

Uncut free pocketing tool-paths generation using pair-wise offset algorithm

S.C. Park, B.K. Choi*

CAM Laboratory, Department of Industrial Engineering, Korea Advanced Institute of Science and Technology (KAIST), 373-1 Kusong-dong, Yusong-gu, Taejeon 305-701, South Korea

Received 20 March 2000; revised 2 August 2000; accepted 7 August 2000

Abstract

For die-cavity pocketing, contour-parallel machining is the most popular machining strategy. Two issues in generating contour-parallel tool paths for pocketing are: (1) a robust two-dimensional (2D)-curve offsetting algorithm; and (2) detecting and removing uncut regions. The 2D-curve offsetting solution has been widely studied, because it has so many potential applications. However, though the importance of the uncut problem in pocketing has long been recognized there have been few reported investigations on detecting and removing uncut regions. This paper presents a procedure for generating pocketing tool paths leaving no uncut regions. For the 2D-curve offsetting algorithm, we employ the *PWID offset algorithm* suggested by Choi and Park (Computer Aided Design, 31(12) (1999) 735) and expand the algorithm for offsetting areas having islands. Based on the *expanded PWID offset algorithm*, our solution to the uncut problem removes the uncut regions. Empirical tests show the usefulness of the proposed procedure for improving the productivity of pocket machining. © 2001 Elsevier Science Ltd. All rights reserved.

Keywords: Pocketing; Contour-parallel tool paths; PWID offset; Uncut removal

1. Introduction

The most popular method for machining a two-dimensional (2D) pocket is the contour-parallel offset method. In this paper, the term ‘contour-parallel machining’ is used to refer to pocketing with contour-parallel tool paths. As shown in Fig. 1, contour-parallel machining uses successive offsets of the boundary curve as tool-path elements. Thus, the 2D-curve offsetting problem has been regarded as the key issue for the generation of contour-parallel tool paths [3]. However, we should note that the productivity of contour-parallel machining is mainly dependent on the tool-path interval, because an increase in the tool-path interval brings a decrease in the total length of the tool paths. Uncut regions may result when the tool-path interval is larger than the tool radius [3,5,8] (Fig. 1). Thus, to improve the productivity of contour-parallel machining, the generation of contour-parallel tool paths without uncut regions is essential.

The 2D-curve offsetting algorithm has been widely studied because of its many applications. In the literature,

the offsetting problem has been approached from three different directions: the pair-wise offset, Voronoi diagram and pixel-based approaches. The conventional *pair-wise offset approach* [6,7,13] consists of four steps: (1) an *offset segment* is generated for each element of a *boundary curve*; (2) a *raw offset curve* is constructed by closing the gaps with trimming arcs; (3) all pair-wise self-intersections in the raw offset curve are detected; and (4) all *invalid loops* are removed. Fig. 2 shows the basic geometric entities related to the pair-wise offset method. The thick curve denotes a boundary curve and the thin curve its *raw offset curve*. A raw offset curve is divided into *loops* at its *self-intersection points*, and a loop that has to be removed to obtain valid offset curves is called an *invalid loop*. As shown in Fig. 2, there are two types of invalid loop: a *local invalid loop* bounded by a single self-intersection point and a *global invalid loop* bounded by a pair of self-intersection points. The pair-wise offset method is intuitive and simple. As pointed out by Held et al. [8], however, the intersection-detection step (Step 3) is time consuming, and the loop-removal step (Step 4) is prone to numerical errors. The *Voronoi diagram method* [8–10] is known to be more efficient and robust, but it may also suffer from numerical instability, such as the near-circular singularity [1,11]. The merit of the *pixel-based methods* [3,12] is robustness, but

* Corresponding author. Fax: +82-42-861-1692.

E-mail addresses: psc@vmslab.kaist.ac.kr (S.C. Park), bkchoi@bezier.kaist.ac.kr (B.K. Choi).

