Algorithms for Real-Time Tool Path Generation

Gyula Hermann
John von Neumann Faculty of Information Technology, Budapest Polytechnic
H-1034 Nagyszombat utca 19 Budapest Hungary, hermgyviif.hu

Abstract:The conventional input of a CNC, a sequence of data points, describing the tool motion necessary for machining of sculptured surfaces, represent a huge amount of information. As free-form surfaces are usually described as a set of bicubic pathes, it is a strightforward solution to extend the capabilities of the Numerical Control unit with bicubic interpolation, offset generation and collision checking. The tool path is calculated during the cutting process.

Keywords:CNC, patch programming, motion planning, tool path calculation, sculptured surfaces, 2½D and 3D machining

1 Introduction

In Computer Aide Geometric Design we wish to create, in the computer, a model which will adequately describe the particular class or classes of three dimensional shapes which are to be designed. The creation of the model is a mean to produce the necessary data to manufacture the physical object.

Unfortunatly numerical control was established as a discipline before CAGD and whereas it is possible to generate numerical control data directly from stored geometric information, conversation either manually, or by means of a computer takes place. The surface curves describing the tool path are in the majority of control units are interpolated by a sequence of short line segments. In the subsequent paragraphs it becomes apperent, that remarkable improvement in performance and memory utilisation can be gained, when the tasks are transferred in the vinicity of the manufturing process by increasing the CNC's capabilities.

This problem is especially important when we are machining free-form surfaces. Here the geometric model contains the information in condenced form, whereas the NC information is at least two orders of magnitude greater. The algorithms described in this paper enable the tool path generation for sculptured surfaces in the CNC unit directly form the surface model.

2 The Place of Motion Planning in the Design Process

The procedure for determining how a part is to be manufactured called process planning. In order to determine the process not only the geometry of the workpiece and the blank are needed, but material properties, form and dimensional tolerances and surface finish has to be taken into account.

The process planning can be broken down into three levels: process sequence, operation planning and motion planning level. This last step became in the last years more and more the integral part of CNC units. One can recognize the analogy with a conventional workshop, where the machining of small batches relies to a great extent on the skill of the worker.

In off-line computer aided motion planning two different approaches can be distinguished: the so-called object surface driven approach, where the contact point between the tool and the final workpiece is determined first, together with the surface normal vector on appropriate points. After that the offsetting of the cutter location is calculated. In this case modification of the tool correction can be easily determined. The second approach is the so-called offset surface driven approach, where the offset surface is generated first and than the reference point of the tool is calculated. The two approaches are compared qualitatively in Table 1.

Point of view	**Object surface driven**	**Offset surface driven**
Deviation control	Direct	Indirect
Collision testing	Difficult	Simple
Maintaining constant cutting speed	Simple	Simple
Modification of tool correction	Simple	Complex
Exception handling	Simple	Simple

Table 1. Comparison of object and offset surface driven approaches

Careful evaluation showed that the following steps can be effectively realized in a numerical control unit:

In **turning**

- Determination and following the path of the reference point both in roughing and finishing
- Modification of feeds and speeds if necessary

In **milling**

- Determination of the path traced by the point of contact between the tool and the final surface, and the unit surface normal vector
- Computation of the reference point's path according to the given tolerances
- Adjustment of feeds and speeds

3 Real-Time Cutter Path Generation in Turning

The algorithms first step is the determination of a close approximation of the offset profile[5]. As the original curve is either a B-spline or a rational B-spline the following simple procedure can be used:

> Each leg of the control polygon is offsetted by the tool radius. The intersection of the offset polygon leg's taken in a pairwise manner form the set of contro points of the approximate offset profile. These points define a B-spline or a rational B-spline [4]. The accuracy of the approximation is checked at the curve segment's midpoint. If it is greater then the permitteddeviation the original profile is subdivided and process is repeated again.

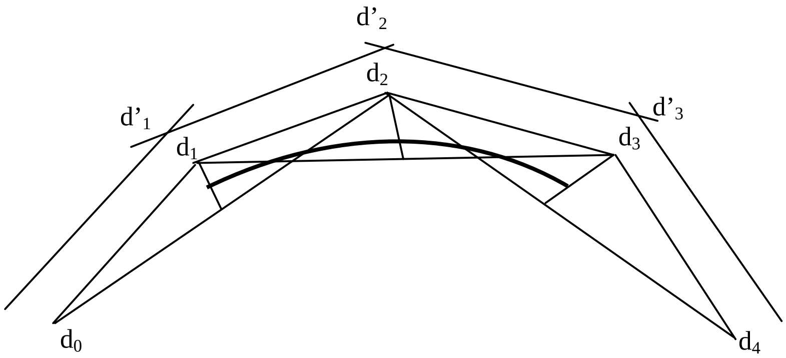

Fig. 1 The construction of the offset curve

One can easily realize that in the limit these steps lead to the offset curve. The procedure is general and has several advantages. Two of them are worth to be mentioned here: line and circles are offsetted precisely with one iteration, the parametrization of the offset corresponds to that of the original curve; $\mathbf{O}(s_0)$ is very close to $\mathbf{C}(s_0)+r.\mathbf{N}(s_0)$ for every s_0. The procedure is given in Fig. 1.

