Automated Machinability Checking for Sculptured Surface Manufacture

Kwangsoo Kim and Byungchul Ko
Department of Industrial Engineering
Pohang Institute of Science and Technology

ABSTRACT

Determining tool-approach directions is an important issue when an effort is made to transfer CAD data into manufacturing automatically. An algorithm is developed to determine whether a given part can be machined on a three-axis milling machine. If a set of feasible tool-approach directions exists for a sculptured surface, the NC tool path and G-codes for machining the surface on a three-axis milling machine can be generated automatically by an NC tool path generation algorithm. The algorithm can be used for orientation and fixturing of the workpiece for interference free machining. The algorithm can also be applied to checking the translational separability of polyhedral parts in automatic assembly.

1. Introduction

A major area of interest in the applications of CAD/CAM is in the modeling and manufacturing of sculptured surfaces. Sculptured surfaces are mainly encountered in aerospace, ship building, and automobile industries, as well as in the production of consumer goods such as plastic articles, glassware and ceramics. Sculptured surfaces have free-form and non-analytical contours so they cannot be represented with simple mathematical equations. For this reason, the manufacturing of sculptured surfaces has always presented difficulties to manufacturing engineers.

Automatic NC code generation is an important issue that has gained attention in research in the integration of sculptured surface design and manufacturing. To generate NC code automatically from CAD data, it is necessary to analyze sculptured surfaces to determine whether or not a part surface is machinable on a three-axis milling machine (Kim, 1990). The problem of machinability checking can be solved by finding all the feasible directions from which a sculptured surface can be approached for machining (Joshi and Chang, 1986; Tseng and Joshi, 1991). This direction is called the tool-approach direction. The tool-approach direction in 3-axis machining is used to determine the part orientation so that the sculptured surface can be presented in a proper direction to the cutting tool for fixturing and machining (Su and Mukerjee, 1991). When the tool axis is aligned with the tool-approach direction, no interference occurs between the tool and the part surfaces that are not currently being machined.

In this paper, an algorithm is proposed for checking machinability of three-dimensional sculptured surfaces. The algorithm calculates all the feasible tool-approach directions for sculptured surfaces. In the algorithm, a sculptured surface is first approximated with smaller subpatches by an adaptive subdivision method (Schmitt *et al.*, 1986). Then, the set of feasible tool-approach directions for the subpatches is calculated.

In Section 2, the subdivision method is explained. The algorithm for finding tool-approach directions is explained in Section 3. The implementation of the machinability checking is presented in Section 4. Conclusions are discussed in Section 5.

2. Subdivision of Sculptured Surfaces

This section describes how the surface is subdivided to generate the approximating subpatches so that the deviation from the true surface is less than a user-defined approximation error or tolerance. In the subdivision method, a sculptured surface is first approximated with four subpatches. An accuracy metric is then used to measure the closeness of the approximating subpatches to the actual surface. If the approximation is not within a user-specified tolerance, the subpatch is subdivided into four smaller subpatches. The process is then recursively performed on these subpatches until the set of patches approximates the given sculptured surface within the specified tolerance.

2.1 Bezier patch subdivision

A Bezier patch is easily subdivided into four subpatches as discussed below. The subdivision procedure will be explained by using a cubic case as an example. The same holds true for other than cubic case as well. Let us first consider the problem of subdividing a cubic Bezier curve $r(t)$ at $t=½$. Let $r_a(t)$ and $r_b(t)$ denote the two subcurves corresponding to the parameter ranges $t \in [0, ½]$ and $t \in [½, 1]$, respectively. From the de Casteljau algorithm, then, the control vertices of $r_a(t)$ are given by

$$V_{0a} = V_0$$
$$V_{1a} = (V_0 + V_1) / 2$$
$$V_{2a} = (V_0 + 2 V_1 + V_2) / 4, \text{ and}$$
$$V_{3a} = (V_0 + 3 V_1 + 3 V_2 + V_3) / 8.$$

The control vertices of the second patch $r_b(t)$ are also expressed similarly.

A Bezier patch having control vertices $\{V_{ij}\}$ is subdivided as follows:

1) Subdivided each row i of $\{V_{ij}\}$ as if it is a Bezier curve, and

2) Subdivided each column j of the subdivided control vertices.

