Proceedings of IMECE'03
2003 ASME International Mechanical Engineering Congress
Washington, D.C., November 15–21, 2003

IMECE2003-42489

HAPTIC SCULPTING AND 5-AXIS PENCIL-CUT PLANNING IN VIRTUAL PROTOTYPING AND MANUFACTURING

Weihang Zhu and **Yuan-Shin Lee***

Department of Industrial Engineering
North Carolina State University
Raleigh, NC 27695-7906
U. S. A.
yslee@eos.ncsu.edu

ABSTRACT

In this paper, a *Two-phase approach* to tool collision detection and local gouging elimination is proposed for haptic pencil-cut of sculptured surfaces. *Pencil-cut* is a special kind of machining operation, whose purpose is to use relatively smaller tools to remove rest material on the corners or highly curved regions that are inaccessible by bigger tools. Tool orientation determination and tool collision avoidance are critical issues for 5-axis *pencil-cut* tool path planning. Detailed techniques of *haptic rendering* and *tool interference avoidance* are discussed for haptic-aided 5-axis *pencil-cut* tool path generation. Hardware and software implementation of the haptic pencil-cut system with practical examples are also presented in this paper.

1. INTRODUCTION

Multi-axis machining of complex surfaces has attracted great attention in Computer-aided Manufacturing (CAM) and NC machining. There are many issues of 5-axis machining to be addressed, among them the tool path geometry and tool orientation are two main issues. Great progress has been made in 5-axis tool path planning after years of research [Choi 1998, Jun 2003, Lee 1998]. However, till now, no one can completely solve all the problems in 5-axis tool path planning. While continuing work is being pursued to improve 5-axis tool path planning, haptic interface has gradually aroused the interest of CAD/CAM (Computer-aided Design / Computer-aided Manufacturing) research area. Haptics is concerned with information and object manipulation through touch. Besides transducing position and motion commands from the user, these devices can present controlled forces to the user, allowing him or her to feel virtual objects and to control or deform the objects [Biggs 2002]. Haptic interface has found its applications in design, medicine, entertainment, education, industry, graphic arts, *etc* [Srinivasan 1997, Balijepalli 2002]. In this paper, we are especially interested in the haptic application in CAD/CAM and NC machining.

In the computer-aided design area, researchers from University of Utah have published their work on direct manipulating of NURBS with their special haptic manipulator [Thompson 1997]. Basically it is a system wherein one can trace along a NURB surface and feel it. A dynamic sculpting system for free-form subdivision solids was developed in SUNY Stony Brook [McDonnell 2000]. A virtual carving system with the commercial haptic interface was developed at University of Missouri-Rolla [Leu 2002]. These presented systems, although different in their implementations and some underlying theories, can both be traced back to volume sculpting [Wang 1995].

In the computer-aided manufacturing area, some initial attempts and inspiring work have been done at MIT [Ho 2001]. They have produced some interesting results in 5-axis tool path generation. In their work, a quick collision detection method was proposed between a tool and an environment represented by point clouds. In their 5-axis tool path generation application, they machined a part with constant Z-height machining method [Balasubra 2002].

This paper presents the technique of employing haptic interface in the 5-axis *pencil-cut* tool path planning. A lab-built 5-DOF (degree of freedom) haptic device is used as the haptic interface between a human and a virtual environment.

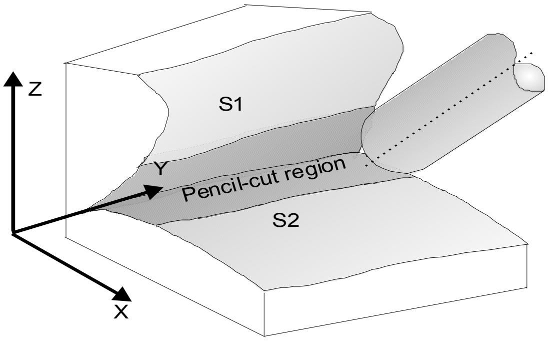


Fig. 1. *Five-axis pencil-cut*

2. PROBLEMS OF PENCIL-CUT MACHINING OF SCULPTURE SURFACE

In sculpture surface machining, there are several tasks in tool path planning, which usually includes roughing, semi-finishing, finishing, clean-up, and so on [Choi 1998]. The result of roughing gives us the rough shape of the product we want to make. After semi-finishing and finishing, the shape of the machined stock is almost similar to the designed part surface. Clean-up machining cleans those rest material left at the sharp corners or edges. *Pencil-cut* is kind of clean-up machining (Figure 1). *Pencil-cut* in 3-axis machining has been researched for years and this function is provided in some commercial software for parametric surface. Algorithms for 3-axis *pencil-cut* of polyhedral models have been proposed by our lab, presented earlier in [Ren 2002]. Work on 5-axis *pencil-cut*, however, has not been reported so far, to the best of our knowledge. Five-axis *pencil-cut* has the obvious advantage over 3-axis *pencil-cut*, due to its better accessibility. For 3-axis *pencil-cut*, the tool orientation is fixed as the Z-axis. Five-axis is especially useful for cleaning rest materials in some complex parts. As can be seen from Figure 1, this edge can be accessed by a 5-axis machine, but not accessible by a 3-axis machine. For 5-axis *pencil-cut*, the difficulty for planning a tool path lies mostly in the tool orientation determination.

The determination of tool orientation solely by computation usually works in the way described as follows. In 5-axis machining, a cutting tool can be oriented by rotating around one axis with an inclination angle λ_L and another axis with a tilt angle ω_L [Lee 1995]. To facilitate searching for a feasible tool orientation, usually a local coordinate system is set up as follows: X_L axis is the instantaneous feed direction (cutting direction); Z_L axis is the surface normal direction at the current CC (cutter contact) point; Y_L axis is determined by the right-hand rule [Lee 1998]. Then the tool orientations are searched based on this local coordinate system. A *C-space* (configuration space) is formed with the inclination angle λ_L and the tilt angle ω_L [Choi 1998, Jun 2003].

