Int J Adv Manuf Technol (2000) 16:100–106

The International Journal of
Advanced Manufacturing Technology

Multipatched B-Spline Surfaces and Automatic Rough Cut Path Generation

S. H. F. Chuang and I. Z. Wang

Department of Mechanical Engineering, National Chung-Hsing University, Taichung, Taiwan

A method is proposed to generate rough cut machining tool paths for multipatched B-spline surfaces. For a group of B-spline surfaces, smoothly connected, separated or intersecting one another, this method can guarantee generating paths without overcutting any of the surfaces under consideration.

The use of multipatched surfaces to construct complex surfaces can greatly increase the controllability and friendliness in the design of surfaces. A method for rough cut planning by convex hull boxing is proposed to generate paths guaranteeing no over-cut. Since no computation for solving nonlinear equation is involved, the rough cut plan is robust and efficient.

Keywords: B-spline surface; Convex hull; Multipatched surface; Path generation; Rough cut

1. Introduction

In today's industry, products change rapidly. The shapes of industrial products become complicated and the degrees of variation increase. A single freeform surface, such as the most widely used surface type, a B-spline surface [1,2], no longer satisfies all the needs for industrial shape design. Therefore, multipatched B-spline surfaces are introduced for design conditions such as section shapes changing abruptly or punched holes existing in a surface. The use of multipatched surfaces will also increase controlling capability and friendliness in constructing complicated freeform shapes. Therefore, there is a need to develop an efficient method to design and manufacture multipatched surfaces.

The object of this research is to develop a method which can efficiently generate rough cut machining tool paths for multipatched B-spline surfaces without overcutting. A multipatched B-spline surface includes several patches of surfaces existing in the design database. They can be continuously patched together, distantly separated or even intersect one another. The method can guarantee the generation of rough cut tool paths without over-cutting any of the included B-spline surfaces.

Generally, there are two methods to generate rough cut tool paths. One method is to offset the shape contour, as shown as in Fig. 1(*a*). This method is appropriate for stock material which is a drop-forged part or a casting whose profile is an enlarged form of, or is similar to, the finished part [3]. The second method is to slice the stock material into layers and to remove unwanted material layer by layer, as shown in Fig. 1(*b*). It is suitable for the stock which has primitive simple shapes, such as a block or a cylinder [4,5].

Conventionally, slicing planes are evenly spaced for constructing machining layers for automatic rough cut path generation of B-spline surfaces [4]. A surface is directly sliced into the same depth layers and each layer is considered as a pocket for machining. Because the intersecting contour of a B-spline surface and a plane may involve high degree curves with singularities, the computation is complex and prone to numerical errors. A practical solution is to construct an approximate model to replace the original surface, so that the computation complexity is reduced based on the simpler model for rough cut path planning.

A method, called the highest z-valued method or z-mapped method is used to define tool paths in the x, y-domain of the design surface, then, the highest z-value for each x, y-point is computed [6–8]. You and Chu used a matrix of grid heights to approximate the surface embedded in a solid model [9]. A rigid net of rectangles was planned on the X, Y-plane and the corresponding z-heights of vertices of rectangles are found.

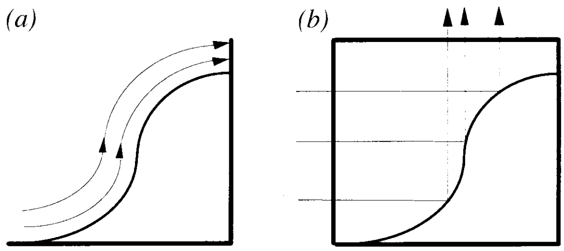

Fig. 1. Two methods for generating rough cut paths. (*a*) by contouring, (*b*) layer by layer.