The subdivision process is continued for each thick subpatch until it becomes thin enough to meet the user-defined tolerance. When the subdivision process is recursively continued, the resulting subpatches are concisely stored in quadtree data structure. An essential part of the subdivision algorithm is an efficient maintenance of the quadtree storing the subdivided patch information. After the completion of the adaptive subdivision process, all the thin subpatches are stored in leaf nodes of the quadtree.

2.2 Thickness of Bezier patches

Actually measuring the thickness of a bicubic Bezier patch requires analytical knowledge of the particular surface definition techniques. It may be computationally easier to get some conservative bound on the error (e.g., it is possible to use the convex hull property of Bezier patches to compute this bound). A method of estimating the thickness of a bicubic Bezier patch defined by a control vertex net $\{V_{ij}\}$ is as follows (Choi, 1991):

1) Calculate a center point V_m as an average of the corner vertices by
   $V_m = (V_{00} + V_{03} + V_{30} + V_{33}) / 4$
2) Estimate a unit surface normal N by
   $N = \{(V_{33} - V_{00}) \times (V_{30} - V_{03})\} / |(V_{33} - V_{00}) \times (V_{30} - V_{03})|$
3) Estimate the thickness τ_1 of the patch by
   $\tau_1 = \max \{| N \cdot V_{ij} - N \cdot V_m |,\ i,j = 1,2\}.$

Actually, the thickness τ_1 represents the maximum deviation of the Bezier control net (not the surface) which should be larger than the thickness of the patch. A patch is called a thin patch if its thickness τ_1 is less than a specified tolerance.

2.3 B-spline surface subdivision

The same subdivision strategy is directly applicable to subdividing B-spline surfaces as proposed by Peng (1984). However, with the subdivision algorithm for Bezier patches on hand, a more practical strategy would be to convert each B-spline surface into a composite Bezier patch. The conversion process is easily carried out by using the knot insertion algorithm (Boehm, 1980; Boehm, 1981).

A non-uniform B-spline curve is defined by a sequence of control vertices $\{V_i\}$ and knot spans $\{\Delta_i\}$. A B-spline curve segment of degree 3, for example, is defined by four control vertices $\{V_i\ \ V_{i+1}\ V_{i+2}\ V_{i+3}\}$ and five knot spans $\{\Delta_{i-2}\ \Delta_{i-1}\ \Delta_i\ \Delta_{i+1}\ \Delta_{i+2}\}$. The Bezier points $\{B_i,\ i=0,...,3\}$ are obtained by subdividing the B-spline control polygon according to the ratios of knot spans. The Bezier points of a B-spline surface can be obtained by the same algorithm. It is first applied to each column of the B-spline polyhedron net, and then to each row of the intermediate net obtained (Farin, 1990).

3. Feasible Tool-Approach Directions

3.1 Definitions

The following definitions will be used throughout the paper.

Unit sphere: A unit sphere is a sphere centered at the origin of the coordi-nate system with a unit radius. The term sphere denotes the surface rather than the interior of the sphere in this paper.

B-plane: A plane perpendicular to a vector and passing through the origin is the b-plane of the vector.

M-hemisphere: A b-plane, normal to a vector, divides a unit sphere into two hemispheres. The m-hemisphere is the hemisphere which contains the normal vector.

M-polyhedron: The intersection of m-hemispheres is an m-polyhedron if it is not null.

3.2 Feasible Tool-Approach Directions

Suppose that a sculptured surface is subdivided into subpatches by the adaptive subdivision method explained in Section 2, and the unit normal vector to each subpatch is calculated. For each unit normal vector, an m-hemisphere is defined. Any point on the m-hemisphere defines a tool-approach direction vector for the corresponding subpatch. The intersection of the m-hemispheres for all the subpatches is an m-polyhedron. A set of points on the m-polyhedron is the set of feasible tool-approach directions for the sculptured surface.

The calculation procedure of the feasible tool-approach directions is a step-by-step process. First, the intersection of the m-hemispheres of the normal vectors to the first three subpatches is calculated. The intersection is an m-polyhedron. Then, the intersection is updated by calculating the intersection of the fourth m-hemisphere and the current m-polyhedron. This updating process is continued for the remaining subpatches until the m-polyhedron is updated by the last m-hemisphere. The intersection, an m-polyhedron, is determined by the vertices on the m-polyhedron.

The procedure of determining the feasible tool-approach directions can be summarized as follows.

