Title

Tool Path Planning Generation For Finish Machining of Freeform Surfaces in the Cybercut Process Planning Pipeline

Authors

Wright, Paul K
Dornfeld, David
Sundararajan, V.
et al.

Publication Date

2007-05-02

TOOL PATH GENERATION FOR FINISH MACHINING OF FREEFORM SURFACES IN THE CYBERCUT PROCESS PLANNING PIPELINE

Paul K. Wright, David A. Dornfeld, V. Sundararajan, and Debananda Misra.
The Berkeley Manufacturing Institute
University of California at Berkeley
Berkeley, CA 94720

ABSTRACT

The research describes part of a "Pipeline of Design and Manufacturing Tools" for product designers who are driven by short delivery-times. The overall project has been called CyberCut because the modules can interoperate in a distributed, Internet-based environment. This particular paper focuses on tool path generation of sculptured surfaces using 3-axis CNC machines. Importance is given to generating cutter location points that will meet the tolerance requirement while maintaining a certain surface finish. Sections of the paper describe: a) the offset-generation method, b) the tool path generation scheme and c) the tool holder collision detection algorithm. The algorithms that have been developed are used to machine sample parts.

INTRODUCTION

The CyberCut project at the University of California at Berkeley, aims to develop an Internet-coordinated manufacturing service for rapid part design and fabrication [1]. CyberCut is based on the idea of “concepts to parts” in analogy to the services provided by MOSIS for integrated circuits [2]. However, unlike the typical VLSI fabrication process (e.g., CMOS, NMOS etc.), the link between conceptual design and fabrication in the mechanical domain is more complicated and is not as well defined. Therefore, the parts must be designed with some basic understanding of the process by which they will be fabricated. This forces the designer, during the conceptual design phase, to consider the “physical constraints” of the downstream fabrication process. Such constraints are more challenging for machining than for solid freeform fabrication (SFF). Design for Manufacturability (DfM) constraints in 2.5D machining include: a) designing to allow for corner radii in pockets, b) allowing for tool-length relative to pocket depth, c) allowing for tool-holder collisions, and d) anticipating fixturing constraints. DfM constraints for parts that have a combination of 2.5D and 3D sculptured surfaces are similar, and the complex geometry requires additional considerations concerning tool-end radius versus surface finish.

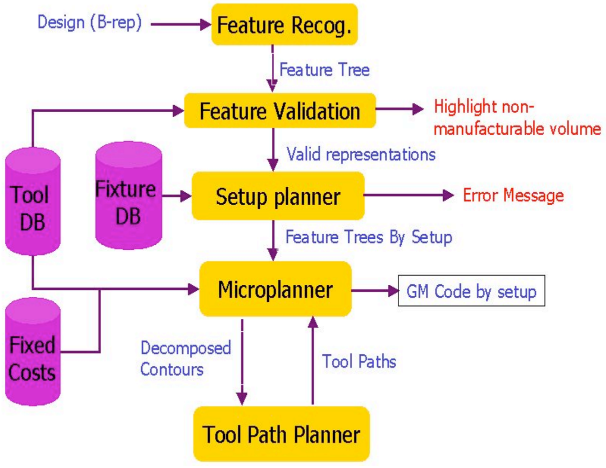


FIGURE 1. 2.5D PIPELINE [1, 2].

2.5D PIPELINE AND 3D SURFACES

Figure 1 shows the 2.5D pipeline that was reported in previous studies by Ahn et al [1]. The 2.5D pipeline was capable of accepting CAD files in a boundary representation and issuing a set of instructions for the machinist and G&M codes for the machine tools.

The current work extends the pipeline for the machining of parts with freeform surfaces such as injection molds. Given a part with arbitrary sculptured surfaces, G&M machine codes have been generated to machine that surface using ball-end milling cutters on a 3-axis CNC machine.