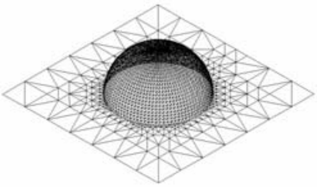
Past research on the tool orientation mainly adjusts the inclination angle λ_L, due to the theoretically infinite combinations of inclination angle λ_L and tilt angle ω_L. Most of the research only considers the local optimal tool orientation, which is based on analysis of the part surface property and cutting conditions near the CC point. When seeking for the globally optimized tool orientation, we have to deal with expensive computation concerning the interference between the whole tool assembly with the stock. The reason is that we have to find the feasible (not necessarily the optimal) tool orientation by trial and error. There are several theoretical research attempts in circumventing this exhaustive search [Lee 1995]. However, there are also some limitations in the results, due to the different assumptions made on the designed part surface or the cutting tool [Jun 2003].

In this paper, we developed a haptic system to help determine and select the tool orientations in 5-axis *pencil-cut* machining. Details of developing the haptic system are discussed in the following sections.

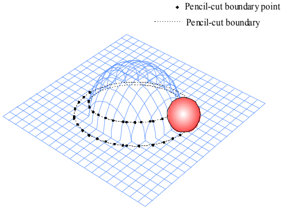
3. DEVELOPING HAPTIC SYSTEM FOR 5-AXIS PENCIL-CUT MACHINING

3.1 Identification of pencil-cut region and tool path generation

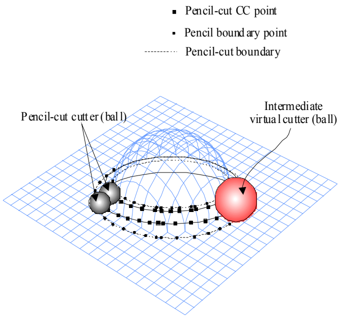
The first step for *pencil-cut* machining is to identify where the *pencil-cut* operation should be executed. In the developed haptic system for 5-axis *pencil-cut* tool path planning, the designed surface is represented in STL (stereo lithography) format. The use of STL (stereo lithography) format for representing a CAD model has been widely accepted in industry for some time [Koc 2000]. An STL model (see Figure 2(a)) is composed of a collection of triangles and basically is a kind of polyhedral model.



(a) an illustrative model for pencil-cut



(b) Identifying pencil-cut region



(c) Tracing pencil-cut tool paths with 2 smaller balls

Fig. 2. *Rolling ball method for identifying pencil-cut regions of sculptured surfaces* [Ren 2002]

As shown in Figure 2, the *Rolling-ball method* developed in our earlier work presented in [Lee 2000, Ren 2002] was revised to extract the *pencil-cut* regions of sculptured surfaces. The Rolling-ball method works in the way described as follows. Figure 2(a) shows an example polyhedral model. A CC-net, which is a net of CC (cutter contact) points, is cast on the polyhedral model, as shown in Figure 2(b). A ball with an appropriate radius, which is usually set as the radius of finishing cutting tool, rolls along this CC-net to identify the *pencil-cut* boundary points [Lee 2000]. Two smaller balls, whose radii are usually set as half of the radius of the previous big reference ball, are rolling along a route parallel to the *pencil-cut* boundary, as shown in Figure 2(c). In this way, the *pencil-cut* regions can be identified and the parallel *pencil-cut* tool paths can be traced for each *pencil-cut* region. In 3-axis *pencil-cut*, since the tool orientation is fixed as Z axis, the work up to here has almost generated the tool path. However, for 5-axis *pencil-cut*, the tool orientations are to be determined, by considering both the local surface property and global machining environments. Problems of global tool collision and local gouging in 5-axis machining, as shown in Figure 3, still need to be solved. More detailed discussion of solving these problems will be discussed later in Section 4.

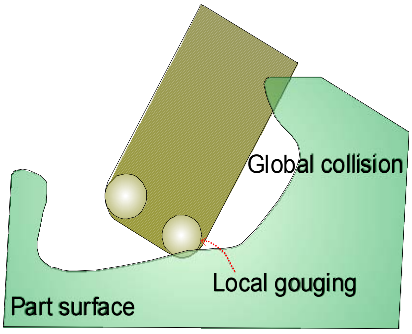


Fig.3. *Global collision & local gouging in 5-axis machining*

3.2 Developing haptic interface for 5-axis pencil-cut tool path planning

After the *pencil-cut* tool locations are found, the feasible 5-axis tool orientation is determined by using haptic interface. As shown in Figure 4, the haptic device we are using is a lab built 5-DOF (degree of freedom) pen-based electro-mechanical device made by Suzuki, *Inc.*, Japan. It can detect 6-DOF of haptic probe movement and gives 5-DOF force feedback, with both force and torque feedback. Figure 4 shows the lab setup with the haptic device located in the middle. In Figure 4, the left hand side is its controller. Its right hand side is the dual-CPU 2.4GHz workstation with the implemented software and rendering programs presented in this paper.

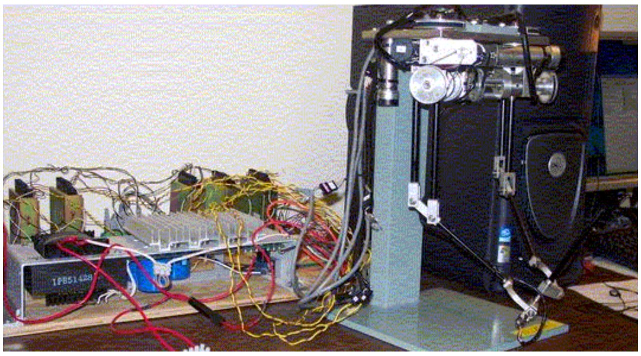


Fig.4. *Lab setup of haptic device controller and 5-DOF haptic device, with a dual-CPU workstation*

The controller for this haptic device was made in the lab. Figure 5 shows the working principle of our haptic system. The movement of the haptic device probe is recorded by the encoders. The encoder's signals are collected by a counter board, which is installed in a dual-CPU workstation. As shown in Figure 5, the program computes the haptic response of force feedback and sent the control signals out via D/A board. The control signals are amplified to drive the motors on the haptic device. The motors' movements let the user feel the force feedback.

