

Zig-Zag Tool Path Generation for Sculptured Surface Finishing

Debananda Misra * V.Sundararajan † Paul K. Wright ‡

{deb,vsundar,pwright}@me.berkeley.edu

Abstract

Sculptured surfaces are relatively difficult to machine due to their complex geometry. Sculptured surface machining primarily has two stages, the first being roughing and the other being finishing. The current work focuses on generating tool paths for the finishing operation. Given any arbitrary sculptured surface, G&M machine codes have been generated to machine that surface using ball-end milling cutters on a 3-axis CNC machine. 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 to 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.

Keywords: Sculptured Surface, Inverse Tool Offset, Feed-forward Marching, Tool Holder Collision

1 Introduction

Sculptured surfaces are extensively used in designing complex parts such as automobile bodies, turbine blades and packages for electronic hand-held devices. 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, flat end milling cutters are used for roughing operations (Chan et al. 2003; Hwang and Chang 1998; Lee et al. 1994) while ball end milling cutters are used for finishing operations.

Due to the inherent nature of the motion of the tool, 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,

*GE Global Research Center, Bangalore, India

†Department of Mechanical Engineering, University of California, Berkeley, CA, USA

‡Department of Mechanical Engineering, University of California, Berkeley, CA, USA

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.

The current work is aimed at providing a systematic approach to tool path generation of sculptured surfaces using 3-axis CNC machines. Tool paths have been generated for finishing operation only. Tool paths for the roughing stage are assumed to have been previously generated while a set of cutter location points are planned on the offset surface. Importance is given in generating cutter location points that will meet the tolerance requirement while maintaining a certain surface finish.

The layout of the paper is as follows: Sec. 2 reviews previous approaches to the same problem. Sec. 3 lists some of the definitions and notations used while Sec. 4 gives an overview of some of the standard tool path generation methods. Sec. 5 describes the offset generation method employed, followed by a section describing the tool path generation scheme in the current approach. Sec. 7 presents the tool holder collision detection algorithm. Sec. 8 and 9 show the resulting implementation of the algorithm with some example of machined parts.

2 Previous Work

In one of the earliest methods of machining sculptured surfaces, Sata et al. (1981) 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. (1992) and Li and Jerard (1992) thus proposed an algorithm that generated tool paths based on the non-constant parameter tool contact curves. At about the same time, Suh and Lee (1990) and Hwang (1992) 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 (1994). 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. (2002) also presented an approach for constant scallop height tool path generation. Sarma and Dutta (1997) 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. (1997) 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 (1999) developed an algorithm to pick the direction of minimum number of switch-backs for a zig-zag path on a polygon. This lead to reduction in machining time of planar surfaces. An offset parametric CNC tool path planning was developed by Kim and Kim (1995) . Their idea was to first obtain an offset surface of a desired part surface and then obtain toolpaths on the offset parametric plane. Bi-cubic 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 (2001). 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.

3 Basic Definitions and Notations Used

Below are 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

4 Tool Path Generation Methods

The main objective of tool path generation (TPG) is to compute a sequence of cutter location points (CL-points) from the design surface. Various investigators (Choi and Jerrard 1998; Kim and Choi 2000; Sarma and Dutta 1997) have given a detailed description and classification of various tool path generation methods. Tool-path generation methods are classified as either the *CC-based method* or the *CL-based method* depending on the type of tool path generation surface.

4.1 CC-Based Method

In the Cutter Contact(CC)-based method, tool paths are generated by sampling a sequence of CC-points from the part surface and then each CC-point is converted to a cutter location(CL)-point. Tool path generation is done on the design surface. In general this method can be classified in to 3 main categories as (1) *parametric method*; (2) *drive surface method*; (3) *guide plane method* depending on the path planning techniques. These are often called as the “conventional” tool path generation methods.