The pipeline comprises of several modules that work in tandem to generate tool paths for the machine. The input to the system is a boundary representation of the part. The part is decomposed to machining volumes called features by a feature recognition module. The features are classified according to their type as pure 2.5D features (comprising a 2D contour along with an extrusion depth), 2.5D features with draft angles and freeform features. These features are passed on to a Feature Validation module that examines the features and tests them against a tool database to determine tool availability and accessibility. Errors at this stage are reported to the designer and the system halts. Otherwise, the features are passed on to the setup planner that determines admissible fixture configurations and passes on the features to the Microplanner. This module is responsible for the optimal selection of tools and sequencing them. The features are finally passed on to the tool path planner which plans for the features.

This paper focuses on the finish tool path planning aspects for freeform surfaces. Details of the other modules may be found in [1]. In particular, it assumes the tool selection has been performed by the microplanner and that rough machining has already be done. The microplanner considers the maximum curvature of the surface (i.e. the minimum radius of the tool) and assigns it for the finishing operation. Machining large surfaces with tools of small radii does have the disadvantage of increased machining time and reduced cutter strength. However, from a finish machining perspective, this is tolerable since the amount of material removal is very small. One possible solution that has not been explored in our work is to use multiple finish passes. The preliminary finish passes could be done with a set of tools based on the curvature of the regions i.e. regions with low curvatures could be done with larger tools and then the regions left behind can be cleaned up smaller tools. The final finish pass could be performed with a cutter of the radius corresponding to the highest curvature. It should, however, be noted that the Feature Validation stage of the process planning will reject any surfaces with curvatures higher than that can be achieved using the tools from the database.

Tool path planning has been done on the ‘offset surface’. Iso-planar zig-zag toolpaths, due to their robustness and simplicity, have been generated. The generated tool paths comply with a certain tolerance and surface finish specified on the surface. Tool holder collision detection has been done to help select the appropriate tool. The input is a solid model of the surface in .SAT format while the output is a text file containing the machine code. Examples are presented to validate the algorithm.

Broadly, sculptured surface machining is done in two stages, 1. *roughing* and 2. *finishing*. The majority of the material is removed in the roughing stage using larger tool(s) while the left over material is removed in the finishing stage. Typically, end milling cutters are used for roughing operations [4-6] while ball end milling cutters are used for finishing operations.

Due to the inherent nature of the motion of the machine tools, tool paths are always a series of straight lines/arcs whereas the actual sculptured surface is a surface of varying slope and curvature. Thus, the sculptured surface is approximated by a series of straight lines and arcs for machining, and the machining goal is to get as close as possible to the design surface. The current commercially available CAD/CAM systems for generating cutter location (CL) data for machining of such sculptured surfaces require some important human decisions, such as determination of the precise interval between successive toolpaths. After the cutter location (CL) data are generated, post processing is done to get machine executable codes for actual production. A tool path interval that is too large can result in a rough surface while one that is too small can increase the machining time, making the process inefficient. Due to the complex geometry of the surface, tool body and tool holder interference with the surface pose many constraints on tool path generation. Smaller tool lengths and the vertical motion of the tool might make some regions inaccessible in the case of 3-axis CNC machining.