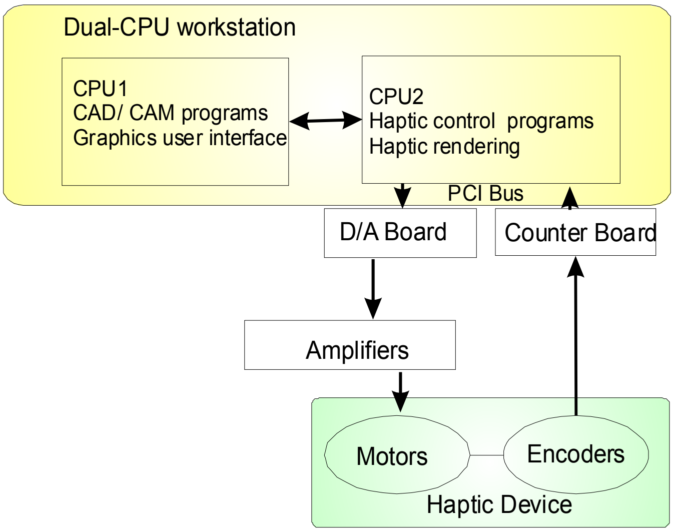


Fig.5. *Controlling principle of the haptic system*

The driver for the haptic system was developed for this hardware. The tasks of this driver include interfacing with D/A board and counter board and distributing the desired forces to haptic device's actuators' (DC motors) torques. The haptic device has a left-manipulator and a right one. Both of them working together feed back the force and torque to the user. Figure 6 shows the corresponding force and torque configuration in the virtual tool assembly with the physical haptic device probe. In the virtual tool assembly, the pivot point is defined as rotation center of a NC machine spindle head, as shown in Figure 6. In the physical haptic device, the pivot point is the middle point between the two ends of the left and right manipulators (Figure 6). The pivot point is used for calculating the torque required in the virtual world. The calculation details for the required force and torque of one instance during the haptic interaction process will be presented in next section.

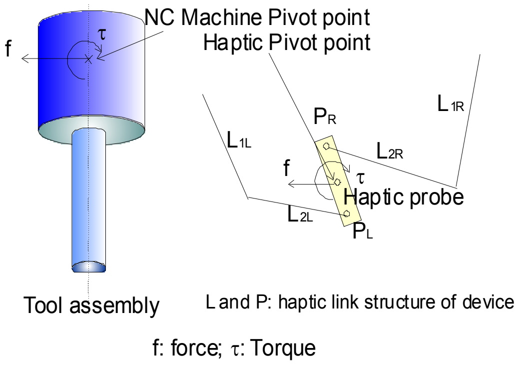


Fig.6. *The transformation between the tool assembly and the physical haptic system link structure*

Assume that from the virtual reality calculation, it is desired that the left manipulator needs to generate force f_L and the right manipulator needs to generate force f_R, then the necessary torques τ_L and τ_R generated from the motors can be calculated by the Jacobian Matrices (J_L^T and J_R^T), as they have the following relation (Figure 6):

$$\tau_L = J_L^T f_L \tag{1}$$

$$\tau_R = J_R^T f_R \tag{2}$$

The derivation of the Jacobian Matrix is dependent on the link structure of a specific haptic device, which is a kind of special robot arms [Craig 1989]. These two equations will be referred later in this paper. The details of force and torque feedback calculation for this haptic interface are to be presented as part of haptic rendering in Section 4.

3.3 Utilizing haptic interface to help determine the tool orientation

Generally in 5-axis NC machining, the interference between the workpiece system and the tool assembly can be divided into two types (see Figure 3):

(1) Global collision between the workpiece, fixtures and the non-cutting portion of the tool assembly, e.g. tool holder and tool shank;

(2) Local gouging between the designed part surface and cutting portion of the tool.

The user needs to avoid both of them when choosing a feasible 5-axis tool orientation. During the haptic interaction process, it is quite easy for the user to orient the tool assembly to avoid *global collision*, as one can feel the force and torque feedback from the haptic system in the case of tool collision. On the other hand, *local gouging* is relatively difficult to be completely eliminated. During the haptic interaction process, the virtual tool touches the surfaces, which means there must be at least a slight interference between the tool and the designed surface. Hence *global collision* is removed during the haptic interaction. Post-processing for the tool path is pursued to eliminate *local gouging*. Details on how interferences are detected and corrected are to be presented in Section 4 and illustrated examples are presented in Section 5.

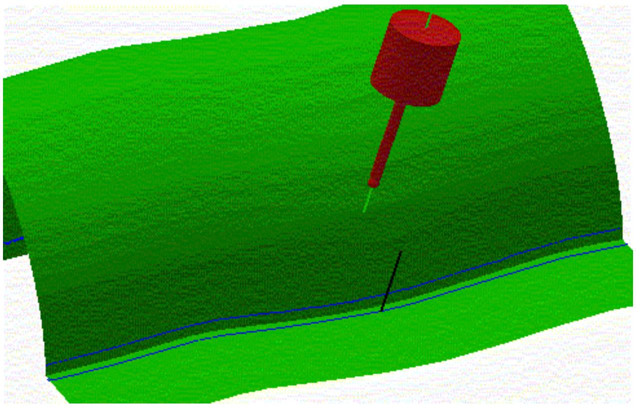


Fig.7. *Pencil-cut tool path (two blue parallel paths) identified on a test surface, with the nearest CC point checked out and highlighted with its normal vector*