- *Parametric Method* computes the tool paths (Figure 1(a)) in the parametric domain and then maps it back to the design/part surface. Iso-parametric machining, one of the earliest techniques (Sata et al., 1981), involves machining along iso-parameter lines. Uniformly spaced parametric lines in the parametric domain are non-uniformly spaced on the part surface due to non-uniform transformation between Parametric and Euclidean spaces. This results in varying scallop height and thus non-uniform surface finish.
- *Drive Surface Method* computes toolpaths (Figure 1(b)) by intersecting the design surface with a series of planes called drive surfaces. The curves generated by intersection are then used to generate accurate tool paths. If the intersecting planes are parallel then its known as iso-planar machining. However, the orientation of the intersecting plane can be changed to achieve optimal conditions of machining. This method is very robust and can handle complex surfaces reliably.
- *Guide Plane Method* plans the tool paths (either in the form of straight line or contours) first on a 2D plane and then maps it back (Figure 1(c)) to the design surface. For 3-axis milling, the plane perpendicular to the tool axis is mainly used as the guide plane. The major strength of this method is that the shape of the region to be cut on the part surface can be considered during planning the tool path layout on the guide plane.

4.2 CL-Based Method

In the Cutter Location(CL)-based method the CL-surface is used as a path generation surface. In this approach, the CL surface has to be first generated from the design surface. Surface finish criteria are met on the CL surface. In most of the previous work in this area (Kim and Kim 1995; Choi et al. 1997), the offset surface was approximated as the CL-surface. *CL-based methods have been employed in the current work.* The offset surface is first generated from the design surface and tool paths are planned on the offset surface. Zig-zag tool paths, parallel to each other, are planned according to the machining parameters.

5 Offset Surface Generation

The tool path generation strategy adopted in this paper is based on the offset of the design surface. Thus, the final surface finish is greatly affected by the accuracy of the offset surface. Offset surface generation has

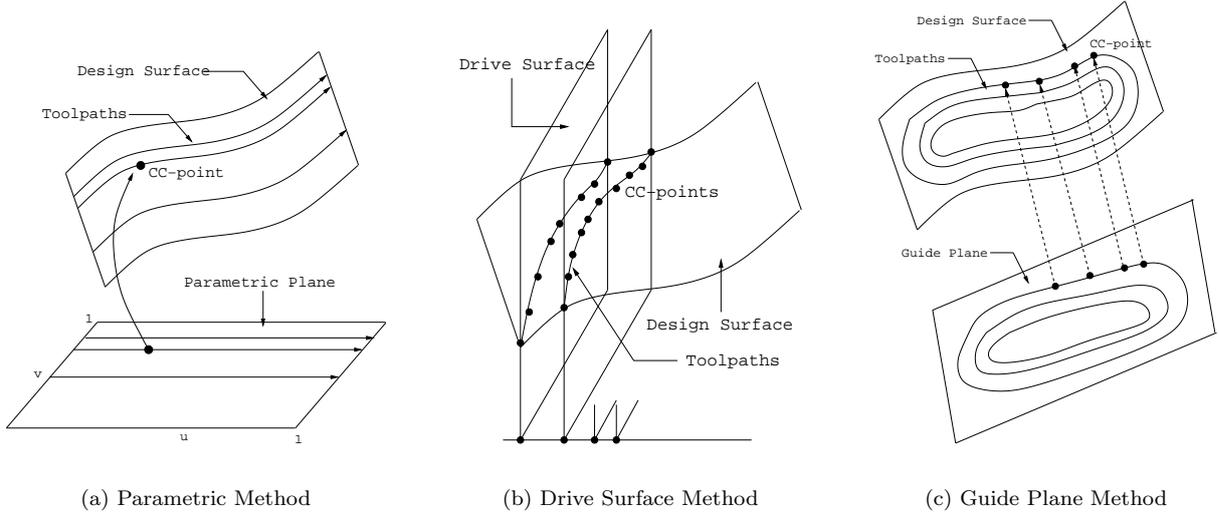


Figure 1: CC-Based Tool Path Generation Methods [based on refs. (Sarma and Dutta 1997; Choi and Jerrard 1998; Kim and Choi 2000)]

been done in two steps which are described below.