PREVIOUS WORK

In one of the earliest methods of machining sculptured surfaces by Sata et al. [7] chose small incremental iso-parametric curves as the tool paths. However, the non-uniform relationship between the parametric coordinates and the corresponding physical coordinate made tool path planning difficult. Huang et al. [8] and Li and Jerard [9] thus proposed an algorithm that generated tool paths based on the nonconstant parameter tool contact curves. At about the same time, Suh and Lee [10] and Hwang [11] presented methods to determine the tool paths by calculating, at each path, the smallest tool path interval and using it as a constant Offset in the next tool path. However, there still remained an unpredictable scallop height in the part surface, which caused either surface roughness or inefficient machining. A tool path strategy that kept the scallop height constant was thus presented by Suresh and Yang [12]. Their work led to a reduction of cutter location data; however reduction of cutter location data may not guarantee a proportional shortening in machining time. Feng et al. [13] also presented an approach for constant scallop height tool path generation. Sarma and Dutta [14] proposed a method to generate tool paths within any specified distribution of scallop heights on the manufactured surface. Tool paths were generated as offsets of an initial curve on the design surface. Choi et al. [15] generated tool paths using a configuration space (C-Space) approach. The geometric data describing the design surface and stock surface were transformed to the C-Space elements. Sarma [16] developed an algorithm to pick the direction of minimum number of switch-backs for a zig-zag path on a polygon. This led to reduction in machining time of planar surfaces. An offset parametric CNC tool path planning approach was developed by Kim and Kim [17]. Their idea was to first obtain an offset surface of a desired part surface and then obtain toolpaths on the offset parametric plane. Bicubic surface patch and surface sub-division were used to generate the Offset surface accurately. A new tool path scheduling method to improve the accuracy and efficiency of 3-axis surface machining was presented by Chih and Lin [18]. In their approach, the compromise between the efficiencies in the feed-forward segmentation and the path-interval segmentation was based on the selection of scallop height limit. The proposed method included compensation for the tool location to control the surface error and the segmentation efficiency. To the present authors' knowledge, none of the above approaches proposed solutions for tool holder collision detection. This detection is a contribution of this paper. Table 1 gives some of the basic definitions and notations that are being followed through the rest of the paper.

Term	Definition	Notation Used
CC-Point	Point where tool touches the design surface	C_i
CL-Point	Tool center location	L_i
Feed-forward Distance	Distance the tool travels in a single continuous tool motion	δ_{cc} : along CC-points δ_{cl} : along CL-points
Side-step Distance	Distance between two consecutive parallel tool paths	ϵ_{cc} : along CC-points ϵ_{cl} : along CL-points
Feed-forward error	Deviation of the tool path from the actual design surface	τ
Scallop Height	Height of material left between two consecutive passes of the tool	h

TABLE 1. BASIC DEFINITIONS.

TOOL PATH GENERATION METHOD

Zigzag tool-paths are used in the CyberCut system for the machining of freeform surfaces. The Cutter Location (CL) method is used to compute the tool paths i.e., tool paths are generated on the offset surface (Kim and Kim 1995; Choi et al. 1997) unlike the Cutter Contact (CC) method where tool paths are generated directly on the design surface. The steps involved in the generation of tool paths are as follows:

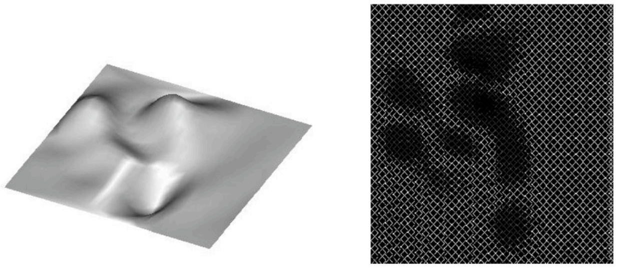
1. The surface is first discretized by sampling.
2. The inverse tool offset is computed.
3. Steps are taken in the feed-forward direction to meet the tolerance requirements.
4. The side-step amount is computed depending on the scallop height requirements.
5. Tests are performed to detect tool holder collisions.
6. Toolpaths are generated by connecting the points determined above by straight segments and NC codes are generated.

DISCRETIZATION OF THE SURFACE

Discretization of the surface involves sampling the design surface with a certain resolution and storing the *z* co-ordinate values associated with the sampled (x,y) co-ordinates. Since the surface is one of varying slope and curvature, it is desired to get denser points at steep regions than in the flat regions. In order to do this, surface-plane intersection are done to get the initial curves on the surface. The intersecting planes are chosen to be vertical and parallel to each other. To be specific, the plane with the normal along the y-axis is chosen to perform the intersection. The intersecting curves are sampled at constant arc length to the get the (x,y,z) coordinates of the discretized points. This has the benefit over uniform sampling in the (x,y) grid that in the steeper regions, the (x,y) increments can be reduced.The smaller the arc length, the finer are the points obtained on the surface. Figure 2(a) shows a design surface while Figure 2(b) shows the top view of its discretized model. The dark areas in the figure correspond to steep regions and the light areas correspond to at regions on the surface.