Figure 7 shows an example surface designed for demonstrating *pencil-cut* tool path planning. Two parallel *pencil-cut* tool paths were identified along the edges. During the haptic interaction process, the nearest CC point to the current virtual tool tip is found and highlighted with its normal vector, as shown in Figure 7. To speed up the computation and to reduce interaction load of the haptic system, it is proposed that only the tool orientations of those critical CC points need to be specified and the other CC points' tool orientation are to be defined by interpolation. This is analogous to animation creation, where key animation frames are defined and interim animation frames are automatically created by interpolation. During the interaction process, if a certain CC point is picked up as a key point, the user moves the virtual tool to that CC point, orients the virtual tool along an appropriate orientation, based on what she/he sees and the force and torque she/he feels at the haptic probe, and presses a hot key to record the current tool orientation for this CC point. The selection of the critical tool orientations is based on the user's judgment. Figure 8 shows the interpolated tool orientations between the critical tool orientations. Assume two adjacent critical tool orientations are $\overrightarrow{O_1}$ and $\overrightarrow{O_2}$ and there are n CC points between them, the i-th tool orientation on of the i-th interim CC point is interpolated as (see Figure 8):

$$\overrightarrow{O_i} = i * (\overrightarrow{O_2} - \overrightarrow{O_1}) / (n+1) + \overrightarrow{O_1} \quad (3)$$

$$\overrightarrow{O_{i,unit}} = Normalize(\overrightarrow{O_i}) \quad (4)$$

A complex machining environment can be composed of a workpiece, jigs and fixtures, *etc*. The interference between the tool and all these components should be considered and thus affects the tool orientation selected. If the user is unsatisfied with the current results, she/he can choose to re-specify the critical tool orientation or adjust the interpolated tool orientations directly, until all the orientations are satisfying. This will be illustrated more clearly in the examples presented later in Section 5. The calculation of interference detection and force feedback is presented in the next section.

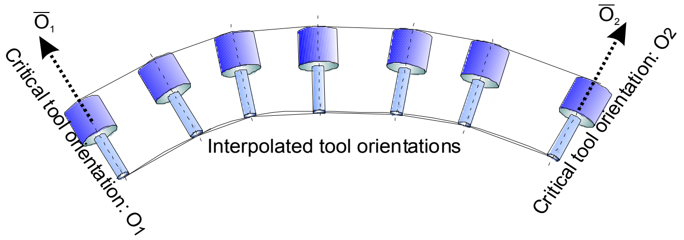


Fig.8. *Tool orientation interpolations from two critical tool orientations*

After the tool orientations are generated, the tool path information is post-processed to eliminate local gouging. In the post-processing, the tool orientation for each CC point is kept intact and the tool assembly is checked for local gouging. The tool is lifted up to a position along the current tool orientation to eliminate any local gouging. Figure 9 illustrates this process.

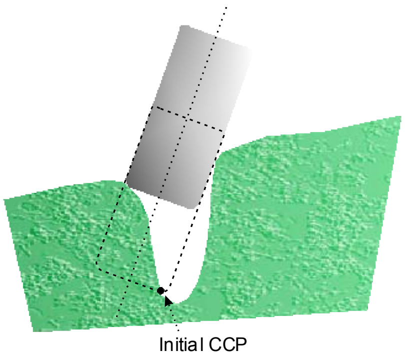


Fig.9. *Retracting cutter to avoid local gouging*

After the automated post-processing, the tool path can be generated by computing CL (cutter location) points $\overrightarrow{CL_i}$ from CC points $\overrightarrow{CC_i}$ with the following formula for filled-end mill (see Figure 10):

$$\overrightarrow{CL_i} = \overrightarrow{CC_i} + r_{tc} * \overrightarrow{n_i} + e * \frac{((\overrightarrow{n_i} \times \overrightarrow{O_i}) \times \overrightarrow{O_i})}{\left|((\overrightarrow{n_i} \times \overrightarrow{O_i}) \times \overrightarrow{O_i})\right|} + r_{tc} * \overrightarrow{O_i} \quad (5)$$

where r_{tc} is the minor radius of a fillet-end mill,
e is the major radius of a fillet-end mill,
$\overrightarrow{n_i}$ is the normal vector associated with the current CC point, $\overrightarrow{O_i}$ is the tool orientation for the current CC point.

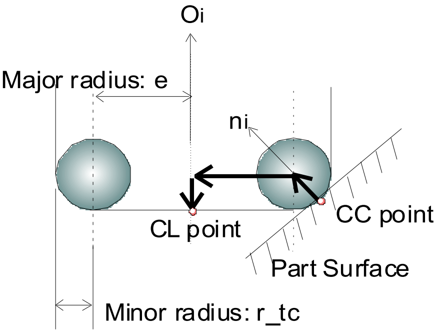


Fig.10. *Calculating the CL points from cutter contact (CC) points*

The CL data is further processed to generate NC code to drive the NC machine. In this paper, the generated NC code is simulated on commercial NC simulation and verification software.

4. TWO-PHASE APPROACH TO HAPTIC RENDERING PROGRAMMING FOR 5-AXIS PENCIL-CUT AND FORCE FEEDBACK

Haptic rendering software first checks the interference/collision (*including both global collision and local gouging*) between a moving tool assembly and the static product surface. If any interference is detected, the software continues to calculate the extent of interference and calculate the required force and torque in the virtual world. The desired force and torque are converted to haptic manipulator force f_L and f_R, as indicated in Equations (1) and (2), so that they can be used to drive the haptic device.

The collision detection speed is critical for haptic applications. The update rate is estimated to be at least 1,000 Hz in order to maintain a sustained feeling of force feedback. In every 1 millisecond, the computer needs to detect the coordinates of the haptic probe, transform the coordinate values from the physical device coordinate system to the graphics environment, detect the collisions between the virtual objects and calculate the force feedback. Object / object collision detection methods are required in the current application. In order to achieve high update rate for complex environments, computing techniques like the spatial decomposition tree [Klosowski 1998, McNeely 1999] and the OBB (object oriented bounding-box) tree [Gottschalk 1996] are used in this paper.