The offset curve is generated for each span separetely during the machining process.

This curve is approximated in turn by short line segments according to the given tolerances. The parameter step is estimated using the error formula for the piecewise Lagrange interpolation:

$$\Delta s = \frac{\sqrt{2}\ tol}{\sqrt{\|\mathbf{C}^2{}_s\ (s)\|}} \qquad (1)$$

where $\|\mathbf{C}^2{}_s\ (s)\|$ is the Chebisev norm of the second derivative of the original curve. As the parameter step is found the points on the curve are calculated by Horner scheme.

In rough cutting the the tool is moved along parallel lines either perpendicular or parallel to the axis of rotation, fromintersection to intersection either cutting metal or in air. The path should be traversed in an appropriate sequence to minmize inproductive motion.

The first step in the algorithm is the construction of the complete offset profile using the procedure described earlier. After that the comparison box is calculated for each curve segment to see whether it has an intersection with the current line or not [2]. The technique of the minimax box test is used so full advantage can be taken of the cheap division method by simplifying the conquer process. Those segments that might intersect, are transformed into their Bézier form, suitable for further subdivision. The Bézier form[6] is used instead of the B-spline form because the corresponding convex hull is smaller and therefore faster convergence is reached. The intersection point is localized by subdividing the curve and testing each portion of it until the interval is smaller than the required accuracy. By subdividing the curve's Bézier representation in the first iteration, at the extremal point, the intersections of the minimax boxes resulting in the subsequent steps contains only the curve segment's end points.

The Bézier segments are sorted in right to left order and dynamic limitation of the search space is used to speed up computation. A stack mechanism keeps track of the next cavity to be visited by the tool, and hereby minimizes the inproductive motion.

4 3D Algorithms for Milling

First the form of the path traced by the contact point has to be decided. From the wide variety of surface curves available, based on computational and technological consideration, the subsquent four strategies are used:

- Machining along the cross section lines of the final surface and planes, parallel to one of the principal planes of the coordinate system
- Machining along isoparametric lines
- Tracing curves generated by intersecting the objects with concentric cylinders
- Polyhedral machining

The first two strategies are described in the subsequent paragraphs.

In the first algorithm, a piecewise linear approximation of the cross section line of the patch [3] with the tool driving plane is obtained, then a normal at each cross point is computed, thus the location of the center of a ball end tool is easily calculated by shifting the touching point's coordinates along the surfac normal with ball radius. This step is repeated at the adjecent point on the cross section until the boundary of the patch is reached.

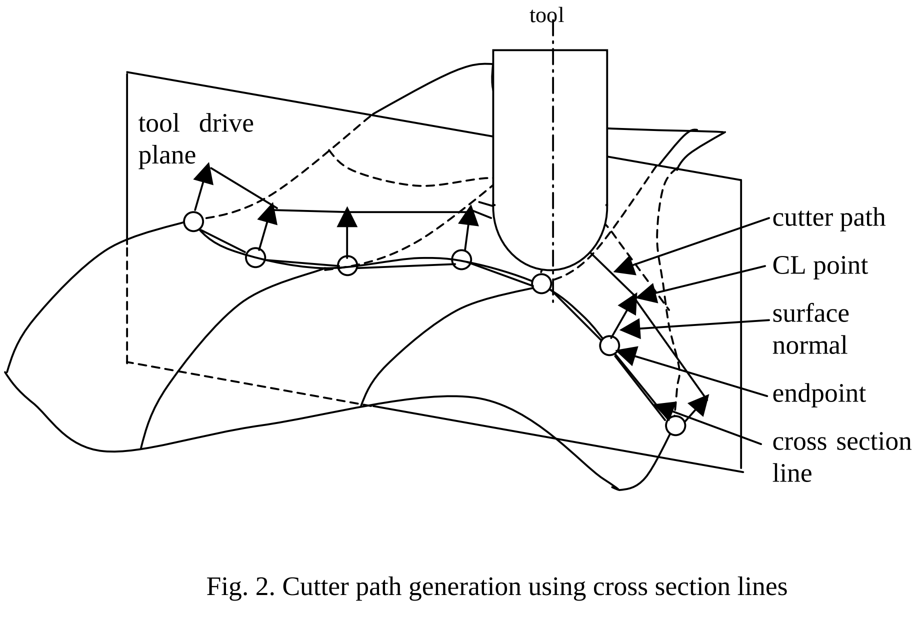

Fig. 2. Cutter path generation using cross section lines

Heap [1] published an algorithm which produces contour lines from a triangular mesh approximation of the surface. The mesh consists of a set of planar triangular elements whose vertices are points on the surface. For each cross section the algorithm follows the contour line segment line by line. As each triangle is found, the edges re intersected by the tool drive plane. A modification of this algorithm forms the basis of the new approach. Since the B-spline and rational B-splines are controlled by parameter values, any subregion of the patch can be calculated if the corresponding parameter values are known. Using the methode of plates from finite element analysis an estimate of the parameter step in both direction can be given according to formula (2):

$$\Delta s = \Delta t = \frac{tol}{4\sqrt{2}M} \qquad (2)$$

where

$$M = \max(\|D^2_s S_x\|,\|D^2_{st} S_x\|,\|D^2_t S_x\|) + \max(\|D^2_s S_y\|,\|D^2_{st} S_y\|,\|D^2_t S_y\|) + \max(\|D^2_s S_z\|,\|D^2_{st} S_z\|,\|D^2_t S_z\|)$$

