A Curve-Based Approach for Clean-up Machining

Dong-Soo Kim[1], Cha-Soo Jun[2] and Sehyung Park[3]

[1]CSCAM, dscam@cscam.co.kr
[2]Gyeongsang National University, csjun@gsnu.ac.kr
[3]Korea Institute of Science and Technology, sehyung@kistmail.kist.re.kr

ABSTRACT

This paper presents a new efficient and robust tool-path generation methods for clean-up machining by employing a curve-based approach. The clean-up machining to be discussed in this paper are pencil-cut and fillet-cut for a polyhedral model of the STL form with a ball-end mill. The pencil-cut and fillet-cut paths are obtained from the curve-based scanning tool paths on the xz, yz, and xy planes. The scanning tool path has exact *sharp-concave points* and *bi-contact vectors*, both of which are very useful to detect 'pencil-points', trace the pencil-cut path, and generate the fillet-cut path. In the paper, some illustrative examples are provided, and the characteristics of the proposed method are discussed.

Keywords: clean-up machining, fillet-cut, pencil-cut, curve-based approach

1. INTRODUCTION

Tool-path generation (TPG) for clean-up is one of the most challenging problems in sculptured surface machining. The purpose of clean-up machining is to remove uncut volumes, which are left at concave regions after finish machining, by employing same or smaller size ball end-mills. Shown in Fig. 1(a) is the uncut volume left by a finishing cut with a ball-end mill of large size. The uncut volume may be cleaned up either by employing a pencil-cut (Fig. 1(b)) or by using a fillet-cut (Fig. 1(c)). In this paper, new TPG methods for pencil-cut and fillet-cut by employing a curve-based scanning approach are proposed. The methods use polyhedral models of the STL (Stereolithography) form as part surfaces since geometric processing with the polyhedral model is very robust and the models are very popular in CAD/CAM applications.

A number of commercial CAD/CAM systems including CATIA, UG, and Work-NC are capable of generating clean-up tool paths, but a very few methods [3,6,7,8] have been published in open literature. Some characteristics of the proposed approach over the previous methods are discussed in detail in chapter 3 of the paper. In order to find '*pencil points*' of pencil-cut path and fillet-cut path in this method, scanning tool paths are generated using '*curve-based approach*' developed by Jun et al.[4]. The curve-based scanning approach is described briefly in the following chapter. Then, the systematic methods for detecting and tracing pencil-cut paths and generating fillet-cut path are presented in chapter 3.

2. CURVE-BASED SCANNING APPROACH

2.1 Point-Based vs. Curve-Based Approach

One of the critical problems in generating tool-paths for a compound surface is cutter gouge (or interference). A lot of research works [1-9] has been done to solve the problem. There are several previous studies related to the gouge-free tool path generation for a polyhedral model, including those by Duncan and Mair [1] and Hwang [2]. The polyhedral model based TPG has been called *'Polyhedral machining'* since Duncan and Mair invented the term in their works [1].

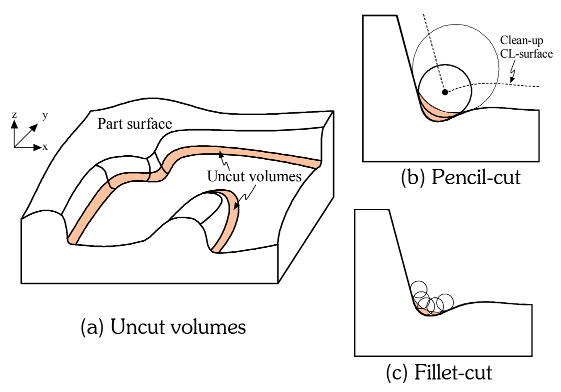


Fig. 1. Tool path generations for clean-up machining [3].

In the previous polyhedral machining method [1], for a given cutter-center axis (x_1, y_1), the *highest cutter position* z_1 is determined by lowering the cutter until it touches a triangular facet under the cutter, as shown in Fig. 2(a). Another method finds the intersection points of *inverse tool offset surfaces* with the cutter axis, and takes the highest point as the current cutter position [3] (Fig. 2(b)). The next cutter position (x_2, y_2, z_2), which is distant by an adequate *step-length* from the current position (x_1, y_1, z_1), is calculated by the same process. The cutter moves linearly from position (x_1, y_1, z_1) to position (x_2, y_2, z_2). Hereinafter, these methods are referred to as a *point-based scanning approach*. The point-based approach has inherent limitations in the accuracy of convex gouge handling and in identifying *sharp-concave points* (Fig. 2(c)).