4.1 Two-phase approach of collision detection and force response

As shown in Figure 11, a *Two-phase rendering approach* has been proposed for the haptic pencil-cut interface and the rendering programming. In this Two-phase approach, for convenience of the first phase, the triangles in STL model are organized into an OBB (object oriented bounding-box) tree, as shown in the 'First phase' of Figure 11 [Zhu 2003]. For convenience of the second phase, the triangles in STL model are discretized into point clouds at the beginning of the haptic application, as shown in the 'Second phase' in Figure 11. Each point is associated with a normal direction, which is set as the normal direction of the triangle it belongs to. Adaptive point cloud density is used for different features of the tool assembly.

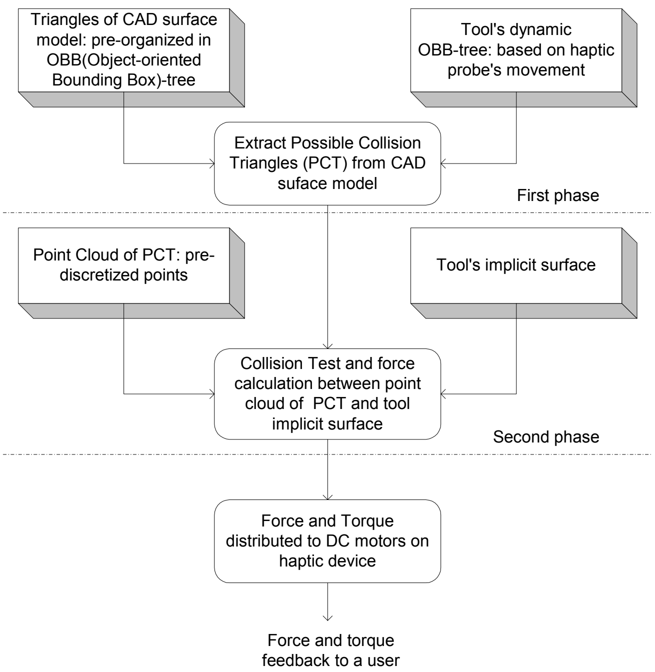


Fig.11. *Two phase approach for haptic rendering*

In the first phase (Figure 11), the possible collision triangles (PCT) are extracted by checking the interference between the OBB-tree of the STL surface model and the simple OBB-tree of the tool assembly. In our approach, the tool definition is a combination of implicit surfaces. We would like to simplify the definition of the tool while trying to accommodate a more complex part surface. Triangulated model in STL file format is adopted as the representation of CAD surface model as the computation on STL model is relatively faster. Besides, programs developed for STL model machining could also be ported to the haptic application. The next step is to find out how we can quickly detect the collision and calculate the force feedback. STL surface model's OBB-tree is static during the interaction process. The tool assembly's OBB-tree is changing, corresponding to the movement of haptic probe [Zhu 2003]. As the tool assembly's OBB-tree is quite simple, there is not much overhead in updating it.

In the second phase (Figure 11), the points corresponding to the PCT (possible collision triangles) are checked against the tool's implicit surface for collision test [Zhu 2003]. If a point is inside the tool assembly, we find the nearest tool surface point to the collision point and then calculate the distance between the collision point to the corresponding tool surface point.

The advantages of the proposed *Two-phase rendering approach* are:

i) The OBB-tree used for the haptic application environment can speed up the computation efficiency significantly, especially when the sculptured surfaces become complex;

ii) The densities of the point clouds are adaptive for different features of a tool;

iii) The cutting tool's shape can be complex and a generalized fillet-end mill [Chiou 1999] is used as the general representation for different endmills of ball-end mills, flat-end mills, taper-end mills and fillet-end mills;

iv) The force and torque feedback calculation (will be discussed in next section) is tightly integrated with the hardware driver level.

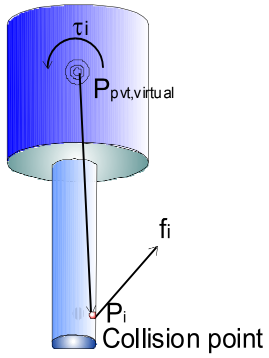


Fig.12. *Retracting cutter to avoid local gouging*

4.2 Force and torque feedback distribution to the haptic device hardware

The haptic rendering programs detect the possible gouging or collision and compute the force and torque response through the haptic interface system, as shown in Figure 11. The force magnitude is proportional to this distance. Assume this distance is Δx_i , k is a pre-defined coefficient, the force magnitude is calculated as follows [Zhu 2003]:

$$f_i = k * \Delta x_i \quad (6)$$

The direction of the force $f_{i,dir}$ is already associated with the collision point as in the process of generating point cloud. Assume the virtual pivot point's location is P_{pvt}, the collision point's location is P_i, and the force magnitude f_i is calculated with the previous equation, the torque τ_i induced by this collision point is calculated as follows (Figure 12),

$$\tau_i = f_i * (f_{i,dir} \times (P_i - P_{pvt})) \quad (7)$$

Assume there are totally m collision points in the current instance, the force and torque feedback are accumulated as follows [Zhu 2003]:

$$f = \sum_{i=1}^{m} f_i = k \sum_{i=1}^{m} \Delta x_i \quad (8)$$

$$\tau = \sum_{i=1}^{m} \tau_i = k \sum_{i=1}^{m} (\Delta x_i * (f_{i,dir} \times (P_i - P_{pvt}))) \quad (9)$$

Assume that in Figures 12 and 13, the vector from one haptic manipulator end P_L to the other end P_R is r_{LR}, the torque is distributed to two manipulators as follows [Zhu 2003]:

$$f_{Ln} = \frac{\tau}{|r_{LR}|} * \frac{\tau \times r_{LR}}{|\tau \times r_{LR}|} = -f_{Rn} \quad (10)$$

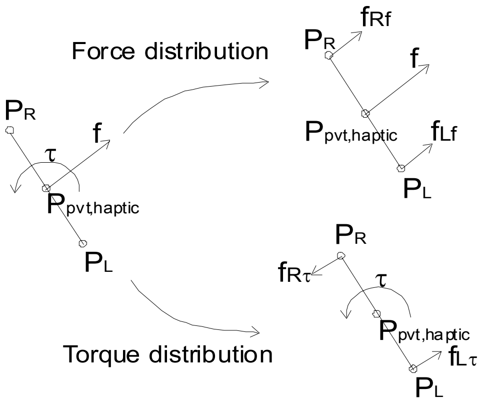


Fig.13. *Force and torque distribution to two haptic manipulators*

Then the desired forces on the left and right manipulators are calculated as follows:

$$f_L = f_{Lf} + f_{Ln} \quad (11)$$

$$f_R = f_{Rf} + f_{Rn} \quad (12)$$

The desired forces f_L and f_R are then substituted into Equations (1) and (2) to get the corresponding DC motor torques. The rotation of the DC motors applies force on the user's hand through the haptic device structure.