INVERSE TOOL OFFSET

The Inverse Tool Offset (ITO) algorithm [19, 21, 22] has been used to generate the offset surface. The algorithm states that the offset surface can be obtained by taking the envelope of the inverse of the tool in the z-direction, with the center of the inverted tool moving on the design surface (Figure 3). The discretization methodology adopted in this paper helps in getting accurate results for the ITO algorithm. For every discretized point on the design surface, the ITO algorithm gives a tool center location so that the tool is tangent to the surface. The algorithm works by considering all potential tool body collision points for each discretized point on the surface. The main advantage of the algorithm lies in generating gouge free Offset points [15].



(a) Design Surface (b) Top view of discretized surface

FIGURE 2. SURFACE DISCRETIZATION USING CONSTANT ARC LENGTH METHOD.

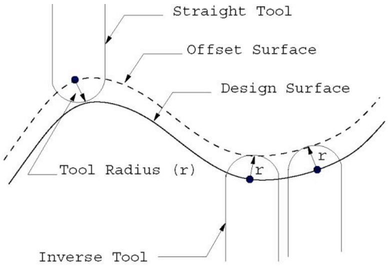


FIGURE 3. INVERSE TOOL OFFSET ALGORITHM.

TOOL PATH GENERATION STRATEGY

Zig-zag toolpaths are generated for the finishing operation. In the case of the zig-zag tool paths, it is desired that the tool moves in a straight line in the feed-forward direction. Planning tool motion on the design surface guarantees straight line motion of the tool contact point, but the tool center may not move in a straight line. On the other hand, tool path planning on the offset surface ensures straight line motion of the tool center. Hence, in the current approach, CL points have been directly planned on the offset surface generated through the ITO. Depending on the direction of motion of the tool, tool motion can be divided into two categories.

1. Feed-forward Marching

2. Side-step Marching

Offset surface generation, described in the previous section, gives a list of offset points (O_k) for a list of points on the design surface (D_k). The Offset points, however, do not meet the criterion of feed-forward or side-step marching. Hence, the objective is to find intermediate CC-points (C_k) on the surface and the corresponding offset points, which will be the required CL-points (L_k) that will meet the machining criterion of specified design-tolerance.

Feed-forward Marching

Most of the cutting operation is done in the feed-forward direction. The tool moves in straight line between two successive CC-points. The deviation of the tool path from the actual surface defines the error (Figure 4) in the feed-forward direction. The maximum error produced should be less then the design tolerance. The feed-forward distance that meets that tolerance requirement is

$$\delta_{cl} = 2[2\tau(r+\rho_f)-\tau^2]^{1/2} \tag{1}$$

δ_{cl} := Distance between two con CL-points in the feed-forward direction
r := Tool Radius
ρ_f := Curvature at the CC-point in feed-forward direction
τ := Specified tolerance

As can be inferred from the equation, a lower limit on error will demand a smaller feed-forward distance and thus more number of CL-points while a higher tolerance will give less number of CL-points. Also, for a given feed-forward distance a tool of higher radius will give less error then a tool of smaller radius. The following lines summarize the steps followed for feed-forward marching.