The method adopted in this paper is referred to as a *curve-based scanning approach*. In this approach, a tool path is generated as a compound curve, instead of a sequence of points. Conceptually, this type of tool path can be obtained by intersecting the offset surface (or CL-surface [3]) with drive planes. However, it is very complicated and time-consuming to construct the offset surface of a polyhedron model exactly and completely. In this work, the offset elements of the model are calculated uniquely in convex regions, but the ones in concave regions are overlapped. The overlapped portions are removed after the intersection curves are obtained. This approach is simple and efficient. It can also overcome the limitations of the point-based scanning approach.

2.1 Scanning Path Generation from Polyhedral Model

The overall procedure of generating tool paths was

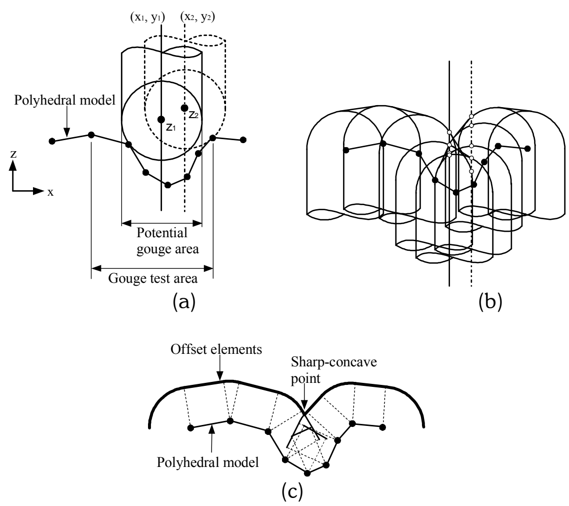
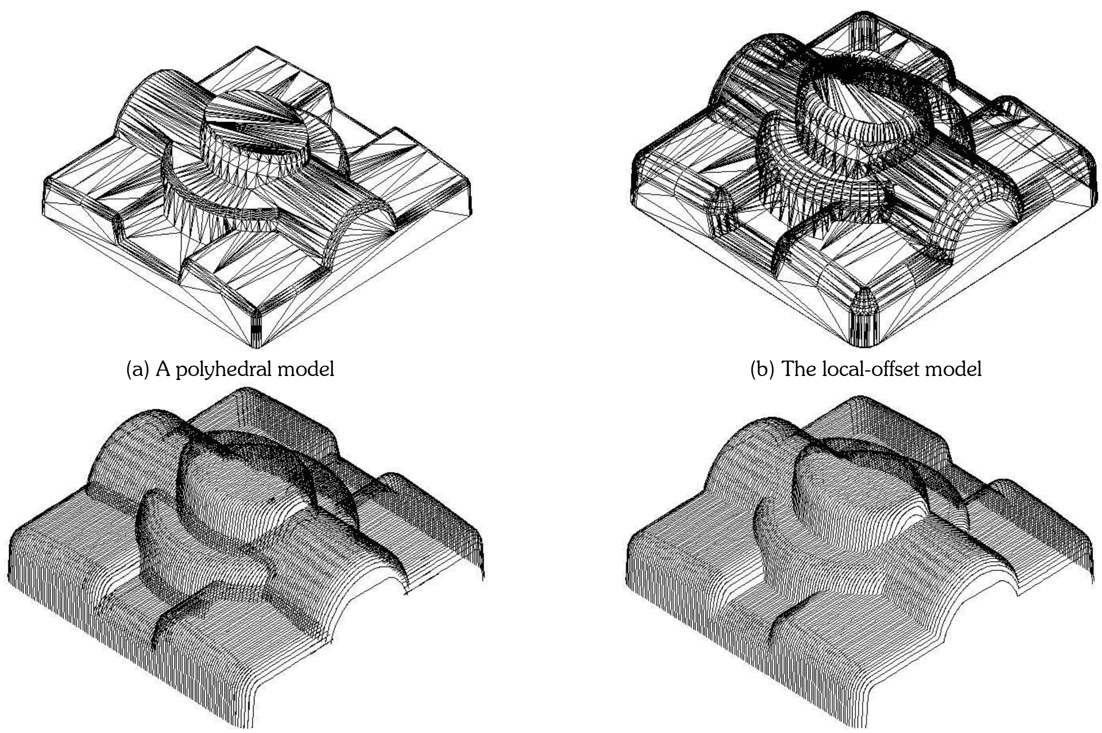