Correspondence and offprint requests to: Dr. S. H. Frank Chuang, Department of Mechanical Engineering, National Chung-Hsing University, 250 Kuo-Kuang Road, Taichung, Taiwan. E-mail: shchuang@nchu.edu.tw

According to the maximum height of the four corners of each grid rectangle, a hexahedral block is built for each grid rectangle. The approximate model for the surface is a combination of all of the blocks for the grid net. Lee et al. proposed an octree method to construct an approximate model [5]. The occupying space of a solid is sliced into eight block components, and each block is further sliced until the octree approximating model satisfying the specified tolerance. The method, to directly slice a solid with surfaces into layers, has long been considered to be a difficult numerical problem of surface/surface intersection [10,11]. The z-mapped grid height method and the octree method still cannot avoid large amount of numerical computation involving line to surface intersections.

Chuang and Pan proposed a boxing method to approximate a single B-spline surface by wrapping convex hulls of Bezier surfaces to obtain boxes for approximation [12]. Based on that method, we have further extended the method for rough cut path generation of multipatched B-spline surfaces.

For a multipatched B-spline surface, in which each B-spline surface can be transformed into a group of Bezier surfaces [13]. Each Bezier surface can be protected by the convex hull of its control points, subsequently, the protecting wrapping box of the convex hull is used because of ease of computation. By uniting all of the wrapping boxes of the Bezier surfaces, an approximate model is constructed. If the tool paths do not interfere with the constructed approximate model, then they will not interfere with the B-spline surfaces belonging to the multipatched surface. The approximate model is then sectioned into non-uniform layers. Each layer is constructed as a $2\frac{1}{2}$-D cavity, which is called a pocket for machining. Any of the available pocketing path generation methods can be used to generate non-overcutting paths for each of layers. The paths assembled from the top, down to the bottom, are the rough cut paths for the multipatched B-spline surface.

The method is implemented to generate NC code. Real objects can be produced efficiently with rough cut machining.

2. Representation of Multipatched Surfaces

The data structure and representation of multipatched surfaces adopted by this paper is explained as follows. A multipatched surface is stored in a doubly linked list for a collection of B-spline surfaces, where each node has a stored B-spline surface, as shown in Fig. 2. The nodes in a doubly linked list do not specify any geometric relations between those corresponding surfaces. The reason for using this simple data structure, is that the relative positions and relationships between surfaces do not influence the processing by the proposed method.

The node of a surface includes a pointer which points to the head of a four-directional linked list representing a net of vertices (see Fig. 3). The net of vertices is the set of control points of a B-spline surface. Each node in the four directional linked list stored with the geometric and topological data represents a control point of the B-spline surface. The geometric data include position coordinates. The topological data contain the four directional links which point to, respectively, the left, right, forward and backward nodes of the vertex.

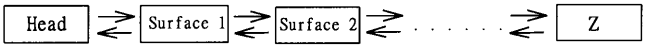

Fig. 2. Data structure of surfaces in a multipatched surface.

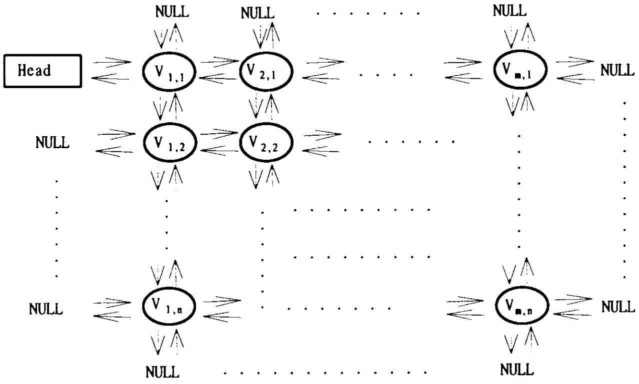

Fig. 3. Data structure of the net of vertices of a B-spline surface.