1) Input the normal vectors (N_i, $i=1,\ldots,n$) of the subpatches of a surface.
2) Calculate the intersection E_1 of the m-hemispheres of the first three normal vectors.
3) For $i = 2,\ldots,n-2$ do Steps 4-5
    4) Calculate the intersection E_i of the m-polyhedron E_{i-1} and the m-hemisphere of the normal vector N_{i+2}.
    5) If the intersection is null, stop the process. There is no feasible tool-approach direction for the surface.
6) Stop the process. The feasible tool-approach directions for the surface are obtained.

The procedures to calculate the intersection of three m-hemispheres and the intersection of an m-hemisphere and an m-polyhedron are explained below.

3.3 Intersection of three m-hemispheres

When N_1 and N_2 are the unit normal vectors of two m-hemispheres, the vertices V_1 and V_2 on the intersection of the two m-hemispheres (Figure 1) are given by

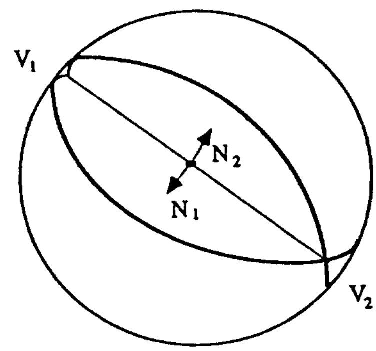

Figure 1. Intersection of two m-hemispheres

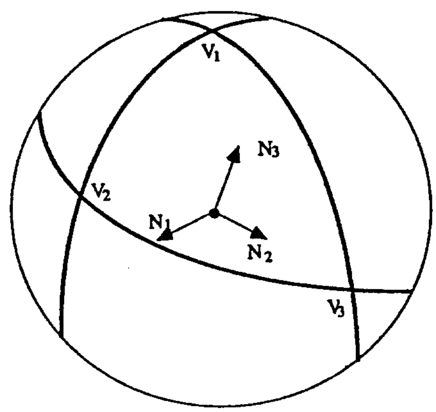

Figure 2. Intersection of three m-hemispheres

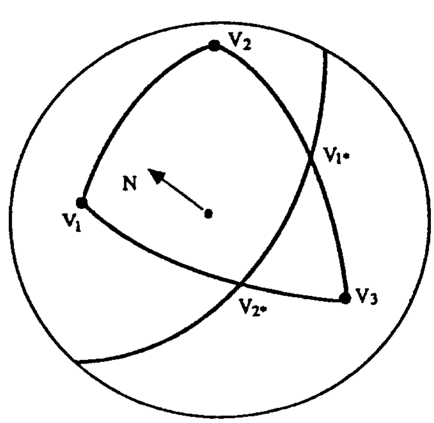

Figure 3. Intersection of an m-hemisphere and an m-polyhedron

$V_1 = P$ and $V_2 = -P$ where $P = \{N_1 \times N_2\} / |N_1 \times N_2|$

When N_i, $i = 1,2,3$ are the unit normal vector of three m-hemispheres, the vertices on the intersection of the three m-hemispheres (Figure 2) are calculated by

$$\text{For } i=1,2,3 \quad V_i = \begin{cases} P_i & \text{if } P_i \cdot N_i \geq 0 \\ -P_i, & \text{otherwise} \end{cases}$$

$$\text{where } P_1 = \frac{N_1 \times N_2}{|N_1 \times N_2|}, \quad P_2 = \frac{N_2 \times N_3}{|N_2 \times N_3|}, \quad P_3 = \frac{N_3 \times N_1}{|N_3 \times N_1|}$$

3.4 Intersection of an m-hemisphere and an m-polyhedron

As shown in Figure 3, the vertices on the intersection of an m-hemisphere and an m-polyhedron is determined by the following three steps:

1) For $i = 1,\ldots,n$ mark the vertices on the m-polyhedron for deletion in the vertex list, if vertex V_i in the vertex list is not on the m-hemisphere.
2) Insert the intersection points of the edges on the m-polyhedron and the b-plane of the m-hemisphere into the vertex list if they intersect.
3) Remove all vertices marked for deletion from the vertex list.

A vertex V_i on an m-polyhedron is on an m-hemisphere if $V_i \cdot N \geq 0$ where N is the unit normal vector of the m-hemisphere.