Using this estimates a B-spline or rational B-spline surface can be triangulated. It is obvious that only those plane triangles are of interest which may intersect with the tool drive plane. It is practical to generate and intersect them one by one as it is required when traversing the cross section line. At the boundary of each triangle the surface normal is determined and the tool reference point offsetted accordingly. The diadvantage of this procedure is that the end points of the cross section line's piecewise approximation do not lay on the surface. This can be improved by subdividing the edges of the curved triangle and getting a more accurate approximation.

The second algorithm moves the tool along isoparametreic lines. Hereby the proble is reduced to one with a single degree of freedom. The surface curve is approximated by short line segments and the surface normal is calculated in the endpoints. The tools's reference point is computed again by moving the contact point on the surface normal by the tool radius.

The parameter step Δs was estimated according to formula (1) and the endpoints of the line segment is calculated again using Horner's scheme. In order to avoid undercut as a result of twisting surface, the distance between surface curve and the reference point's path is checked at the midpoint. If it is out of tolerance the parameter step is decreased and the calculation is repeated with the new value.

The segments of the tool path machining one patch can be connected to each other in three different way: one way, zig-zag and spiral cutting.

The algorithm ensures that the required tolerance is met in the direction of the tool movement. However, we have to take care of the distance between the adjacent tool paths. The surface roughness can deterogate due to the increaseíof the Euclidean distance between two neighbouring paths, due to change of the surface curvature.

In order to avoid these unwanted situations the following actions will be taken:

- If the distance between adjacent tool paths becomes unacceptably big, the distance will be changed to either ½or $^1/_3$ of the original according to the selected strategy. The beginpoint of the modified tool path segment is pushed on a stack. The new path will be followed until the boundary or to the point whre path became close enough again. At this point the coordinate values are taken from the stack and the machining will be resumed from that point
- If the distance became to small the path will be covered by high speed motion until the distance reaches the required level again

5 Collision Checking

Typically a surface to machined consists of a number of patches. The adjacent patches have up to a certain tolerance a common boundary. When the tangent's direction changes at the patch boundary, the tool may interfere with the adjacent patch and may damage the surface.

In CAD systems this problem is solved by computing the intersection curve of the offset surfaces and restrict the parameter domain to the area between them. This solution is time consuming and can't be afforded in real-time environment.

However by approximating the offset of bicubic patches by fifth order surfaces the computation can be reduced significantly. The points of the intersection curves are calculated by subdivision using the minimax box technique. This process is done for each pair of adjacent patches before the machining of the patch starts. The points of the boundary are then stored and used during the tool patch generation to check whether the next increment crosses the boundary or not.

6 Realization in Firmware

To implement the previously developed algorithms two ways have been used:

- The interpolation of the space curves is computed in the interpolator separately for the three axis. The interpolation is implemented in software on digital signal processor
- The restriction of the area for avoiding collision is calculated by a separeted processor working in a pipe line with the interpolator

With this structure the time needed for the calculation of the endpoints of the next line segment is less then 1 msec, while the calculation of the restricted area for one patch takes about 10 msec.

7 Experimental results

In order to evaluate the performance of the processor parabolic and elliptic surface have been described by spline surfaces and machined. For the purpose of reference the surface was machined by approximating the tool path with short line segments generated by the technological procesor. The machnined workpiece was used as a reference with which the results of the built in processor were compared.

Workpiece	**Version**	**Deviation (mm)**	**Rz(μ)**
Elliptic	Reference	0.025	22.0
Elliptic	Option	0.018	13.5
Parabolic	Reference	0.028	23.5
Parabolic	Option	0.022	20.5

Conclusions

The presented algorithms for a natural extension of the so-called contour programming and proved to be very advantageous both in terms of NC data and memory utilization. The part programmer describes the surfaces and curves in a compact form. This leads also to improved man –machine interface.

References

[1] Heap, B. R: Algorithms for Production of Contour Maps over an Irregular Triangular Mesh, National Physical Laboratory report NAC 10, Febr. 1972

[2] Koparkar, P.A., Modur, S.P.: Interval Methods for Processing Geometric Objects, IEEE on Computer Graphics and Applications Febr. 1984 p. 7-17

[3] Peng, Q.S.: An algorithm for finding the intersection line between two B-spline surfaces, CAD Journal Vol. 16 Number 4 July 1984 p. 191-196

[4] Prenter, P.M.: Splines and variational Methods, John Wiley, New York 1975

[5] Tiller, W., Hanson, E.G.: Offsets of Two-dimensional Profiles, IEEE on Computer Graphics and Applications Sept. 1984 p. 36-46

[6] Watt, A. 3D Computer Graphics addison-Wesley, Reading MA, 1999