A multipatched surface is a collection of B-spline surfaces. In order to use the convex hull property of Bezier surfaces to build the approximation model for the collection of B-spline surfaces, all B-spline surfaces are transformed into piecewise Bezier surfaces. The knot insertion method is applied to repeat the knots of a B-spline surface and the corresponding piecewise Bezier control points will be generated without changing the shape of the original surface [13–15]. If $S(u, v)$ represents the position vector of an arbitrarily selected point on a surface, for a bicubic example with degree 3×3, a B-spline surface is defined as follows [2,16].

$$(u, v) = \sum_{i=0}^{m} \sum_{j=0}^{n} N_{i,3}(u) N_{j,3}(v) P_{i,j}$$

where, $P_{i,j}$ is a control point on the B-spline control net; the number of control points is $(m + 1) * (n + 1)$; $N_{i,3}(u)$ and $N_{j,3}(v)$ are, respectively, the B-spline basis functions defined in the U and V knot vector directions. The knot vectors may be open, uniform and non-periodic which are frequently used in free-form surface design. Such knot vectors are defined as follows [16]:

$$U = \{0, 0, 0, 0, u_4, \dots, u_m, 1, 1, 1, 1\},$$
$$V = \{0, 0, 0, 0, v_4, \dots, v_n, 1, 1, 1, 1\}$$

If there are no internal knots, B-spline basis functions are simplified to Bernstein functions and a B-spline surface is equivalent to a Bezier surface with a given set of parameters (e.g. degree of 3). An arbitrary point on the Bezier surface, $Q(u, v)$, is defined as follows [2,7]:

$$(u, v) = \sum_{i=0}^{3} \sum_{j=0}^{3} B_{i,3}(u) B_{j,3}(v) P_{i,j}$$

where, $P_{i,j}$ is a control point on the Bezier control net; the number of control points is 4×4. $B_{i,3}(u)$ and $B_{j,3}(v)$ are, respectively, Bernstein functions defined in the U and V knot vector directions.

For a bicubic B-spline surface, executing knot insertion three times for each internal knot of knot vector *U*, then, three times for each internal knot of knot vector *V*, the control points for the B-spline surface become control points for piecewise Bezier surfaces [13,17]. After the transformation through the knot insertion procedures, a bicubic B-spline surface, as in Fig. 4, is transformed into four piecewise bicubic Bezier surfaces.

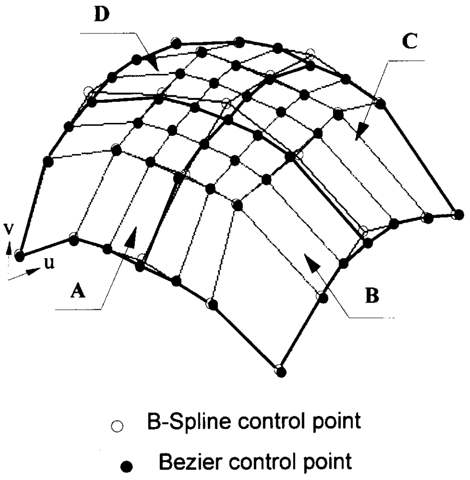

Fig. 4. Transforming a B-spline surface to Bezier surfaces.

3. Design with a Multipatched Surface

For complex shape variation of a local area within a surface, it is required to increase the number of control points such that the increased shaping controllability can help to achieve the desired shape variation. The set of control points of a B-spline surface must be arranged in the form of an $m \times n$ array. Such an array of control points cannot be increased by simply inserting control points in a single column or a single row. Control points must be increased by inserting an entire column or an entire row of control points. This is not convenient for situations where a local shape with a great deal of variation must be constructed or modified.

If a single B-spline surface is used to construct a surface, in order to adjust the shape of a complex partial area to the satisfaction of a designer, the density of control points for that area must be higher than for the others. Since the initial specified control points at the beginning of design are evenly distributed, a large number of control points must be adjusted, as shown in Fig. 5. The movement of a large number of control points increases the design complexity and the distribution of control points is distorted unevenly.

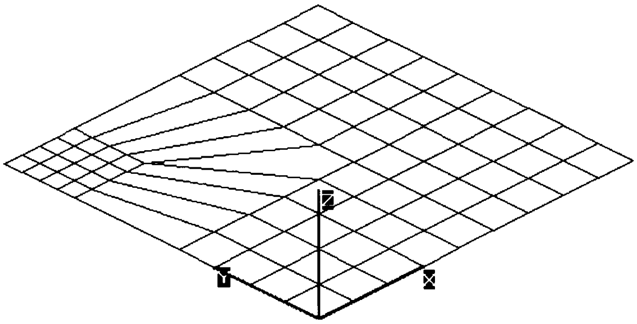

Fig. 5. Model with non-uniform distribution of control points.