5.1 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 flat regions. In order to do this, surface-plane intersection has been done to get the initial curves on the surface. The intersecting planes have been chosen to be vertical and parallel to each other. The intersecting curves have been sampled at *constant arc length* to get the (x, y, z) coordinates of the discretized points. 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 flat regions on the surface.

5.2 Inverse Tool Offset

The Inverse Tool Offset (ITO) algorithm (Choi and Jerrard 1998; Takeuchi et al. 1989; Kondo et al. 1988) 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

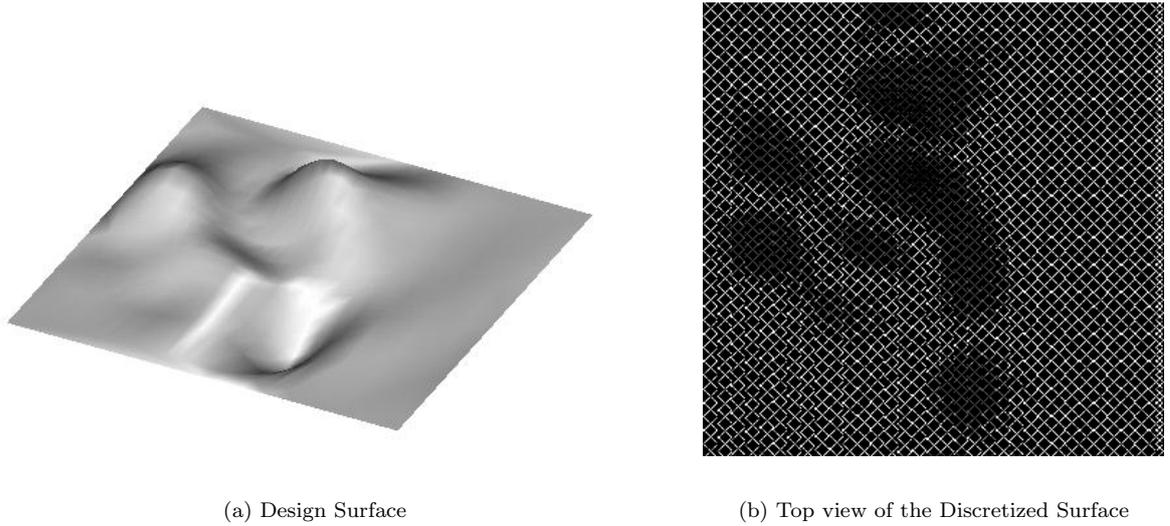


Figure 2: Surface Discretization using constant arc length method

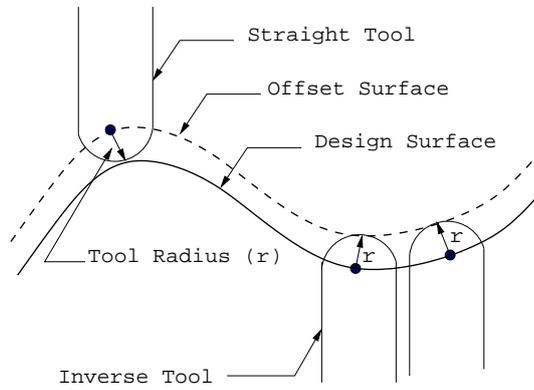


Figure 3: Inverse Tool Offset Algorithm

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 (Choi et al. 1997).

6 Tool Path Generation Strategy

In this paper zig-zag toolpaths have been 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.

6.1 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(a)) in the feed-forward direction. The maximum error produced should be less than the design tolerance. The feed-forward distance that meets that tolerance requirement is given by equation 1 (chapter V of Choi and Jerrard (1998)).