Algorithm - 1

```
Start current pass of the tool
- i, k = 0. Here, CL-point($L_o$) = Offset point($O_o$).
- For a CL-point ($L_i$) calculate the feed-forward distance ($\varepsilon_{cli}$ ).
- Get the distance ($d_k$) of the next Offset point($O_k$) from $L_i$ along the current pass in the feed-forward direction.
- Increment k till $d_k < \varepsilon_{cli} < d_{k+1}$.
- Linearly interpolate between $O_k$ and $O_{k+1}$ to get an intermediate offset point($O_{intm}$). Offset corresponding point ($C_{intm}$) on the design surface vertically below it to get the next CL-point ($L_{i+1}$).
- increment i.
End current pass
```

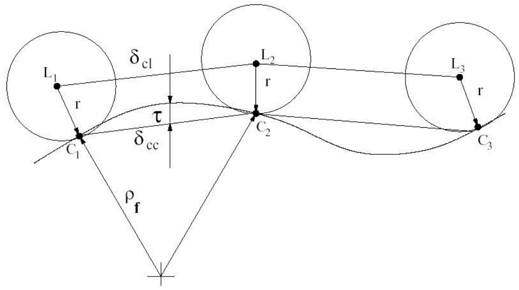


FIGURE 4. FEED FORWARD MARCHING.

Side-step Marching

Side-step marching is the motion of the tool in a direction orthogonal to the feed-forward direction. Referring to Figure 5, $\varepsilon_{cli}\cos(\theta)$ is the path interval where ε_{cli} is given by equation 2 (chapter V of Choi and Jerrard [19]). The maximum path interval at a CL-point is limited by the cusp height specified on the surface. After the CL-points for a single pass have been obtained by feed-forward marching, the minimum path interval of all those CL-points has been taken as the path interval for that specific pass. This ensures that the cusp height stays within the specified limit at all points on the surface.

$$\varepsilon_{cl} = \frac{\sqrt{4(r+\rho_s)^2(r+h)^2 - [r^2 + 2r\rho_s + (\rho_s + h)^2]^2}}{|r+h|} \tag{2}$$

Here, ε_{cl} := Distance between two consecutive parallel tool paths along the CL-points
r := Tool Radius
ρ_s := Radius of curvature at the CC-point in side-step direction
It can be seen from the equation that for a given side-step distance and curvature, a tool of larger radius will generate cusps of smaller height and thus will achieve better surface finish. The following steps summarize the method to obtain the path interval for a particular pass.

Algorithm - 2

```
Start current pass of the tool
- For every CL-point($L_i$) calculate the side step distance ($\varepsilon_{cli}$ ).
- Obtain the neighboring Offset points in the feed-forward direction on the next pass on the Offset surface in the side-step direction.
- Interpolate between the neighboring points to obtain the parallel Offset point to $L_i$ in the side-step direction. Compare this with $L_i$ to get the slope($\theta_i$).
- Calculate $\varepsilon_{cli}\cos(\theta)$) to get the path interval. increment i.
End current pass.
Find the minimum path interval of all the CL-points.
```

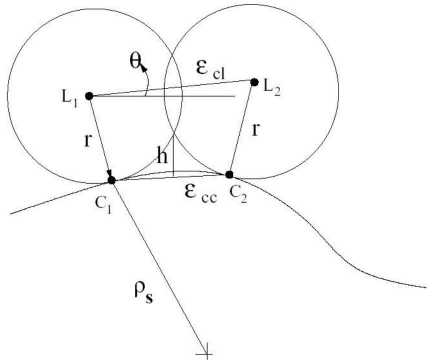


FIGURE 5. SIDE-STEP MARCHING.

Tool Selection Collision Detection

Tool selection is done based on the accessibility of the tool to all regions of the surface. The minimum curvature(ρ_{min}) of all the discretized points on the surface is obtained. The tool with radius r less the ρ_{min} has been selected for machining operation.

Tools of small diameter are used for the finishing operation. Typically, tools of smaller diameter are smaller in length. Hence, in some cases, even if the radius of the tool is bigger than the largest curvature of the surface at the point, the tool cannot enter into the region because of smaller tool body length. This is due to the tool holder, which collides with the surface. Thus, tool holder collision poses a constraint to finish machining. In the current approach the tool holder, along with the tool, has been abstracted as a series of stacked cylinders (Figure 6). Collision detection has been done both for the tool body and tool holder.