Instead of using a single patch for surface construction, a multipatched surface is a good alternative by having a different density for each patch according to the degrees of complexity of shape areas (see Fig. 6). Fewer control points require adjustment and with smaller movements compared to a single patch surface construction.

The model in Fig. 5 is redesigned as a multipatched model, as shown in Fig. 6. Four patches are used to improve the distorted distribution of control points between areas with different density requirements. The designer will not be asked to move a large number of control points as in the design with one single patch, in which a uniform density is specified over the entire surface at the initial design. At the beginning of the design, the designers can specify different densities of surface patches to their satisfaction.

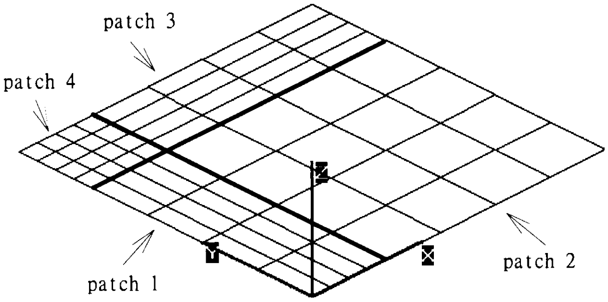

Fig. 6. Model designed with four patches.

The multipatched surface model also has advantages for the situation when the surface sections vary greatly. In a single patch model, each row has the same number of control points for each surface section, no matter how the section size changes. This tends to make the control points too dense in a small area and too sparse in a large area. An example of such a condition is shown in Fig. 7. The surface is redesigned with a multipatched model. Adapting to section changes, a model designer can determine the number of patches, the types of joining and the numbers of control points for each row and each column, see Fig. 8. A uniform distribution of control points can be obtained easily even though sections change greatly.

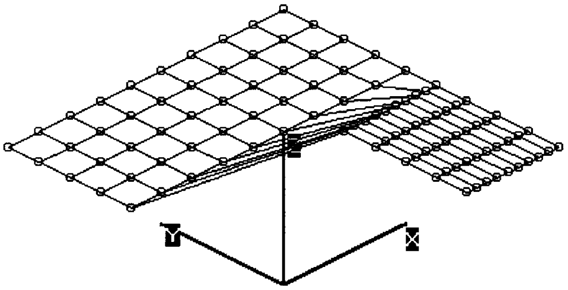

Fig. 7. Model with sections varying greatly.

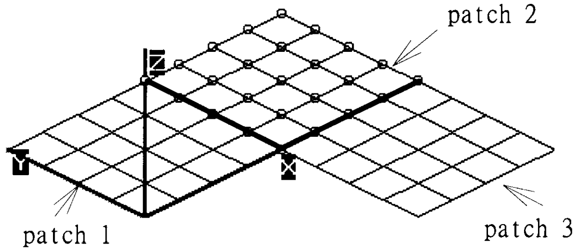

Fig. 8. Model design adapted to section changes.

In order to join two surfaces along a common boundary, the boundary curves on the two surfaces must have the same geometry so that the two surfaces can glue together with at least G0 continuity. A simple method is to give the two surfaces the same set of control points along the common boundary. Thus, there will be no gaps between the two surfaces at the joining boundary [1,7,18].

4. Rough Cut Path Generation

Based on constructing the model described in the previous sections, a method for rough cut path generation of multipatched surface models is presented in this section. The B-spline surface patches, stored in a doubly linked list, are extracted sequentially. Each B-spline surface is transformed into piecewise Bezier patches. Using the convex hull property of Bezier surfaces, an approximate model can be constructed by composing convex hull boxes which are constructed by wrapping up their Bezier control points [12]. The approximate model leads to non-uniform sectioned machining layers with different depths, where each machining layer is treated as a machining pocket.