The intersection V^* of a circular edge between two vertices V_i and V_j on an m-polyhedron and the b-plane of an m-hemisphere is calculated by

$$V^* = \begin{cases} P, & \text{if } [(V_i + V_j) \cdot P] \geq 0 \\ -P, & \text{otherwise} \end{cases}$$

$$\text{where } P = \{(V_i \times V_j) \times N\} / |(V_i \times V_j) \times N|$$

The algorithm for calculating the intersection of an m-polyhedron and an m-hemisphere is summarized as follows.

1) Input the unit normal vector (N) of an m-hemisphere, and the ordered list of vertices (V_i, $i = 1,\ldots,n$) on an m-polyhedron.
2) Set $V_0 = V_n$, $IN_2 = V_0 \cdot N$
3) For $i = 1,\ldots,n$ do Steps 4-8
   4) Set $IN_1 = IN_2$, $IN_2 = V_i \cdot N$
   5) If $IN_1 \geq 0$ AND $IN_2 \geq 0$, continue
   6) If $IN_1 \geq 0$ AND $IN_2 < 0$, mark the vertex V_i for deletion. Then, get the intersection V_1^* of the edge $V_{i-1}V_i$ and the b-plane of the normal vector N, and insert V_1^* before V_i in the linked vertex list.
   7) If $IN_1 < 0$ AND $IN_2 \geq 0$, get the intersection V_2^* of the edge $V_{i-1}V_i$ and the b-plane of the normal vector N, and insert V_2^* before V_i in the list.
   8) If $IN_1 < 0$ AND $IN_2 < 0$, mark the vertex V_i for deletion.
9) Remove all vertices marked for deletion from the list of vertices.

10) Stop the process.

4. Implementation

The machinability checking algorithm is implemented in C on a SUN 4 workstation as a module in the integrated sculptured surface design and manufacturing (ISSDM) system (Kim and Choi, 1992). The input data is a B-spline surface description, and the output is a set of feasible tool approach directions for the surface. The output data consists of the vertices on the m-polyhedron.

Figure 4 shows a screen dump from the system. The upper left window shows a sculptured surface to be checked. The lower left window shows the surface subdivided by an adaptive subdivision method. The lower right window shows the m-polyhedron defining the set of the tool-approach directions for the surface. The upper right window shows the vertices on the m-polyhedron.

Examples for finding feasible tool-approach directions are shown in Figures 4-6. Figures 4-5 show the sets of feasible tool-approach directions for a sculptured surface when the specified subdivision tolerances are 10 and 3 unit, respectively. The set of feasible tool-approach directions for the subdivided surface with tolerance = 10 unit is slightly larger than the set of feasible tool-approach directions for the subdivided surface with tolerance = 3 unit. However, the difference in the set of feasible tool-approach directions is negligible as the subdivision tolerance is getting smaller. In Figure 6, no feasible tool-approach direction exists for the given surface.

If a set of feasible tool-approach directions exists for a sculptured surface, the surface is machinable on a three-axis milling machine. When a surface is machinable on a three-axis milling machine, the NC tool paths and G-codes for the surface can be generated automatically by the NC tool path generation algorithm (Kim and Ko, 1991), as shown in Figure 7.

5. Conclusion

Determining tool-approach directions is an important issue when an effort is made to transfer CAD data into manufacturing automatically. An algorithm is developed to determine whether a given part can be machined on a three-axis milling machine. In the algorithm, a sculptured surface is first approximated with smaller subpatches by an adaptive subdivision method. Then, the set of feasible tool-approach directions for the subpatches is calculated

If a set of feasible tool-approach directions exists for a sculptured surface, the NC tool paths and G-codes for machining the surface on a three-axis milling machine can be generated automatically by an NC tool path generation algorithm.