Fig. 2. Point-based approaches and a curve-based approach for



(a) A polyhedral model

(b) The local-offset model

(c) Untrimmed CL-curves

(d) CL-paths

Fig. 3. An example of scanning tool path generation from a polyhedral model.

a polyhedral model (2D view).

introduced in the previous paper [4]. The input data of the procedure are a polyhedral model of the STL form and the machining conditions. The output is CL-paths of quadratic rational Bezier curve form. The polyhedral model is offset by a *local-offsetting scheme* [4]. Based on the local-offsetting scheme, all the facets of the polyhedron, convex edges and convex vertices are offset to generate the offset surface. Then, the offset elements such as triangular facets, trimmed cylinders, and trimmed spheres are sliced by a series of drive planes. The curve segments on a drive plane are sorted, trimmed and linked, while the concave gouge is removed during the trimming process. The CL path is not a point sequence but a composite curve consisting of line segments, circular arcs, and elliptic arcs, which are free of concave and convex gouges. At the junction point of two adjacent CL-curve segments, *bi-contact vectors*, two normal vectors of the corresponding offset surface elements, are calculated and stored. The vectors are very useful to find pencil-cut paths. Fig. 3 shows an example of scanning TPG from a polyhedral model.

3. TOOL PATH GENERATION FOR PENCIL-CUT

3.1 Overall Procedure

The overall procedure of generating tool-paths for the pencil-cut and fillet-cut is summarized in Fig. 4. Input data for the pencil-cut consists of a polyhedral model of the STL form and machining conditions. The scanning and contouring tool paths are used to find pencil-cut path, and the pencil-cut paths are used as a guide curve for generating the fillet-cut paths.

For a given polyhedral model as a part surface geometry and a ball-end mill, the pencil-cut paths are generated in the following three steps: (1) Construct a clean-up CL-surface with x- and y-directional scanning tool paths [4], and z-level contouring tool paths [5]; (2) Detect pencil-points among the sharp-concave points; (3) Trace the pencil-cut path considering the angle and distance factors from the pencil points.

Fig. 5 shows the clean-up CL-surface that is constructed by combining x- and y-directional CL-paths with z-level CL-paths. Note that the junction points of the paths have bi-contact vectors.

3.2 Detecting pencil-points

There are two main advantages of the proposed pencil-cut path generation approach over the previous approaches: First, the sharp-concave points, which are the candidates of the pencil-points, are calculated exactly by the proposed approach as shown in Fig. 6. The most previous approaches construct the clean-up CL surface by using the point-based scanning methods [3,6], which have an inherent limitation in identifying the sharp-concave points.

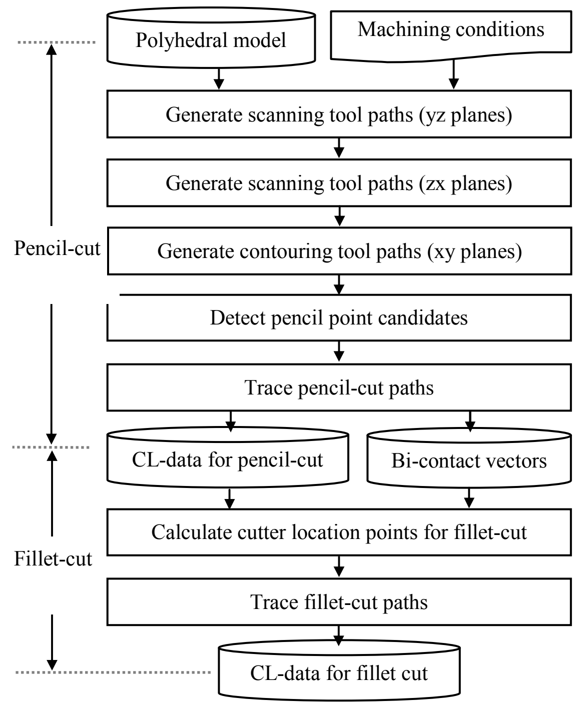


Fig. 4. Overall procedure of CL data generation for pencil-cut and fillet-cut.