4.1 Approximating Model for Rough Cut

Although B-spline surfaces themselves have the local convex hull property, local convex hulls overlap in large areas, and the approximation model constructed according to this characteristic cannot approximate a B-spline surface well. Using the convex hull boxes of Bezier surfaces transformed from a B-spline surface, the overlapping area is comparatively smaller, and the corresponding approximation model is much closer to the B-spline surface than the model directly constructed from local convex hulls. The stock volume will be smaller and the removed volume larger, which means that a better cutting plan can be found.

The approximate model of a multipatched surface is composed of united rectangular boxes which are constructed from Bezier surfaces decomposed from B-spline patches. With such an approximate model built of rectangular boxes, the Boolean operations for obtaining contours of pocketing layers can be greatly simplified and over-cutting can easily be avoided completely. The construction of each rectangular box uses the convex hull property of Bezier surfaces, which states that a Bezier surface is contained in the convex polyhedron of control points of the Bezier surface [1,7]. If a machining process does not cut inside the convex hull of a Bezier surface, the machining will not overcut the corresponding surface. The Boolean operations for general polyhedra of convex hulls still require complex modelling procedures and computations. A rectangular box is used to wrap up a convex hull, see Fig. 9, so that the modelling and computations can be greatly simplified. A side view of an approximation model for two patches of B-spline surfaces are shown in Fig. 10.

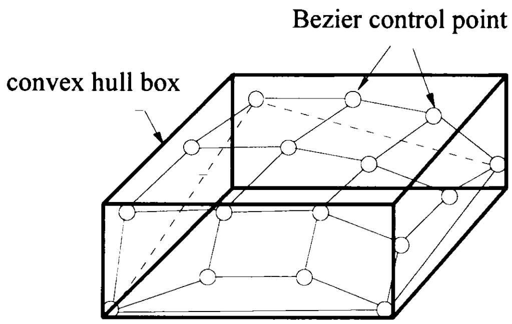

Fig. 9. Convex hull box.

After the approximate model of a multipatched B-spline surface is established, the top faces of rectangular boxes are used as slicing planes to divide the stock material into machining layers as shown in Fig. 10. Each layer is considered as a machining pocket using the intersection contour of the slicing plane and the approximate model as the pocket boundary. The material between the approximate model and stock boundary is thus the pocketing cavity for removal.

After the above slicing process, the obtained depths of layers are different from one another. Such resulting depths of cut are required to be further examined by considering the factors of tool load capacity and cutting time efficiency. If the depth of a layer exceeds the maximum depth permitted cut, the tool life will be greatly reduced. If a cutting layer is too thin and below the minimum depth of cut, the cutting time is too long to obtain an efficient cut. Therefore, the layers which are too thick must be subdivided and the layers which are too thin must be merged.

The given machining conditions for the rough cut will influence the volume and the rate of material removal. The larger the tool size selected, the fewer tool paths are generated and the faster the rough cut can be completed. Increasing the maximum and minimum machining depth will reduce the number of layers, and the total path distance will be shortened, thus the time required for the rough cut will decrease. However,

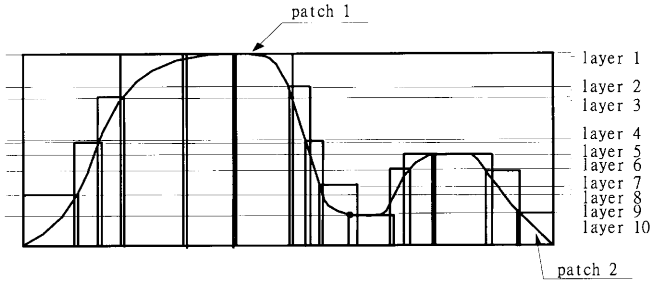

Fig. 10. Stock sliced into non-uniform layers.

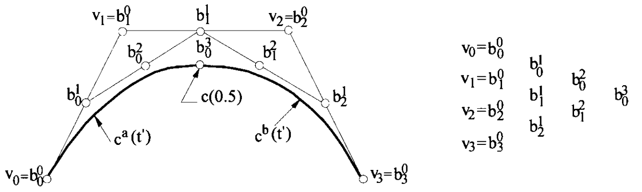

Fig. 11. Subdivision of a Bezier curve.