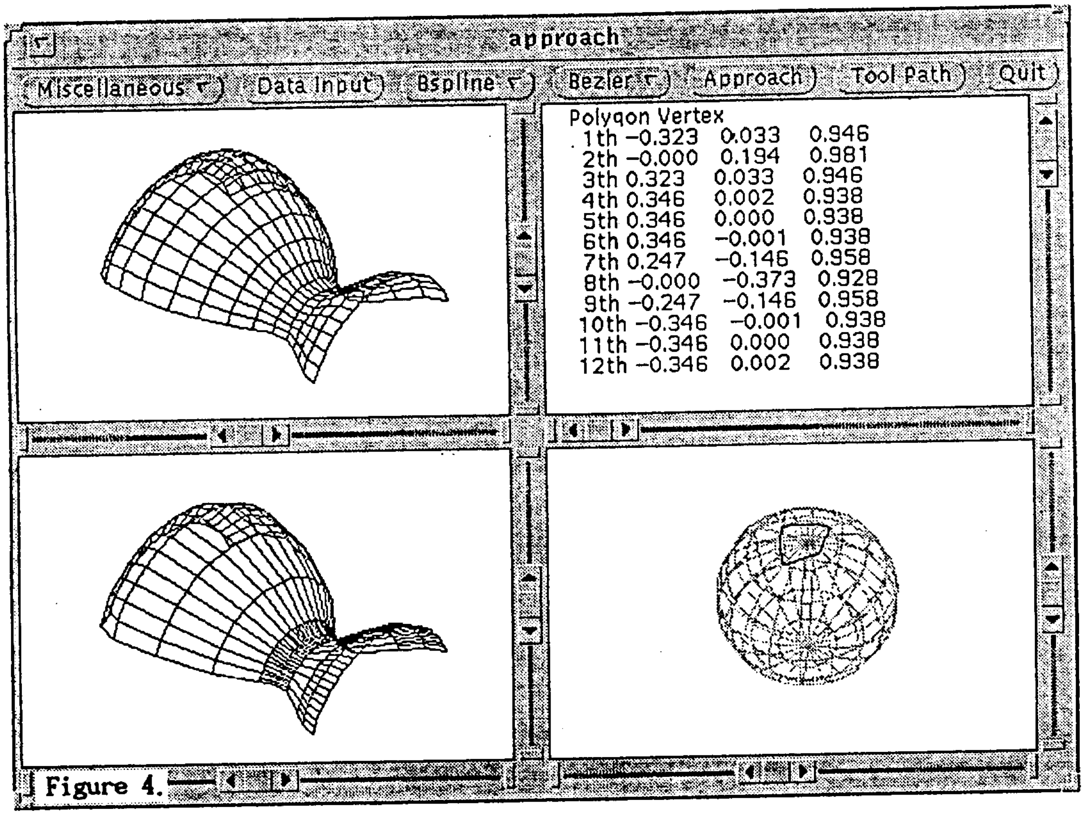

Figure 4.