Therefore, the pencil-points are '*guessed*' and are '*smoothed*' through the sophisticated processes as shown in Fig. 7.

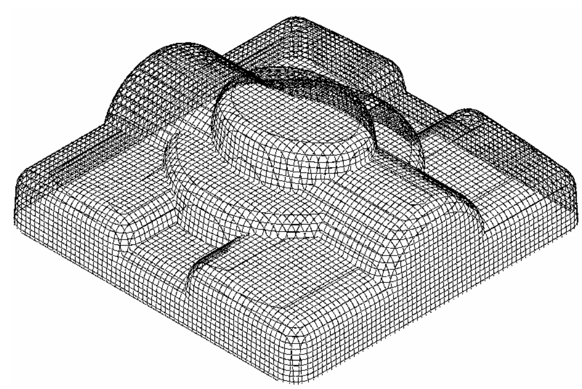


Fig. 5. Clean-up CL-surface.

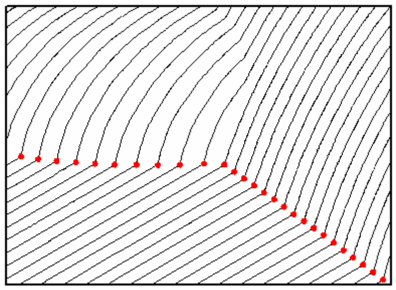
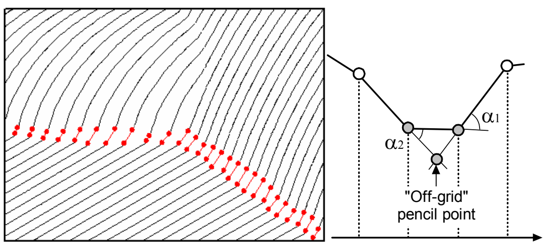
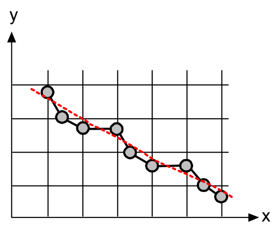


Fig. 6. Pencil-points by the curve-based scanning approach.



(a) Pencil-points



(b) Pencil-cut path fairing

Fig. 7. Pencil-cut TPG by a point-based scanning method [3].

Second, the proposed approach selects the 'pencil points' among the 'sharp-concave' points by using their bi-contact vectors. As shown in Fig. 8, the 3D bi-contact angle β between the two bi-contact vectors $\mathbf{b}_1$, $\mathbf{b}_2$ is the best criterion for estimating uncut area and then identifying the pencil-points. If a bi-contact angle of a concave point is greater than a given angle, the concave point is considered as a pencil point. In the previous methods, the pencil-point is determined by the 2D concave-angle α between line segments of the path in the drive plane as depicted in Fig. 7(a). The 2D angle α, which is equal to or greater than the 3D contact angle β, may cause to select the improper pencil-points as shown in Fig. 9.

3.3. Tracing a pencil-cut path

Starting from an arbitrary initial pencil-point, a complete pencil-path can be traced by repeatedly marching toward the next pencil-point until there is nowhere left to go. Fig. 10(a) shows an example of pencil-cut path tracing.

Suppose that the current point is A and a next candidate point B on a grid line, an estimated error bound between A and B is calculated approximately as follows (Fig. 10(b)):

$$h = \frac{d \sin\theta_A \sin\theta_B}{\sin(\theta_A + \theta_B)} \qquad (1)$$

where, d is distance between the points A and B, $\mathbf{t}_A$ and $\mathbf{t}_B$ are unit tangent vectors at A and B respectively, and θ_A and θ_B are angles between the tangent vectors and line AB. The tangent vectors are calculated by cross product of the bi-contact vectors at the points. If the estimated error bound h is less than the given tolerance, point A and B are linked directly. Otherwise, the pencil-cut path is traced by the intersection of the offset elements between A and B.