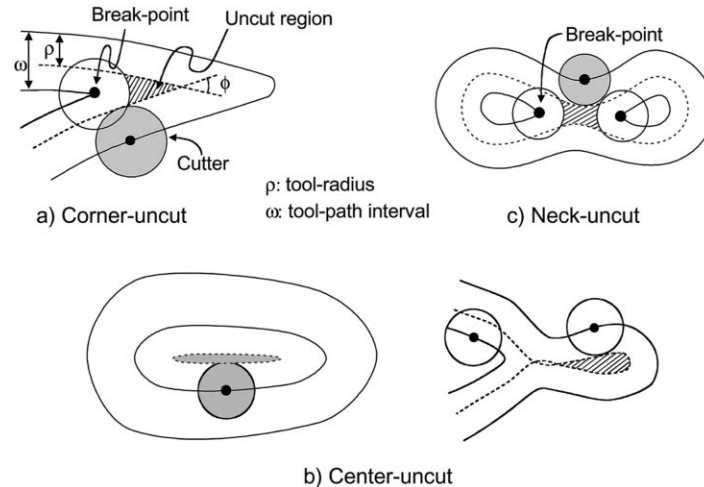


Fig. 1. Types of uncut region in pocket machining.

they require a large amount of memory as well as an excessive computation time to achieve a desired level of precision. Recently, Choi and Park [1] proposed a pair-wise offset algorithm, called the *PWID offset algorithm*, to cope with the difficulties of the conventional pair-wise offset approach. One of the salient features of the PWID offset algorithm is that it removes all *local invalid loops* before constructing a raw offset curve by invoking a *pair-wise interference detection (PWID)* test. As a result, the PWID offset algorithm avoids the near-circular singularity causing the numerical instability.

The importance of the uncut problem in contour-parallel machining has long been recognized; however, there have been few reported investigations on detecting and removing uncut regions. Held et al. [8] suggested a partial solution for the simplest case of the uncut problem where a sharp corner is formed by two lines. Choi and Kim [3] classified the types of uncut region (shown in Fig. 1) and suggested a solution for detecting and removing uncut regions, but it is based on the pixel-based offset approach requiring a large amount of memory as well as an excessive computation time to achieve a desired level of precision.

The procedure presented in this paper generates contour-parallel tool paths guaranteeing no uncut regions. The problem of generating contour-parallel tool paths can be informally specified as follows:

Contour-parallel tool-paths generation for pocketing

- Input: boundary curves defining a machining area, tool-path interval and tool radius.
- Output: contour-parallel tool paths without uncut regions.

We assume that the boundary curves are point-sequence curves (PS-curves) consisting of a large number of points. Practically, the boundary curves are likely to be given in PS-curves from the surface–surface intersection (cutter-location (CL)-surface and plane) or feature extraction from a

CL-surface. For the sake of simplicity, let us assume that the given boundary curves are defined on the CL-surface [4,5] which is defined by the ‘Minkowski sum’ of the part surface and the inverse cutter as shown in Fig. 3. As a result, the boundary curves can be used as tool paths without compensating the tool volume.

For the 2D-curve offsetting algorithm, we employ the *PWID offset algorithm* because of its robustness and efficiency. The original PWID offset algorithm was developed for offsetting a closed PS-curve having no “islands” (island PS-curves have to be manually bridged to the boundary PS-curve to make them a single closed PS-curve.). For generating the contour-parallel tool paths, we expand the *PWID offset algorithm* to offset areas having islands (Fig. 4). Based on the *expanded PWID offset algorithm*, our solution detects and removes uncut regions.

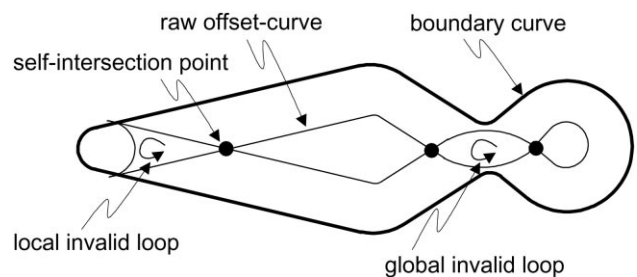


Fig. 2. Basic geometric entities in the pair-wise offset approach.