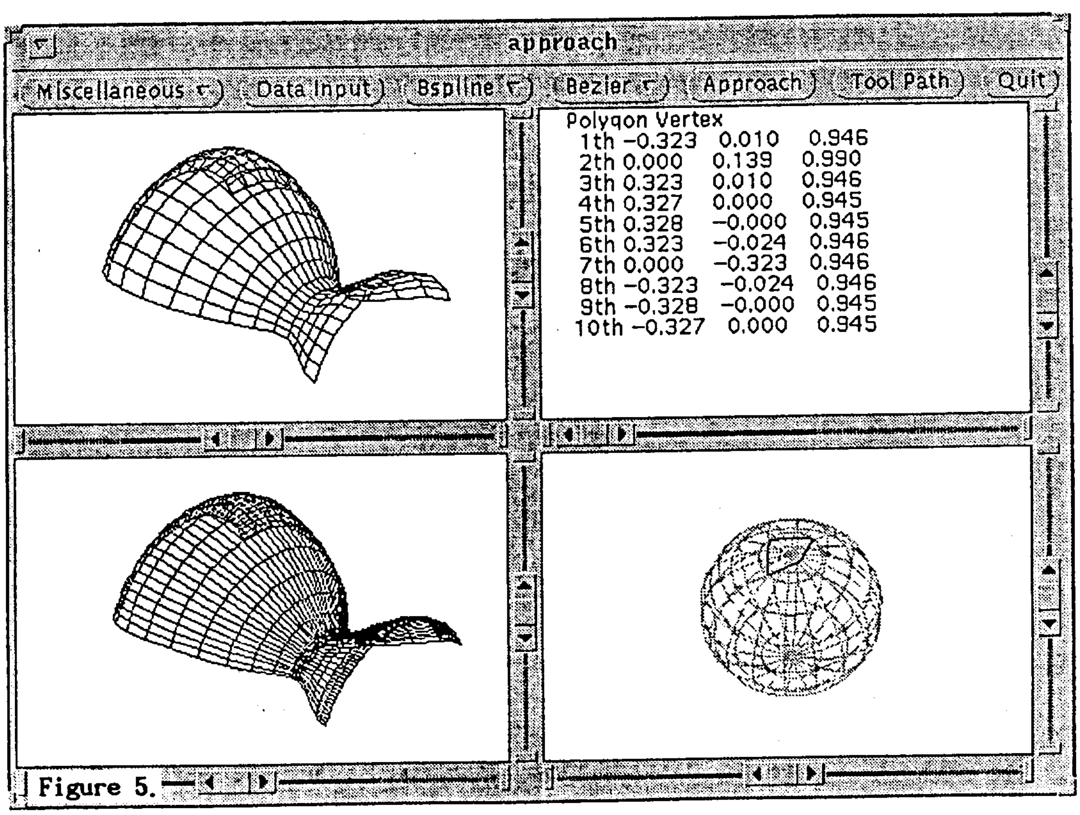

Figure 5.

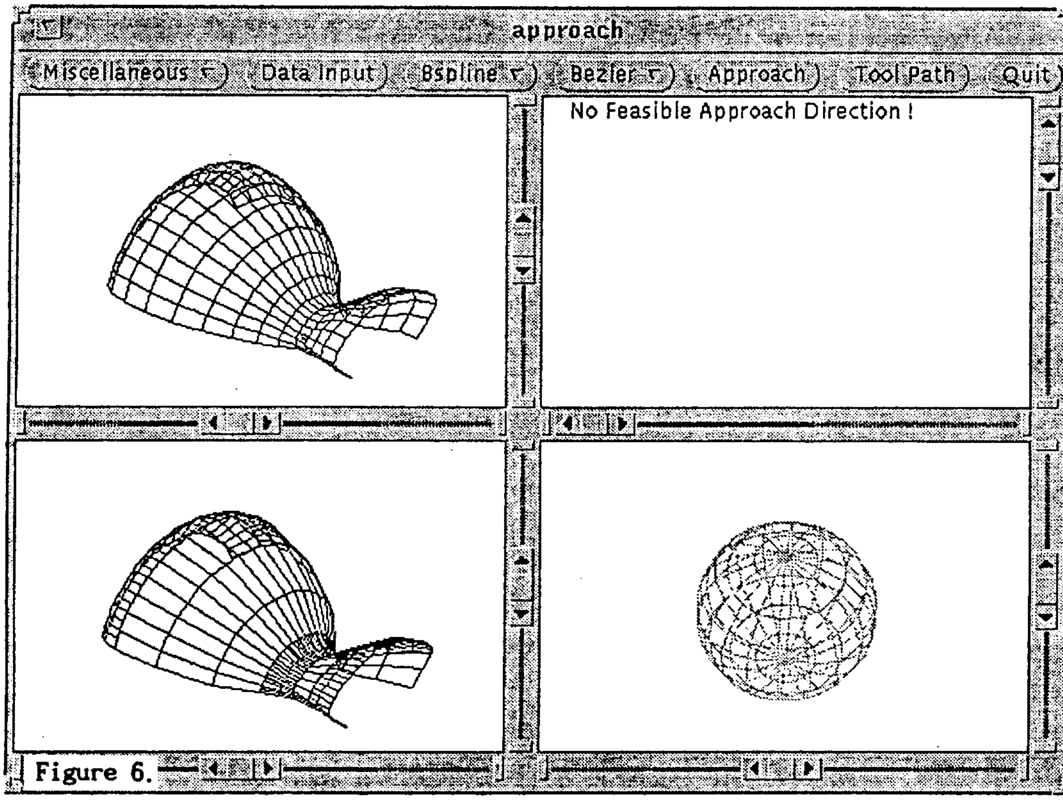

Figure 6.