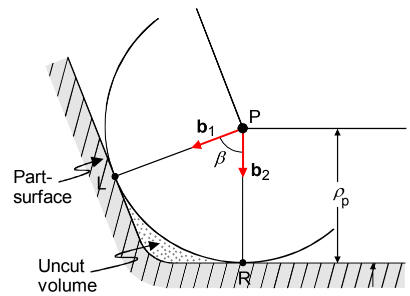


Fig. 8. Bi-contact vectors.

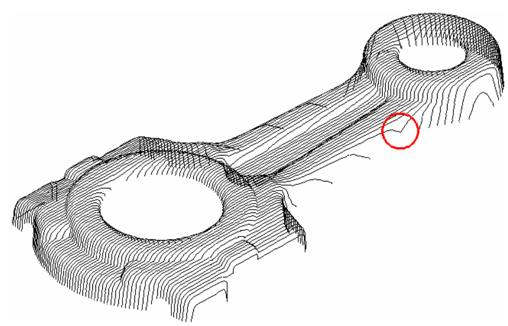
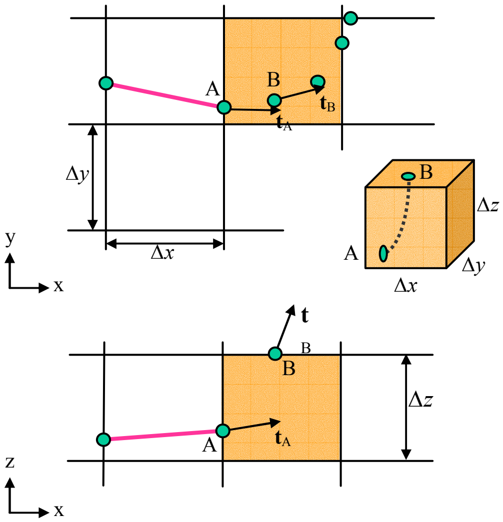
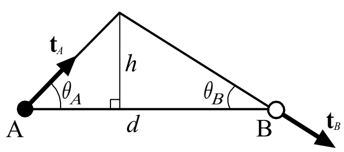


Fig. 9. An improper pencil-point by the concave angle criterion.



(a)



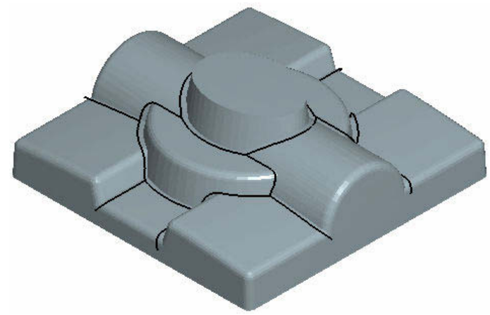
(b)

Fig. 10. Pencil-cut path tracing.

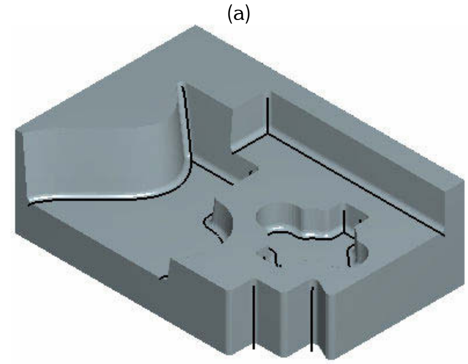
If a pencil-cut curve segment passes through the scanning plane box depicted in Fig. 10(a), the curve can be detected completely by our method. As the plane interval increases, the CL-surface construction and detection time decrease, but the path tracing time may increase.

3.4. Illustrative Examples for Pencil-Cut TPG

Some illustrative examples of the pencil-cut paths are provided. Fig. 11 shows pencil-cut paths for two polyhedral models consisting of 3,378 facets and 9,156 facets, respectively. The nearly vertical pencil curves can be found stably by the proposed approach. Fig. 12 shows pencil-cut paths for the model (with 454,273 facets) acquired by a 3D scanner. Fig. 13 shows another example of pencil-cut paths and machined parts before and after the pencil-cut.



(a)



(b)

Fig. 11. Pencil-cut paths for the compound surfaces.

