible regions were computed and removed for subsequent three-axis ball-end mill machining. A significant amount of milling time was saved by using good machining practice when using multiple tools to machine a cavity and by applying CM^2 methods to the largest possible regions of the cavity. While the authors cannot predict the actual savings one will gain when applying our methods to each and every die or mold, it is clear that this algorithm should find its way into commercial CAD/CAM packages.

Efforts are currently underway to take our ACIS five-axis/ball-end mill VWC implementation and port it to commercial CAD/CAM tools such as CATIA, Pro/Engineering and Unigraphics. This will allow us to take full advantage of a host of existing machining methods and procedures. Future attention must be given to the machining of islands within a VWC,

and the milling of VWC's within VWC's, and the partitioning of more complex free-form surfaces with multiple patch continuities and increased trimming curves.

6. REFERENCES

[1] Cheng, Z; Cheatham, R.; Wang, J.; Jensen, C. G.; Chen, Y.; Bowman, B.: A Benchmark Comparison of CAD 5-axis Machining Packages, 2004 ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Salt Lake City, UT Sept. 28 – Oct. 2 2004.

[2] Cho, I.; Lee, K.; Kim, J.: Generation of Collision-Free Cutter Location Data in Five-axis Milling Using the Potential Energy Method, International J Adv Manuf Technol, 13, 1997, 523-529.

[3] Choi, B. K.; Park, J. W.; Jun, C. S.: Cutter-Location Data Optimization in 5-axis Surface Machining, Computer-Aided Design, 25(6), 1993, 377-386.

[4] Choi, B.; Jun, C.: Ball-End Cutter Interference Avoidance in NC Machining of Sculptured Surfaces, Computer-Aided Design, 21(6), 1989, 371-378.

[5] Elber, G.; Cohen E.: A Unified Approach to Accessibility in 5-Axis Freeform Milling Environments, Computer-Aided Design, 31(13), 1999, 796-804.

[6] Elber, G.; Cohen, E.: Hidden Curve Removal for Free Form Surfaces, Computer Graphics, 32(4), 1990, 95-104.

[7] Foley, J. D. et al: Computer Graphics: Principles and Practices in C, Second Edition, Addison-Wesley Publishing Company, Reading, MA, 1996, 649-720.

[8] Han, Z.; Yang, D.: Iso-Phote Based Tool-Path Generation for Machining Free-Form Surfaces, Journal of Manufacturing Science and Engineering- Transactions of the ASME, 121(4), 1999, 656-664.

[9] Hill, J. K.; Cheatham, R. M.; Jensen, C. G.: A Benchmark Comparison Of Curvature Matched Machining Versus Three- And Five-Axis Machining Methods Applied To Industrial Part Surfaces, Proceedings of DETC2002: ASME Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Montreal, Quebec Canada, 2002.

[10] Hornung, C.; Lellek, W.; Rehwald, P.; Strasser, W.: An Area-Oriented Analytical Visibility Method for Displaying Parametrically Defined Tensor-Product Surfaces, Computer-Aided Geometric Design, 2(1-3), 1985, 197-205.

[12] Hwang, J. S.: Interference-Free Tool-Path Generation in the NC Machining of Parametric Compound Surfaces, Computer-Aided Design, 24(12), 1992, 667-676.

[13] Jensen, C. G.; Anderson, D. C.: Accurate Tool Placement and Orientation for Finish Surface Machining, Journal of Design and Manufacturing, 3, 1993, 251-61.

[14] Jensen C. G.; Red, W. E.; Pi, J.: Tool Selection for Five-Axis Curvature Matched Machining, Computer-Aided Design, 34(3), 2002, 251-266.

[15] Lee, Y. S.; Chang, T. C.: 2-Phase Approach to Global Tool Interference Avoidance in 5-Axis Machining, Computer-Aided Design, 27(2), 1995, 715-729.

[16] Lee, Y. S.; Ji, H.: Surface Interrogation and Machining Strip Evaluation for 5-Axis CNC Die and Mold Machining, International Journal of Production Research, 35(1), 1997, 225-252.

[17] Loney, G. C.; Ozsoy, T. M.: NC Machining of Free-Form Surfaces, Computer-Aided Design, 19(2), 1987, 85-90.

[18] Morishige, K.; Takeuchi, Y.; Kase, K.: Tool Path Generation Using C-Space for 5-Axis Control Machining, Journal of Manufacturing Science and Engineering, Transactions of the ASME, 121(1), 1999, 141-149.

[19] Petrizze, N. P.: An Analytical Study of Interference Algorithms Applied to Curvature Matched Machining, Masters Thesis, Brigham Young University, Provo, UT, 1997.

[20] Pi, J.: Automatic Tool Selection and Tool Path Generation for Five-Axis Surface Machining, Masters Thesis, Brigham Young University, Provo, UT, 1996.

[21] Pi, J.; Red, W. E.; Jensen, C. G.: Grind-free Tool Path Generation for Five-Axis Surface Machining, Computer Integrated Manufacturing Systems, 11(4), 1998, 337-50.

[22] Pi, J.; Red, E.; Jensen, C. G.: Grind-Free Tool Path Generation for Five-Axis Surface Machining, Computer Integrated Manufacturing System, 11(4), 1998, 337-350.

[23] Tseng, Y. J.; Joshi, S.: Determining Feasible Tool-approach Directions for Machining Bezier Curves and Surfaces, Computer-Aided Design, 23(5), 1991, 367-379.

[24] Wang, X. C. et al: Curvature Catering – A New Approach in Manufacture of Sculptured Surface, Journal of Materials Processing Technology, 38(1-2), 1993, 159-194.

[25] You, C.; Chu, C.: Automatic Correction of Tool Interference in Five-Axis NC Machining of Multiple Surfaces, Journal of Chinese Society of Mechanical Engineers, 17(5), 1996, 435-442.

[26] Yang, D. C. H.; Han, Z.: Interference Detection and Optimal Tool Selection in 3-Axis NC Machining of Free-Form Surfaces, Computer-Aided Design, 31(5), 1999, 303-315.
[27] Yoon, J. H.; Pottmann, H.; Lee Y. S.: Locally Optimal Cutting Positions for 5-Axis Sculptured Surface Machining, Computer-Aided Design, 35(1), 2003, 69-81.