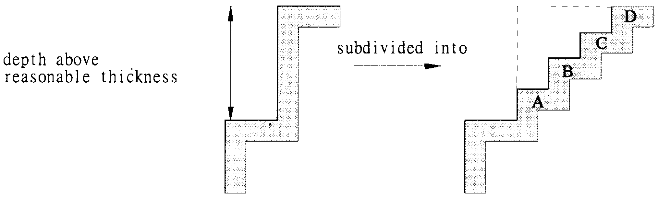

Fig. 12. A layer subdivided into four layers.

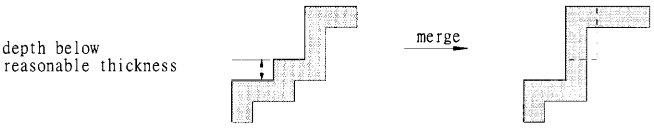

Fig. 13. Two layers merged into one layer.

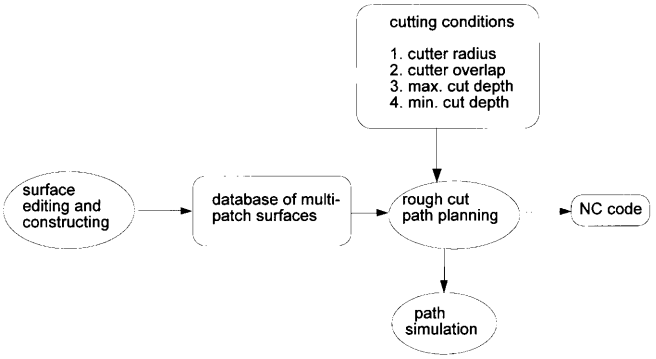

Fig. 14. Infrastructure of rough-cut system.

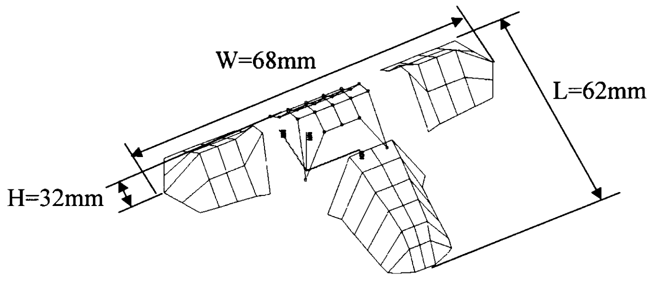

Fig. 15. Water tap model.

selecting a larger tool size or raising the machining depth will leave more volume for finish machining.

For a layer with a depth exceeding the maximum permitted depth of cut, the Bezier surface in the layer is subdivided into smaller Bezier surfaces by applying the de Casteljau algorithm [1,7]. As shown in Fig. 11, the Bezier curve constructed by control points, b_0^0, b_1^0, b_2^0 and b_3^0, and is subdivided into curves controlled by b_0^0, b_0^1, b_0^2 and b_0^3, and b_0^3, b_1^2, b_2^1 and b_3^0. Applying subdivisions in both directions, *u* and *v*, a Bezier surface can be subdivided into four surfaces.

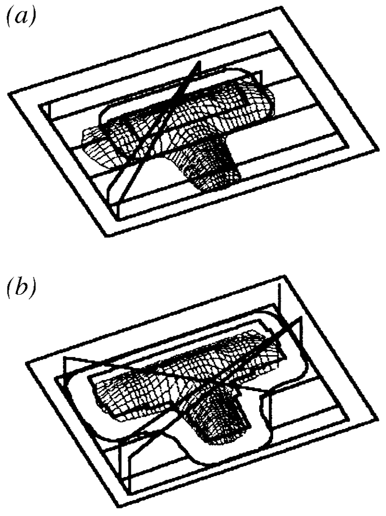

Fig. 16. Cutting simulation. (*a*) The 2nd layer, (*b*) the 9th layer.

As shown in Fig. 12, the Bezier surface in a layer is subdivided into four Bezier surfaces. The layer is sectioned at each top face of convex hull boxes built of subdivided Bezier surfaces, the original layer is thus sliced into four layers.