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

δ_{cl} := Distance between two consecutive CL-points in the feed-forward direction

r := Tool Radius

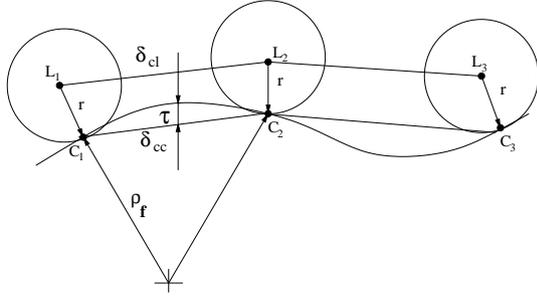
ρ_f := Curvature at the CC-point in feed-forward direction

Figure 4(b) shows the variation of feed-forward distance with change in tolerance for two different tools. The curvature has been kept the same for both cases. As can be seen from the graph, 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 than a tool of smaller radius. The following lines summarize the steps followed for feed-forward marching.

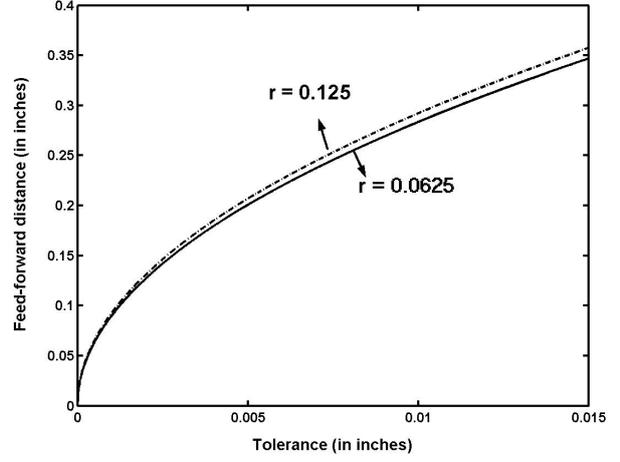
ALGORITHM - 1

- Start current pass of the tool
 - $i, k = 0$. Here, CL-point(L_0) = Offset point(O_0).
 - For a CL-point (L_i) calculate the feed-forward distance(δ_{cl_i}).
 - 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 $d_k < \delta_{cl_i} < d_{k+1}$.
 - 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



(a) Error in feed-forward direction



(b) Plot of Feed-forward Distance Vs Error

Figure 4: Feed-forward Marching

6.2 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(a), $\epsilon_{cl} \times \cos(\theta)$ is the path interval where ϵ_{cl} is given by equation 2 (chapter V of Choi and Jerrard (1998)). 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.

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

Figure 5(b) shows the variation of side-step distance with cusp height for two different tool radius. Again,

ϵ_{cl} := Distance between two consecutive parallel tool paths along the CL-points

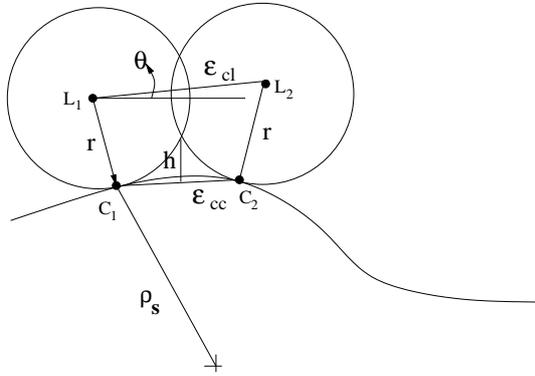
r := Tool Radius

ρ_s := Curvature at the CC-point in side-step direction

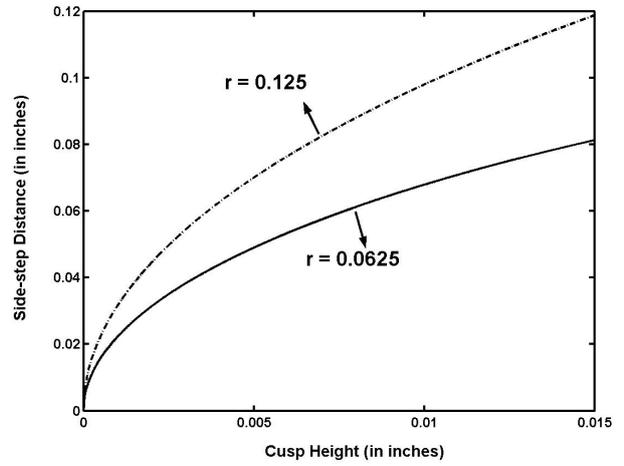
the curvature has been kept the same for both cases. It can be seen from the graph that for a given side-step distance and curvature, a tool of larger radius will generate cusps of smaller height and thus will achieve better surface finish. The following steps summarize the method to obtain the path interval for a particular pass.