Fig.14. *A view of operation process of the 5-axis tool path planning*

5. IMPLEMENTATION AND EXAMPLES

The proposed techniques and the haptic hardware have been implanted at our lab (see Figure 14). Based on the developed haptic device controller system, the haptic rendering programs and the software driver have been developed to this haptic system. The whole haptic system was implemented on a dual 2.4GHz CPU workstation, with Visual C++ and OpenGL®. Interaction scheme is designed for the haptic application. Since we constructed the haptic device controller and developed the programs from the hardware level, we own the greatest flexibility in designing our specific haptic applications.

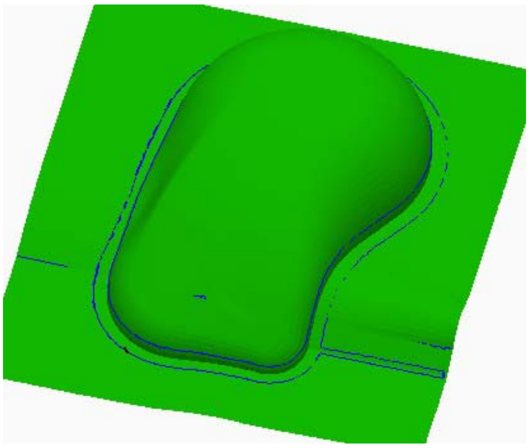


Fig.15. *Pencil-cut tool path (dark blue lines) identified on a mouse surface model*

Figure 14 shows the overview of the operation of the haptic 5-axis *pencil-cut* system. The user's left hand is gripping the haptic probe and the right hand is using the mouse and keyboard. Figure 15 shows an example part of a computer mouse model in STL format. Two parallel *pencil-cut* tool paths are identified along the sharp edges by using the *Rolling ball* *method* developed in our earlier work in [Lee 2000, Ren 2002]. Figure 16 shows some 5-axis tool orientations generated for machining of the pencil-cut critical regions of the example part.

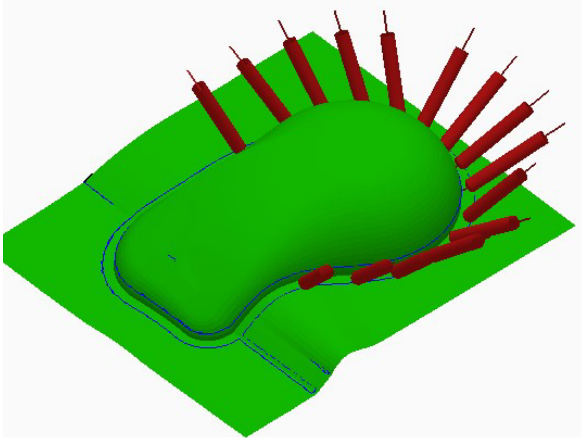
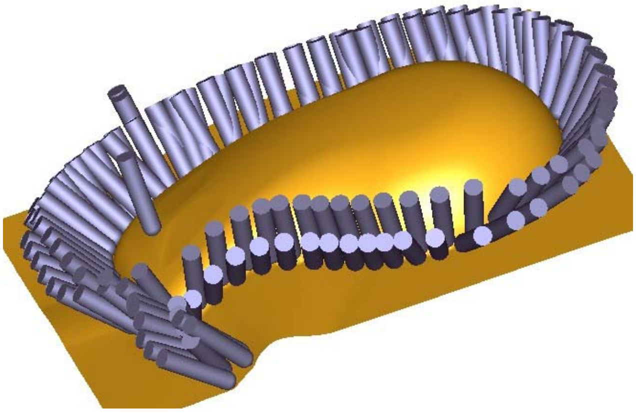
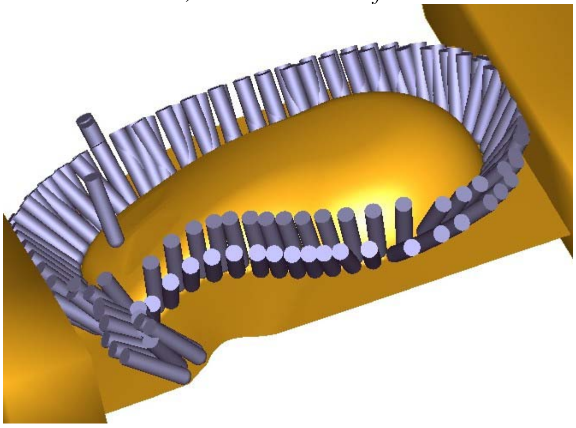


Fig.16. *Tool orientations selected for pencil-cut of the example CAD model*



(a). *Tool orientations selected for pencil-cut tool on the mouse model, when there is no fixtures*



(b). *The selected tool orientations in (a) actually have global collision with fixtures: the tool penetrates into the fixtures virtually*

Fig.17. *Tool orientations selected without considering the machining environment may collide with the fixtures*

Figure 17 (a) shows the generated 5-axis pencil-cut tool paths for machining the critical regions on the example part surface. The tool holders are temporarily hidden in Figure 17 to show the tool orientations more clearly. Without considering the surrounding environment, 5-axis tool paths could easily collide with the adjacent object unintentionally. It can be seen from Figure 17(b) that the generated 5-axis tool paths are actually colliding with the fixtures nearby during 5-axis tool motions. As shown in Figure 17(b), these tools actually penetrate into the fixtures. By taking the whole machining environments into account, the haptic system responds and enables the user feel the collision force feedback and let the user correct the tool orientations by using the haptic interface. When one feels the force and torque feedback and sees the movement of tool, she/he can orient the tool to avoid the global collision with ease.