The process to detect the potential collision at a certain CL-point (x_0, y_0, z_0) for a certain tool of radius *r* involves finding all the points (x_i, y_i, z_i) on the discretized surface that lie with in the tool holder assembly.

Equation 3 describes the set of equations to check tool holder collisions.

$$(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2 < r^2 \; for \, z_i < z_0 \quad 3(a)$$

$$(x_i - x_0)^2 + (y_i - y_0)^2 < r^2 \ for \ z_0 \le z_i < z_0 + L \qquad 3(b)$$

$$(x_i - x_0)^2 + (y_i - y_0)^2 < r_j^{\,2} \ for \, z_0 + \sum_{k=0}^{j-1} L_k < z_0 + \sum_{k=1}^{j} L_k \ 3(c)$$

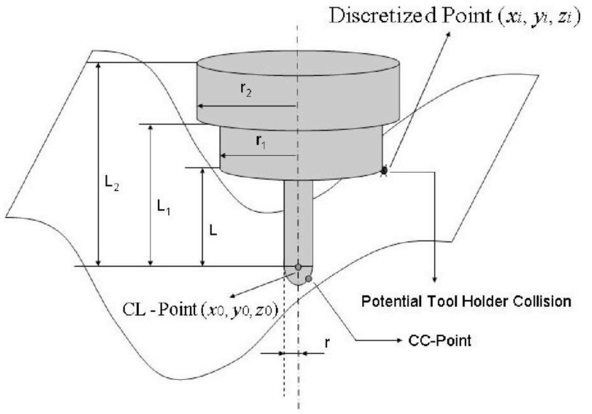
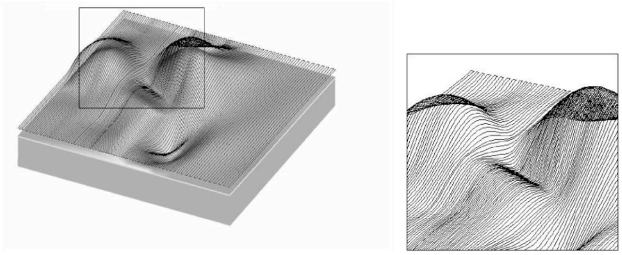


FIGURE 6. TOOL HOLDER COLLISION DETECTION.

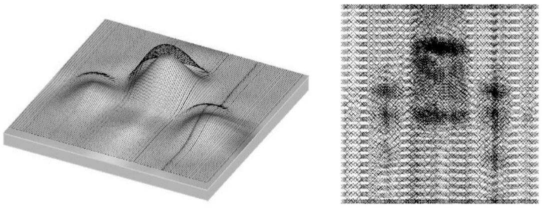
EXAMPLES AND IMPLEMENTATION

Figure 7(a) shows a design surface and the zig-zag tool paths generated to machine the surface. Figure 7(b) shows a magnified square region (Figure 7(a)) of the same tool paths. A maximum scallop height of 0.0015" and feed-forward error of 0.0006" has been used for this case. Ball-end milling cutter of 0.125" diameter was used for machining. Denser tool paths can be observed in steeper regions (Figure 7(b)) to keep the scallop height with in specified limits. Figure 8(a) shows another example part and the tool paths obtained. A maximum scallop height of 0.0012" and an allowable tolerance of 0.0008" has been used for this case. Ball-end milling cutter of 0.125" diameter was used for machining. Figure 8(b) shows top view of the CL-points obtained for the same part. Closely spaced CL-points have been obtained in steep regions. Figure 9 shows the machined surface using a 0.25" diameter ball-end milling cutter. A maximum scallop height of 25 micro-inch and an allowable tolerance of 0.0004" was used. The ACIS© Solid Modeling Kernel has been used for all solid modeling operations.



(a) Example part (b) Toolpaths at a specific region

FIGURE 7. EXAMPLE PART AND TOOL PATHS.



(a) Example Part (b) CL-points for generated parts

FIGURE 8. EXAMPLE PART AND GENERATED CL-POINTS.