ALGORITHM - 2

- Start current pass of the tool
 - For every CL-point(L_i) calculate the side step distance (ϵ_{cl_i}).
 - 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(θ_i).
 - Calculate $\epsilon_{cl_i} \times \cos(\theta_i)$ to get the path interval. increment i .
- end current pass.
- Find the minimum path interval of all the CL-points.



(a) Cusp height in side-step direction



(b) Plot of Side-step Distance Vs Cusp Height

Figure 5: Side-step Marching

7 Tool Selection and Tool Holder 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 within the tool holder assembly. Equation 3 describes the set of equations to check tool holder collisions.

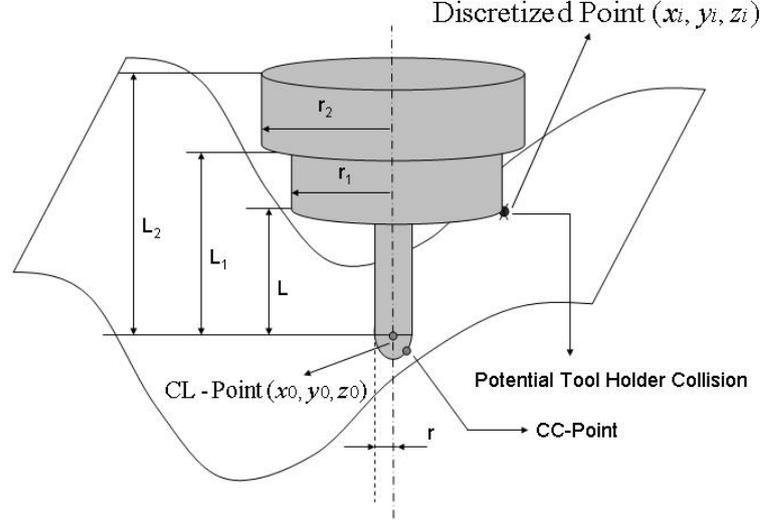


Figure 6: Tool Holder Collision Detection

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

$$(x_i - x_0)^2 + (y_i - y_0)^2 < r^2 \quad \text{for } z_0 \leq z_i < z_0 + L \quad (3b)$$

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

$$(j = 1 - 4, L_0 = L)$$

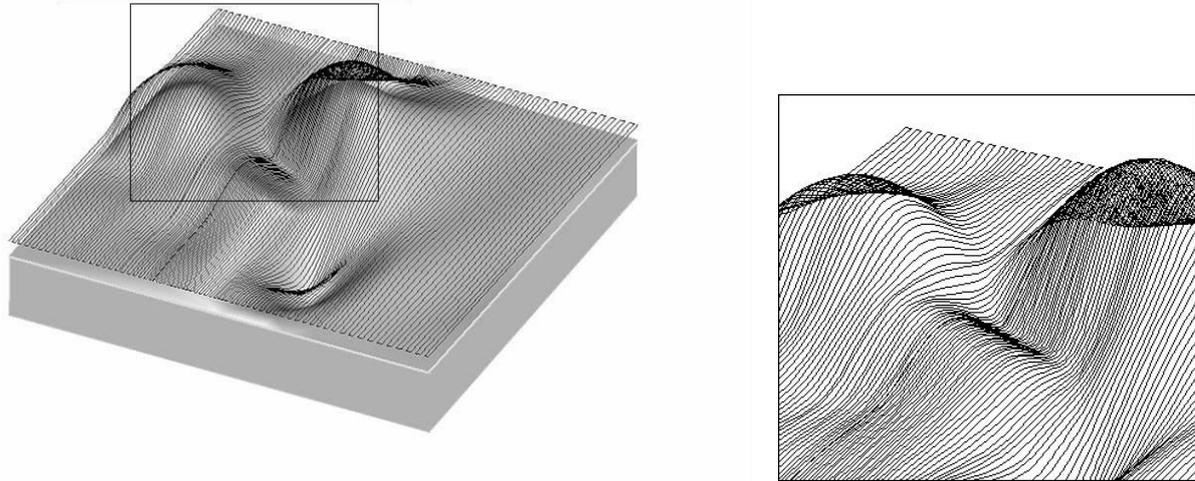
8 Implementation

The complete implementation procedure for generating zig-zag toolpaths is given below.