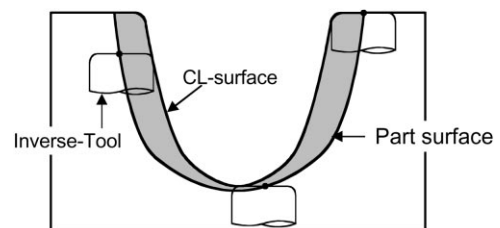


Fig. 3. CL surface: Minkowski sum of part surface and inverse tool.

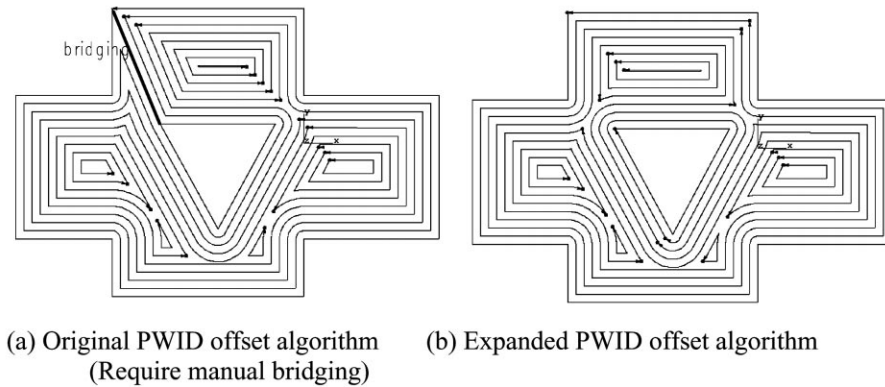


Fig. 4. Expansion of the PWID offsetting algorithm.

The overall structure of the paper is as follows. Section 2 addresses the summary of the *PWID offset algorithm* and its expansion for offsetting areas having islands. Based on the *expanded PWID offset algorithm*, the procedure for generating contour-parallel tool paths without uncut regions is described in Section 3. Illustrative examples and performance tests are given in Section 4. Section 5 contains concluding remarks.

2. Expansion of the PWID offset algorithm

Before explaining the expansion of the PWID offset algorithm, we summarize the algorithm for offsetting a closed PS-curve having no “islands”. To help understand the summary, we explain some terms. A point of a PS-curve becomes an *interfering point* if its *tangential circle* intersects with the PS-curve (Fig. 5). A range of interfering points is called an *interfering range*. An interfering range corresponding to a local invalid loop is called a *local interfering range* (LIR) and an interfering range belonging to a global invalid loop is called a *global interfering range* (GIR).

As mentioned above, one of the salient features of the PWID offset algorithm is that all LIRs are removed before constructing a raw offset curve by invoking a PWID test. During the PWID test, each pair of elementary offset segments is tested for interference and then the *interfering segments* are successively removed. The resulting raw offset

curve, which now contains only GIRs (if any), is subjected to a sweep-line algorithm [2] to find all self-intersections. If global invalid loops are detected, they are removed by invoking the PWID test again. The PWID test, which is the heart of the algorithm, is a function for removing LIRs and GIRs with convex vertices and self-intersection points, respectively. The overall procedure of the *PWID offset algorithm* is as follows:

1. Remove all LIRs by applying PWID tests with convex vertices on the PS-curve (Fig. 6b).
2. A raw offset curve is constructed from the resulting boundary PS-curve (Fig. 6c).
3. All the pair-wise self-intersections in the raw offset curve are detected (Fig. 6d).
4. Remove all GIRs by applying PWID tests with the intersection points (Fig. 6e).
5. Valid offset curves are obtained from the final PS-curve (Fig. 6f).

For generating contour-parallel tool paths, the PWID offset algorithm needs to be expanded for offsetting areas defined by external and internal curves (areas having islands) as shown in Fig. 7. Let us assume that the orientations of the external and internal curves are CCW (counterclockwise) and CW (clockwise), respectively.