MACHINED PARTS AND SURFACES

· (a) Example Part (b) Generated CL-points l-

chine a variety of soft aluminum molds for industrial collaborators. For pragmatic reasons, Auto-CAD's Inventor™ was used for the basic CAD work on the molds because it is based on the ACIS graphics kernel. It meant that in most cases, the Inventor/ACIS-based designs were compatible with CyberCut's ACIS-based pipeline, thereby facilitating feature recognition at the beginning of the pipeline. The process planners could then group and order the features before automatically generating the CNC files. In addition, during CAD, a flow modeling software program, C-Mold™, was used to verify that the injected plastic would penetrate to the cavity edges, assuring proper mold filling. Figure 10 shows an aluminum mold used for a small casing for an electronic product.

(a) Example part (b) Tool paths in a certain regions

FIGURE 9. MACHINED SURFACE USING 0.25IN DIAMETER CUTTER.

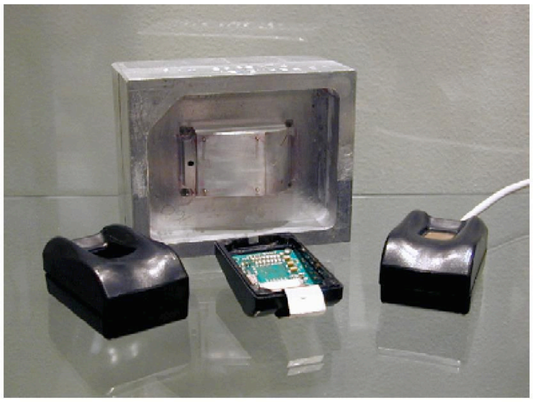


FIGURE 10. MACHINED MOLD FOR PLASTIC PARTS.

CONCLUSION

1. This paper focused on a CL-based approach to generate tool paths for finishing operations of sculptured surfaces. An inverse tool offset algorithm was used to generate the discretized offset surface. Tolerance requirements and maximum allowable scallop height were used as the limiting conditions for CL-data generation in the feed-forward and the side-step direction.
2. *Zig-zag tool path* planning on the offset surface ensured that the CL-points were placed on a straight line thus guaranteeing straight line motion of the tool.
3. Parallel curves on the surface were obtained by surface-plane intersection. Discretization of the surface was done by constant arc length marching on those curves. This gave dense points in steep regions as compared with those in at regions. The accuracy of the inverse tool offset algorithm was dependent on the resolution of the surface.
4. It was shown that the problem of tool holder collision can be successfully tackled by abstracting the tool body and holder as a series of stacked cylinders
5. Linear interpolation and inverse tool Offset provided an efficient method for feed-forward marching and side-step direction. However, for accurate results, the resolution for discretized Offset surface needs to be properly chosen
6. It is clear that the side-step strategy is based on the worst case scenario i.e. the minimum of all the side-step values is chosen to meet the scallop requirements. This can give rise to inefficiencies if the surface contains large flat regions and few very sharply curved regions. A surface decomposition approach with multiple finishing passes can alleviate this problem considerably. The final finish pass would still take a long time. A possible solution is to decompose the surface as above and then construct special tool-paths for the regions of transition between the different decomposed patches.

ACKNOWLEDGMENT

This work was funded by NSF-DMI and we thank Carlo Sequin, Sara McMains, Ganping Sun, Palani Kumaresan and Roshan D’Souza.

REFERENCES

1. Ahn, S., Sundararajan, V., Smith, C., Kannan, B., D’Souza, R., Sun, G., Mohole, A., Wright, P., Kim, J., McMains, S., Smith, J., and Sequin, C., “CyberCut: An Internet-based CAD/CAM system,” *J. of Computing & Information Science in Engineering*, Volume 1, 2001, pp. 52-58.