For any layer with a depth below the permitted minimum depth of cut, the top face of this layer will be elevated to the height of its next top layer, as shown in Fig. 13. If the merged layer is still below the minimum depth of cut, the resulting layer is further merged to the next top layer. Otherwise, the merging process is terminated.

After the positions and depths of layers have been determined, all of the convex hull boxes intersecting each layer are united to obtain the pocketing boundary for that layer. Each layer is then formed with a standard machining pocket boundary for path generation. From the top layer down to the bottom, the generated paths are assembled as the rough cut paths for the designed multipatched B-spline surfaces.

5. Implementation

Based on a single-patch system [12], the proposed method for rough cutting of multipatched B-spline surfaces is implemented in Visual C++ on Windows of PC. The system infrastructure is shown in Fig. 14, where the surface editing and rough cut path planning are two major modules.

There are 3 major functions in the surface editing and modification module for multipatched surfaces:

1. Read in data of control nets and construct multipatched B-spline surfaces.
2. Transform B-spline surfaces into piecewise Bezier surfaces and display the surfaces on a computer screen.

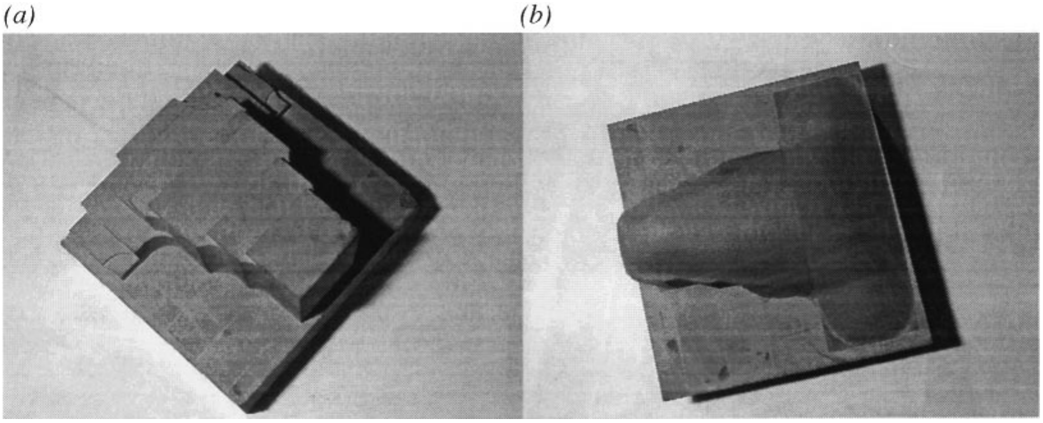

Fig. 17. Cutting results.

3. Move control points and modify each surface shape until every surface shape is satisfactory.

In this editing module, plane models for control nets are used to help select control points in order that the points on the screen can be clearly distinguished and the topological relationships between control points are explicitly displayed [18,19].

The rough cut planning module has two major functions:

1. Construct the surface approximation model according to those piecewise Bezier surfaces transformed from B-spline surfaces.
2. According to specified cutting conditions, the tool paths are generated through layer slicing and structuring of pocketing boundaries.

The tool paths are further simulated, for interference checking on a computer. The generated CL data can be transformed into NC code, which can be transferred to a CNC machine for the actual cutting process.

5.1 Example of a water tap model

The face sections of a water tap model change rapidly. The sections are established by multipatched B-spline surfaces. According to the shape characteristics and variations, the number of patches and the number of control points for each surface is determined in a way that allows the surface construction to be done conveniently. Four surface patches are used to construct the tap model as shown in Fig. 15.

In this example, the machining conditions are set as: tool radius = 6 mm, overlapping distance = 1 mm, maximum depth of cut = 6 mm, and minimum depth of cut = 2 mm. The total number of layers generated is 10, with an NC code of 1118 lines. Simulation of the tool paths in two of the layers is shown in Fig. 16. By applying the above path generation algorithm, the rough cut of the tap model is shown in Fig. 17(*a*). Its corresponding finish model is shown in Fig. 17(*b*).