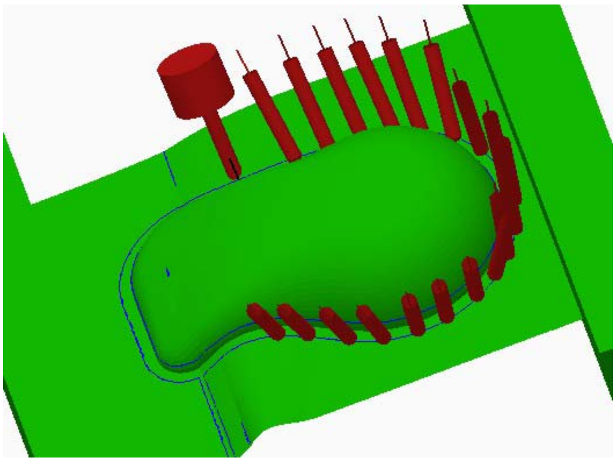


Fig.18. *Corrected tool orientation selection in a complex machining environment*

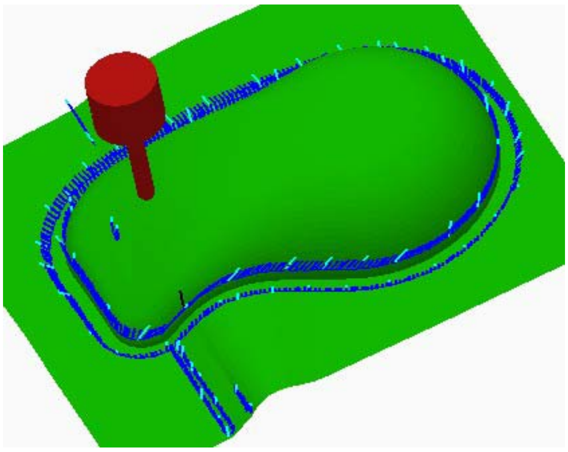


Fig.19. *Re-computing pencil-cut tool orientations after correction of tool collisions*

Figure 18 shows the correct *pencil-cut* tool orientations generated with the haptic interface system. Figure 19 shows the re-computation of the corrected tool orientations for 5-axis pencil-cut after the global collisions have been eliminated by the haptic system. Local gouging is further eliminated by the automated post-processing of the tool paths, as described earlier in Section 3. In Figure 19, the vectors in cyan color (light color in grayscale print-out) represent those critical tool orientations specified by the user during the haptic interaction. In Figure 19, the vectors in blue color (dark color in grayscale print-out) represent those interpolated tool orientations.

Figure 20 shows the generated tool paths and the corrected tool orientations are also rendered in a view where sample tools are displayed with both the example part surface and the surrounding fixture. In Figure 20, it shows how the tool orientations are changing along the *pencil-cut* tool paths without colliding with the surrounding workholders.

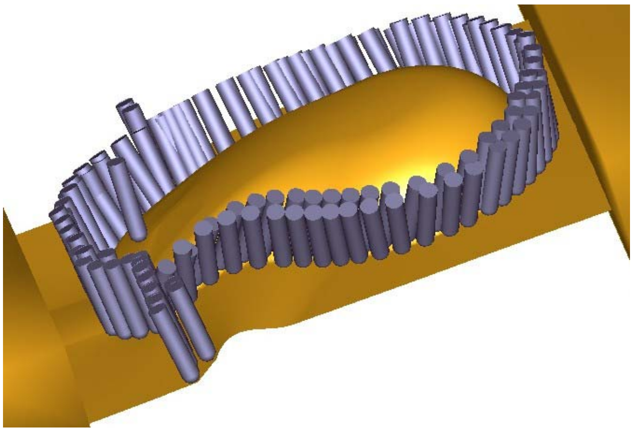


Fig.20. *Corrected tool orientation selection in a complex machining environment*

6. CONCLUSIONS AND FUTURE RESEARCH

In this paper, our work on developing haptic 5-axis pencil-cut system is presented. A Two-phase rendering approach has been proposed for haptic 5-axis *pencil-cut* of complex sculptured surfaces. The presented techniques enable the haptic device be utilized to help determine feasible 5-axis tool orientations in a complex machining environment. The haptic interaction and computation process are elaborated in the paper. As the underpinning technology, haptic interface's development is described concerning both the hardware and software level. A haptic rendering method is proposed and implemented for this special haptic application.

ACKNOWLEDGMENTS

This work was partially supported by the National Science Foundation (NSF) *CAREER Award* (DMI-9702374), NSF Grant (DMI-0300297) and the Army Research Office (Grant #DAAG55-98-D-0003) to Dr. Y. S. Lee at North Carolina State University. Their support is greatly appreciated.

The authors would also like to thank Dr Y. Adachi at Suzuki *Inc*. (Japan), Dr S. Sarma at MIT, and Dr. Y. Ren at NCSU for their helpful discussion and suggestions.