IMPLEMENTATION STEPS

- Input .SAT file
 - Surface-plane intersection to obtain parallel curves on the surface.
 - Constant arch length marching on the curves to obtain discretized model of the surface.
 - Select appropriate tool based on accessibility determination.
 - Inverse Tool Offset to obtain the discretized offset surface.
 - Begin starting feed.
 - * Do feed forward marching on the discretized offset points (*Algorithm 1*).

- * Find the side step distance (*Algorithm 2*).
 - * Do surface-plane intersection to get the discretized points on the surface.
 - * Offset to get discretized offset points.
 - end of surface.
 - Tool holder collision detection at the CL-points.
- Post process data to output .NC file.

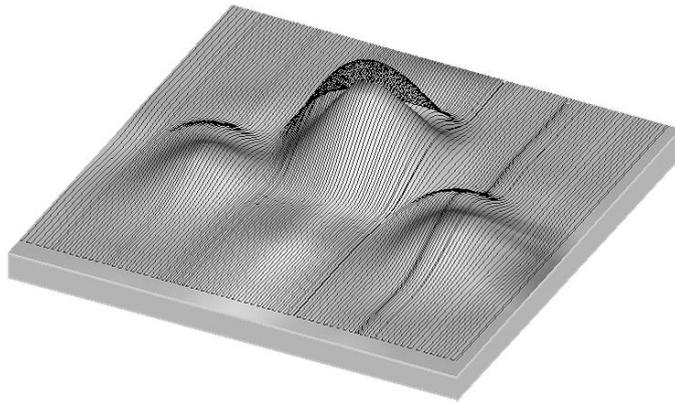


(a) Example part

(b) Toolpaths at a specific region

Figure 7: Example part and generated tool path

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 0.0015in and feed-forward error of 0.0006in has been used for this case. Ball-end milling cutter of 0.125in diameter was used for machining. Denser tool paths can be observed in steeper regions (Figure 7(b)) to keep the scallop height within specified limits. Figure 8(a) shows another example part and the tool paths obtained. A maximum scallop height of 0.0012in and an allowable tolerance of 0.0008in has been used for this case. Ball-end milling cutter of 0.125in 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 a machined surface using a 0.25 diameter ball-end milling cutter. A maximum scallop height of 25μ inch and an allowable tolerance of 0.0004in was used. ACIS[©] Solid Modelling Kernel has been used for all solid modelling operations.



(a) Example part



(b) CL-points for example part

Figure 8: Example part and generated CL-points

9 Conclusions

In this paper, a CL-based approach was taken to generate tool paths for finishing operation 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 respectively. The conclusions of this method can be summarized as follows

- *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.
- 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 flat regions. The accuracy of the inverse tool offset algorithm was dependent on the resolution of the discretized surface.
- 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
- Liner 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.