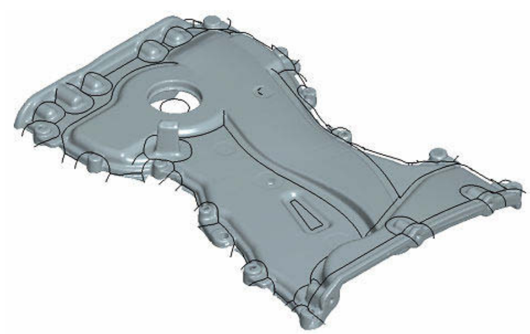
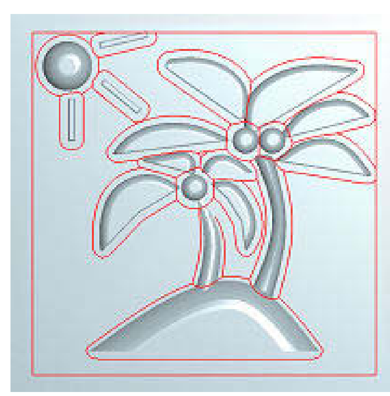
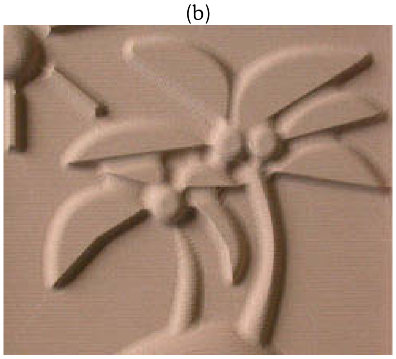


Fig. 12. Pencil-cut paths for a scanned part.



(a)

(b)



(c)

Fig. 13. A pencil-cut path and machined parts (18,082 facets) before and after the pencil-cut.

The computational times to generate pencil-cut paths for the examples are tested on a 2.66GHz Pentium IV machine and are summarized in Tab. 1.

Cutter diameter (5mm)

Model	Path interval (mm)	Computational time (seconds)
Fig. 11(a)	0.5	17.1
	0.1	69.3
Fig. 11(b)	0.5	11.8
	0.1	49.2
Fig. 13	0.5	32.4
	0.1	130.3
Fig. 12	0.5	249.7
	0.1	1,046.2

Tab. 1. Summary of tool path generation times.

4. TOOL PATH GENERATION FOR FILLET-CUT

This section presents a method generating fillet-cut paths, which are an essential requirement in die-cavity machining.

A fillet-cut path is generated by post-processing the pencil-cut path. Shown in Fig. 14 are two examples of the pencil-cut paths and bi-contact vectors.

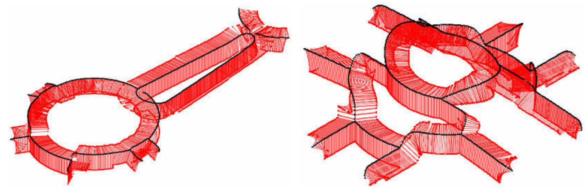


Fig. 14. Pencil-cut paths and bi-contact vectors of pencil points.

With the two bi-contact vectors $\mathbf{b}_1$ and $\mathbf{b}_2$, the first step to fillet-cut path generation is to calculate a number of '*ray vectors*' by rotating the vector $\mathbf{b}_1$ to the vector $\mathbf{b}_2$. As depicted in Fig. 15, the ray vectors $\{\mathbf{v}_j \mid j = 0, 1, \ldots, n\}$ may be obtained from

$$\mathbf{v}_j = R(\mathbf{t}, \theta_j)\mathbf{b}_1, \quad j = 0, 1, \ldots, n \tag{2}$$