REFERENCES

[Balijepalli 2002]	Balijepalli, A., Kesavadas, T. An experimental haptic based skill training simulator for manufacturing applications, ASME Winter Conference, Nov. 2002, New Orleans, LA
[Balasubra 2002]	Balasubramaniam M, Ho Stephen, Sarma S, Adachi Y, Generation of collision-free 5-axis tool paths using a haptic surface, *Computer-Aided Design* 34 (2002) 267-279
[Biggs 2002]	Biggs, J. and Srinivasan M. A. Haptic Interfaces, in *Handbook of Virtual Environments*, edited by Kay Stanney, Chapter 5, Lawrence Earlbaum, Inc., London, 2002
[Chiou 1999]	Chiou C.J. and Lee. Y.-S, A shape-generating approach for multi-axis machining G-buffer models, *Computer-Aided Design*, 31 (1999) 761-776
[Chiou 2002]	Chiou C.J. and Lee. Y.-S, Swept Surface Determination for Five-Axis Numerical Control Machining, *International Journal of Machine Tools and Manufacture*, 42 (2002) 1497-1507
[Choi 1998]	Choi B.K. and Jerard R.B., *Sculptured surface machining*, Kluwer Academic Publishers, 1998
[Craig 1989]	Craig J.J. Introduction to Robotics: Mechanics and Control, Addison-Wesley Publishing Company, *Inc*, 1989
[Gottschalk 1996]	Gottschalk, S., Lin M.C., and Manocha D., OBBTree: A Hierarchical Structure for Rapid Interference Detection, *Computer Graphics (SIGGRAPH'96 Proceedings)*, August, 1996, 171-180
[Ho 2001]	Ho S., Sarma S., Adachi Y., Realtime interference analysis between a tool and an environment, *Computer-Aided Design* 33 (2001) 935-947
[Jun 2003]	Jun C.-S., Cha K and Lee Y.-S., Optimizing tool orientation for 5-axis machining by configuration-space search method, *Computer-Aided Design*, 35 (2003) 549-566
[Klosowski 1998]	Klosowski J.T., Held M., Mitchell J.S.B., Sowizral H., Zikan K., Efficient Collision Detection Using Bounding Volume Hierarchies of k-DOPs, *IEEE Transactions on Visualization and Computer Graphics*,

Vol. 4, No. 1, 1998, 21-36

[Koc 2000] Koc K., Ma Y., and Lee Y.-S., Smoothing STL Files by Max-Fit Biarcs Curves for Rapid Prototyping, *Rapid Prototyping Journal*, 6(3) (2000) 186-204

[Lee 1995] Lee Y.-S, Two-Phase approach to global tool interference avoidance in 5-axis machining, *Computer-Aided Design*, 27 (1995), 715-729

[Lee 1998] Lee Y.-S, Mathematical modeling using different endmills and tool placement problems for 4- and 5-axis NC complex surface machining, *International Journal of Production Research*, 1998, Vol. 36, No.3, 785-814

[Lee 2000] Lee Y.-S., Ma Y., and Jegadesh G., Rolling-Ball Method and contour marching approach to identifying critical regions for complex surface machining, *Computer In Industry*, Vol. 41, No. 2, 2000, 163-180

[Leu 2002] Leu M.C., Velivelli A. Peng X., Creating freeform model by carving virtual workpiece with haptic interface, Proceedings of ASME IMECE (Intl. Mech. Engineering Congress and Exposition) Nov. 17-22, 2002, New Orleans, Louisiana, 2002-32467

[McDonnell 2000] McDonnell K.T. and Qin H., Dynamic sculpting and animation of free-form subdivision solids, *Proceedings of IEEE Computer Animation*, Philadelphia, Pennsylvania, May 2000, 126-133

[McNeely 1999] McNeely W.A., Puterbaugh K.D., Troy J.J, Six degree-of-freedom rendering using voxel sampling, SIGGRAPH 99, 401-408

[Ren 2002] Ren Y., Lee Y.-S. and Yau H.T., Contraction tool method and clean-up tool path generation for machining polynomial models, (in revision), submitted to *Computer In Industry*, 2002

[Srinivasan 1997] Srinivasan, Mandayam A. and Basdogan, Cagatay, Haptics in virtual environments: taxonomy, research status, and challenges, *Comput. & Graphics*, Vol. 21, No. 4, 1997, 393-404

[Thompson 1997] Thompson II T.V., Johnson D.E. and Cohen E., Direct Haptic Rendering of Sculptured Models, *Proceedings Symposium on Interactive 3D Graphics*, Providence, RI, April 27-30, 1997, 1-10

[Wang 1995] Wang W. S. and Kaufman A.E., Volume Sculpting, ACM Symposium on Interactive 3D Graphics, Monterey CA USA, 1995, 151-156 and 214

[Zhu 2003] Zhu W. and Lee Y.-S., Five-axis Pencil-Cut Machining Planning and Virtual Prototyping with a 5-DOF Haptic Interface, (in review) submitted to *Computer-Aided Design*, 2003.

AUTHORS' BIOGRAPHIC SKETCH

Weihang Zhu is a PhD candidate in the Department of Industrial Engineering, North Carolina State University. He received his B.S. and M.S., both in Energy Engineering, from Zhejiang University, China, in 1997 and 2000 respectively. His research interests include CAD/CAM, multi-axis NC surface machining, polyhedral machining, applied computational geometry, discrete computer simulation, and haptics.

Yuan-Shin Lee is Professor of Industrial Engineering at North Carolina State University, USA. He received his Ph.D. (1993) and MS (1990) degrees from Purdue University, USA, both in industrial engineering, and his BS degree from National Taiwan University, Taiwan, in mechanical engineering. His research interests include 3- and 5-axis sculptured surface manufacturing, CAD/CAM, rapid prototyping, high speed machining, and computational geometry for design and manufacturing. He is a registered professional engineer (PE) in mechanical engineering. He is also a certified manufacturing engineer in system integration and control. Dr. Lee received the National Science Foundation (NSF) *CAREER Award*. He also received the 1997 *Outstanding Young Manufacturing Engineer Award* from the Society of Manufacturing Engineers (SME), the 1998 *Norman Dudley Award* from the Taylor & Francis Journals, London, U.K., the 1999 *Anderson Outstanding Faculty Award* and the 2000 *Alumni Faculty Outstanding Teaching Award* from North Carolina State University, and the 2001 *ALCOA Foundation Engineering Research Achievement Award*. He serves as the Associate Editor of the *Journal of Manufacturing Systems* (JMS).