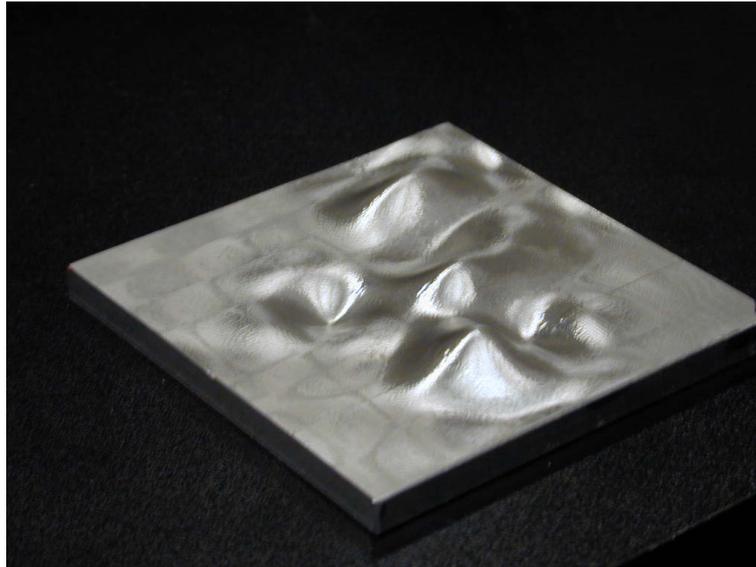


Figure 9: Machined surface using 0.25in diameter cutter

10 Acknowledgments

This work was funded by NSF grant # DMI-0070128 and was done in collaboration with Prof. Jami Shah and his students at Arizona State University. We would also like to thank Profs. Carlo Sequin, Profs. Sara McMains, Dr. Ganping Sun, Dr. V. Sundararajan, Palani Kumaresan and Dr. Roshan D.Souza. for their support through out the work.

References

- 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 Engineering Manufacture* (v217, nB3), pp335–348.
- Chih-Ching, L. and Lin, R.-S. (2001). "An improved method for scheduling the tool paths for three-axis surface machining." *International Journal of Machine Tools and Manufacture* (v41), pp 133–147.
- Choi, B. and Jerrard, R. (1998). *Sculptured Surface Machining: Theory and Applications*. Kluwer Academic Publishers, The Netherlands, First edition.
- Choi, B. K.; Kim, D. H.; and Jerrard, R. B. (1997). "C-space approach to tool-path generation for die and mold machining." *Computer Aided Design* (v29, n9), pp657–669.
- Feng, H.-Y. and Li, H. (2002). "Constant scallop height tool path generation for three-axis sculptured surface machining." *Computer Aided Design* (v34), pp647–654.
- Huang, Y. and Oliver, H. J. (1992). "Non-constant parameter NC tool path generation on sculptured surfaces." *ASME Journal of Computer in Engineering* (v1), pp411–418.
- Hwang, J. (1992). "Interference-free Tool-path in NC Machining of Parametric Compound Surfaces."

- Computer Aided Design* (v24, n12), pp667–677.
- Hwang, J. S. and Chang, T. (1998). “Three-axis machining of compound surfaces using flat and filleted endmills.” *Computer Aided Design* (v30, n8), pp641–647.
- Kim, B. H. and Choi, B. (2000). “Guide surface based tool path generation in 3-axis milling:an extension of guide plane method.” *Computer Aided Design* (v32), pp191–199.
- Kim, K. I. and Kim, K. (1995). “A new machine strategy for sculptured surfaces using offset surface.” *International Journal of Production Research* (v33, n6), pp1583–1697.
- 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.
- Lee, K.; Kim, J. K.; and Hong, S. E. (1994). “Generation of toolpath with selection of proper tools for rough cutting process.” *Computer Aided Design* (v26, n11), pp822–831.
- 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, Rensselaerville, NY, USA, 27 September - 1 October, (B8 of IFIP Transactions)*, pp 251–265.
- Sarma, R. and Dutta, D. (1997). “The Geometry and Generation of NC Tool Paths.” *Journal of Mechanical Design* (v119), pp253–258.
- Sarma, S. (1999). “The crossing Function and its application to zig-zag tool paths.” *Computer Aided Design* (v31, n14), pp881–890.
- 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), pp369–372.
- Suh, Y. S. and Lee, K. (1990). “NC Milling Tool-Path Generation For Arbitrary Pockets Defined By Sculptured Surfaces.” *Computer Aided Design* (v22, n5), pp273–283.
- Suresh, K. and Yang, D. C. H. (1994). “Constant Scallop Height Machining of Free-form Surfaces.” *Journal of Engineering for Industry* (v116), pp253–259.
- 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).