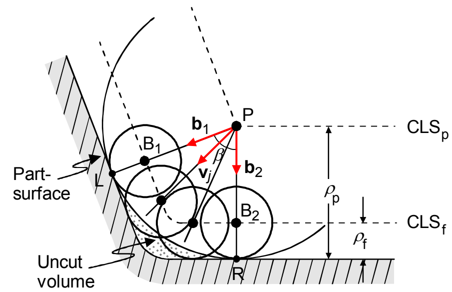


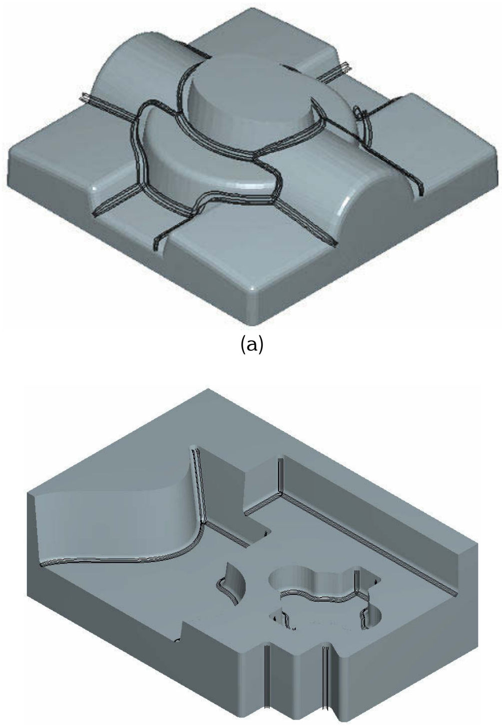
Fig. 15. Basic construction for generation of fillet-cut CL-paths.

where $(n+1) \geq 2$ denotes the number of ray vectors, $R(\mathbf{t}, \theta_j)$ is a 3D rotational matrix around the tangent vector $\mathbf{t}$ = normalize($\mathbf{b}_1 \times \mathbf{b}_2$) by angle θ_j = □j/n, and □ is the bi-contact angle. Furthermore, each of the ray vectors in (2) defines a ray line passing through the pencil-point P. A parametric equation of the line may then be expressed as

$$\mathbf{r}(t) = \mathrm{P} + \mathbf{v}_j t \tag{3}$$

The individual CL-points for the fillet-cut are obtained by intersecting the fillet-cut CL-surface with the ray lines given by (3).

As a final step, the CL-points are linked to form the fillet-cut paths along the corresponding pencil-cut paths. Examples of fillet-cut tool-paths are shown in Fig. 16.



(a)

(b)

Fig. 16. Examples of fillet-cut tool-paths.

5. CONCLUSION

A new pencil-cut and fillet-cut TPG approach is proposed in this paper. The main features of the proposed approach are as follows: (1) The pencil-cut and fillet-cut paths even for the vertical edges can be found stably by constructing CL-surface with xz, yz and xy plane scanning tool paths; (2) The sharp-concave points and bi-contact vectors are exactly calculated by employing the curve-based scanning and contouring methods; (3) The points and vectors are very useful to detect pencil-points, to trace pencil-cut paths and to generate fillet-cut paths. The process to generate pencil-cut and fillet-cut path is clear to understand, robust and efficient and the quality of the result path is high.

6. ACKNOWLEDGEMENTS

This study was supported financially by the Korea Science and Engineering Foundation through the Research Center for Aircraft Parts Technology (ReCAPT), Gyeongsang National University, Korea.

7. REFERENCES

[1] Duncan, J.P. and Mair, S.G., *Sculptured Surfaces in Engineering and Medicine*, Cambridge University Press, 1983, USA.

[2] Hwang, J.S., Interference-free tool-path generation in the NC machining of parametric compound surfaces, *Computer-Aided Design*, Vol. 24, No. 12, 1992, pp. 667-676.

[3] Choi, B. K. Jerard, R. B., *Sculptured surface machining: theory and applications*, Kluwer Academic Publishers, 1998.

[4] Jun, C.S, Kim, D.S. and Park, S., A new curve-based approach to polyhedral machining, *Computer-Aided Design*, 34(5), 2002, 379-389.

[5] Kim, D.S., *A curve-based approach to polyhedral model machining*, Ph.D. Thesis, Gyeongsang National University, 2002, Korea.

[6] Park, J.W., Kim, B.H. and Choi, B.K., Pencil curve tracing via virtual digitizing, *Trans. of the Society of CAD/CAM Engineers*, 2(4), 1997, pp. 253-266.

[7] Flutter, A. and Todd, J., A machining strategy for toolmaking, *Computer-Aided Design*, Vol. 33, No. 13, 2001, pp. 1009-1022.

[8] Choi, B.K. and Jun, C.S., Ball-end cutter interference avoidance in NC machining of sculptured surfaces, *Computer-Aided Design*, Vol. 21, No. 6, 1989, pp. 371-378.

[9] Ren, Y., Yau, H.T. and Lee, Y.S., Clean-up tool path generation by contraction tool method for machining complex polyhedral models, *Computers in Industry*, Vol. 54, No. 1, 2004, pp. 17-33.