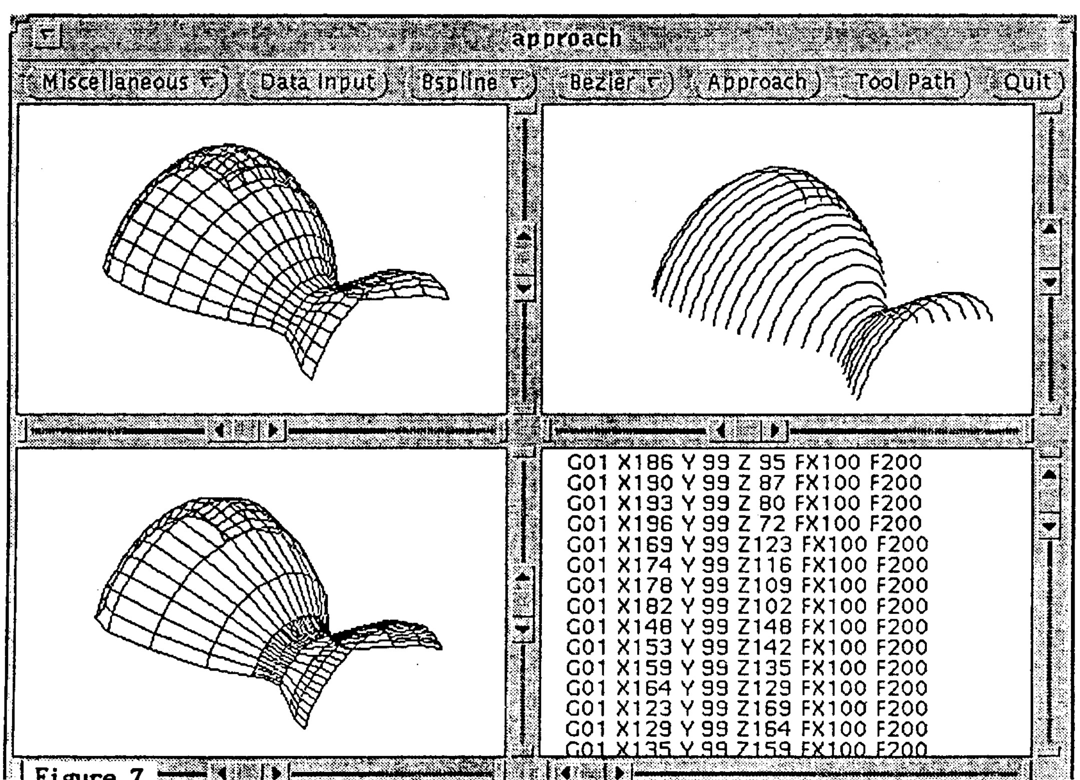

Figure 7.

The machinability checking algorithm is implemented in C running on a SUN engineering workstation as a module in an integrated sculptured surface design and manufacturing (ISSDM) system. The algorithm can be used for orientation and fixturing of the workpiece for interference free machining, and act as a filter that rejects improper part design for three-axis milling before the NC tool path generation module is activated. The algorithm can also be applied to checking the translational separability of polyhedral parts in automatic assembly.

References

1. Boehm, W. (1980) Inserting new knots into B-spline curves. *Computer Aided Design*, 12(4), 199-201.

2. Boehm, W. (1981) Generating the Bezier points of B-spline curves and surfaces. *Computer Aided Design*, 13(6), 365-366.

3. Choi, B. (1991) *Surface Modeling for CAD/CAM*, Elsevier.

4. Farin, G. (1990) *Curves and Surfaces for Computer Aided Geometric Design*, Academic Press.

5. Joshi, S. and Chang, T. C. (1986) Feasible tool approach directions for machining holes in automated process planning systems. *Proc. ASME Symposium on Integrated and Intelligent Manufacturing*, Anaheim, CA, USA.

6. Kim, D. (1990) Cones on Bezier curves and surfaces. Ph.D. Dissertation, Dept of Industrial and Operations Engineering, University of Michigan, USA.

7. Kim, K. and Ko, B. (1991) An NC tool path generation algorithm for sculptured surface manufacture. Technical Report IE-TR-92-5, Department of Industrial Engineering, Pohang Institute of Science and Technology, Korea.

8. Kim, K. and Choi, Y (1992) An integrated sculptured surface design and manufacturing system. Technical Report IE-TR-92-6, Department of Industrial Engineering, Pohang Institute of Science and Technology, Korea.

9. Peng, Q. S. (1984) An algorithm for finding the intersection lines between two B-spline surfaces. *Computer Aided Design*, 16(4), 191-196.

10. Schmitt, F. J. M., Barsky, B. A. and Du, W. (1986) An adaptive subdivision method for surface fitting from sampled data. *Computer Graphics*, 20(4), 179-188

11. Su, C. and Mukerjee, A. (1991) Automated machinability checking for CAD/CAM. *IEEE Trans. on Robotics and Automation*, 7(5), 691-699.

12. Tseng, Y. and Joshi, S. (1991) Determining feasible tool approach directions for machining Bezier curves and surfaces. *Computer Aided Design*, 23(5), 367-379.