6. Conclusions

Multipatched B-spline surfaces have advantages over a single B-spline surface in design. In this paper, a tool path planning method for rough cutting is proposed. For multipatched B-spline surfaces existing in the database, no matter how those surfaces are formed, e.g. smoothly connected together, intersected or widely separated, the proposed method can guarantee generating tool paths with no over-cutting on any part of the designed surfaces.

For tool path generation with flat-ended mills, almost all of the existing methods in commercial CAM systems solve the nonlinear equations for very high degree intersection curves between a slicing plane and a B-spline surface. Since the intersection curves are approximated and solved linearly in this paper, the proposed method is much more efficient and numerically stable than the methods used in commercial systems.

The next step in our work will be to take the resulting stock from rough cutting and plan the tool paths for the finish cut. In implementation, the finish cut will use the same design database as the rough cut, and paths generated for the resulting surface cut can satisfy the specified tolerance.

References

1. G. Farin, Curves and Surfaces for Computer Aided Geometric Design, Academic Press, 1990.
2. D. F. Rogers and J. A. Adams, Mathematical Elements for Computer Graphics, McGraw-Hill, 1990.
3. C. H. Chang and M. Melkanoff, NC Machine Programming and Software Design, Prentice-Hall, 1989.
4. W. Chungwatana, A. C. Line and W. F. Liu, "Automated process planning and numerical control code generation for mechanical parts with sculptured surfaces", The Second International Conference on Automation Technology, vol. 2, pp. 171–178, July 1992.
5. K. Lee, T. J. Kim and S. E. Hong, "Generation of tool path with selection of proper tools for rough cutting process", Computer-Aided Design, 26(11), pp. 822–831, November 1994.
6. B. K. Choi, C. S. Lee, J. S. Huang and C. S. Jun, "Compound surface modeling and machining", Computer-Aided Design, 20(3), pp. 127–136, 1988.
7. B. K. Choi, Surface Modeling for CAD/CAM, Elsevier, 1991.
8. R. B. Jerard and R. L. Drysdale, "Geometric simulation of numerical control machining", Proceedings ASME International Computers in Engineering Conference, ASME, pp. 129–136, 1988.
9. C. F. You and C. H. Chu, "NC rough cut machining for solid models", The Second International Conference on Automation Technology, vol. 2, pp. 75–82, July 1992.

10. D. Lasser, "Intersection of parametric surfaces in the Bernstein–Bezier representation", Computer-Aided Design, 18(4), pp. 186–192, 1986.
11. R. B. Lee and D. A. Fredricks, "Intersection of parametric surfaces and a plane", IEEE Computer Graphics and Applications, pp. 48–51, August 1984.
12. S. H. Chuang and C. C. Pan, "Rough cut tool path planning for B-spline surfaces using convex hull boxes", The International Journal of Advanced Manufacturing Technology, 14, pp. 85–92, 1998.
13. W. Boehm, "Generating the Bezier points of B-spline curves and surfaces", Computer-Aided Design, 13(6), pp. 365–366, November 1981.
14. W. Boehm, "Inserting new knots into B-spline curves", Computer-Aided Design, 12(4), pp. 199–201, July 1980.
15. W. Boehm and H. Prautzsch, "The insertion algorithm", Computer-Aided Design, 17(2), pp. 58–59, March 1985.
16. L. Piegl and W. Tiller, "Curve and surface construction using rational B-spline", Computer Aided Design, 19(9), pp. 485–498, November 1987.
17. N. Anantakrishnan and L. A. Piegl, "Integer subdivision algorithm for rendering NURBS curves", The Visual Computer, 8(3), pp. 149–161, 1992.
18. S. H. Chuang and Z. I. Lin, "Deformation control of freeeform surfaces using plane models", Journal of Engineering, National Chung-Hsing University, 9(1), pp. 17–25, 1998.
19. C. U. Lin, "Computer aided surface design using a plane model", Master's thesis, National Chung-Hsing University, Taichung, Taiwan, 1994.