2. Sequin, C. and McMains, S., 1995, “What Can SFF CAD Learn From the VLSI Revolution”, *NSF Workshop on SFF*, CMU, June, 1995

3. D’Souza, R., Wright, P.K. and Sequin, C., “Automated Microplanning for 2.5-D Pocket Machining,” *Journal of Manufacturing Systems*, 2001, Volume 20, Number 4, pp. 288-296.

4. Chan, K. W., Chiu, W. K., Tan, S. T., and Wong, T. N. (2003). A high-efficiency rough milling strategy for mold core machining. *Journal of Eng. Manufacture* (v217, n3), pp. 335-348.

5. Hwang, J. S. and Chang, T. (1998). Three-axis machining of compound surfaces using at and filleted endmills. *CAD* (v30, n8), pp. 641-647.

6. Lee, K., Kim, J. K., and Hong, S. E. (1994). Generation of toolpath with selection of proper tools for roughing. *CAD* (v26, n11), pp. 822-831.

7. Sata, T., Kimura, F., Okada, N., and Hosaka, M. (1981). A New Method of NC Interpolator for

Machining the Sculptured Surface. *Annals of the CIRP* (v30, n1), pp. 369-372.
8. Huang, Y. and Oliver, H. J. (1992). Non-constant parameter NC tool path generation on sculptured surfaces. *Journal of Computers in Engineering* (v1), pp. 411-418.
9. Li, S. X. and Jerard, R. B. (1992). Non-isoparametric three-axis NC tool-path generation for finish machining of sculptured surfaces. Geometric Modeling for Product Realization, Selected and Expanded Papers from the IFIP TC5/WG5.2 Working Conference on Geometric Modeling, Rensselaer, NY. 27 September - 1 October, (B8 of IFIP Transactions), pp. 251-265. **10**. Suh, Y. S. and Lee, K. (1990). NC Milling Tool-Path Generation For Arbitrary Pockets Defined By Sculptured Surfaces. *CAD* (v22, n5), pp. 273-283.
11. Hwang, J. (1992). Interference-free Tool-path in NC Machining of Parametric Compound Surfaces. *CAD*, (v24, n12), pp. 667-677.
12. Suresh, K. and Yang, D. C. H. (1994). Constant Scallop Height Machining of Free-form Surfaces. *J. of Eng. for Ind.* (v116), pp. 253-259.
13. Feng, H.-Y. and Li, H. (2002). Constant scallop height tool path generation for three-axis surface machining. *CAD* (v34), pp. 647-654.
14. Sarma, R. and Dutta, D. (1997). The Geometry and Generation of NC Tool Paths." *Journal of Mechanical Des.* (v119), pp. 253-258. **15**. Choi, B. K., Kim, D. H., and Jerrard, R. B. (1997). C-space approach to tool-path generation for mold machining. *CAD* (v29, n9), pp. 657-669.
16. Sarma, S. (1999). The crossing Function and its application to zig-zag tool paths. *CAD* (v31, n14), pp. 881-890.
17. Kim, K. I. and Kim, K. (1995). A new machine strategy for sculptured surfaces using Offset surface. *J. of Prod. Res.* (v33, n6), pp. 1583-1697.
18. Chih-Ching, L. and Lin, R.-S. (2001). An improved method for scheduling the tool paths for three-axis surface machining. *J. of Machine Tools and Manufacture* (v41), pp 133-147.
19. Choi, B. and Jerrard, R. (1998). Sculptured Surface Machining: Theory and Applications. Kluwer Academic Publishers, The Netherlands.
20. Kim, B. H. and Choi, B. (2000). Guide surface based tool path generation in 3-axis milling, *CAD* (v32), pp. 191-199.
21. Kondo, T., Kishinami, T., and Saito, K. (1988). Machining system based on Inverse Offset Method. *Journal of Japan Society of Precision Engineering* (v54, n5), pp971-976
22. Takeuchi, Y., Sakamoto, M., Abe, Y., and Orita, M. (1989). Development of a personal CAD/CAM System for Mold Manufacture Based on Solid Modeling Techniques *Annals of the CIRP* (v38, n1).