Steps 1, 2 and 5 of the PWID offsetting procedure described above have no need to be changed for the

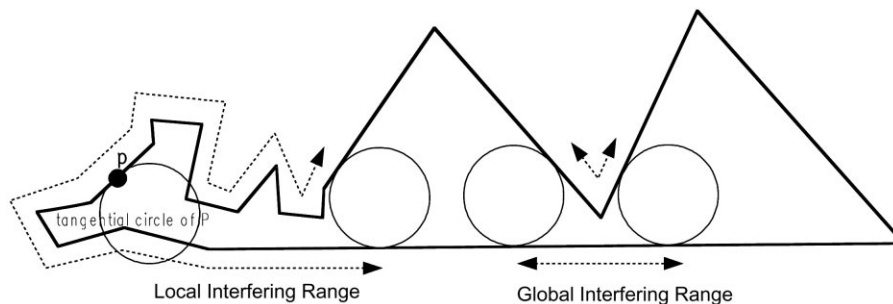


Fig. 5. Interfering point and interfering range (LIR, GIR).

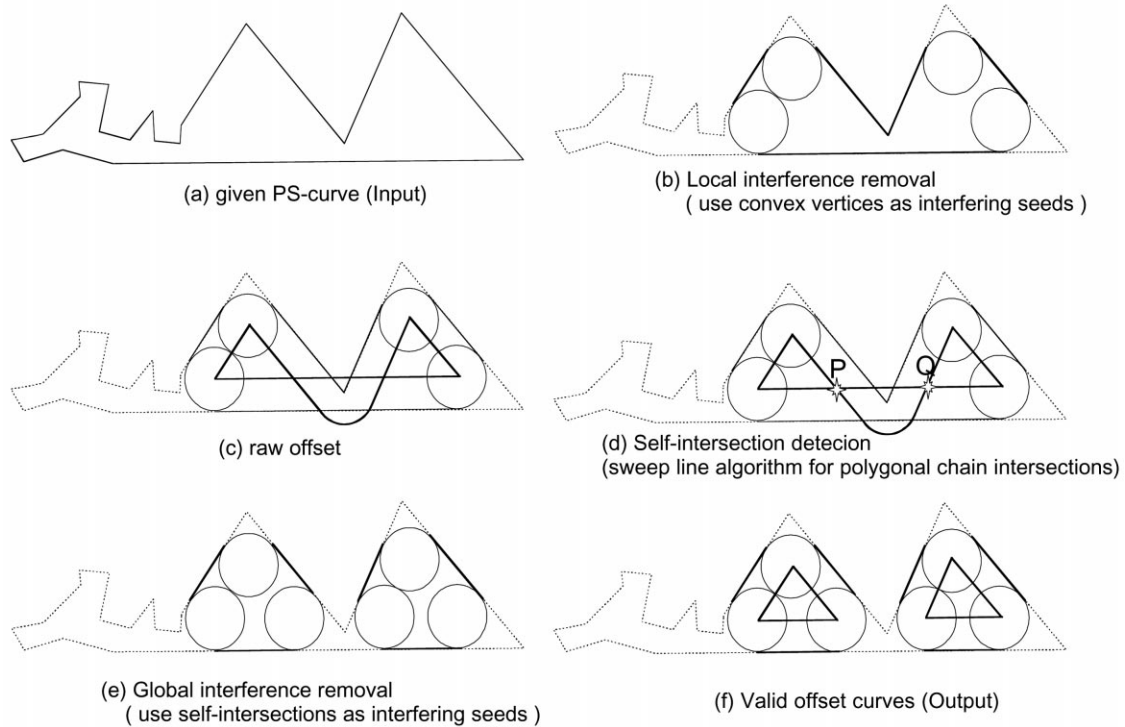


Fig. 6. The procedure of the PWID offsetting algorithm.

expansion because they are independent operations working with a single PS-curve. In step 3, however, it is necessary to find all intersections among raw offset curves including self-intersection points; that is performed by the sweep line algorithm [2]. To remove GIRs in step 4, before applying the PWID tests, we need to check the inclusion relationship between the raw offset curves. As shown in Fig. 8b, some raw offset curves may fully belong to a GIR and cannot be removed by the PWID test, because there is no intersection point to start the PWID test. Every internal curve has its corresponding external curve. If an internal raw offset curve does not have intersection points with its external raw offset curve, then we must check the inclusion relationship between the internal and external raw offset curves. Both of the internal and external raw offset curves should be removed if the inclusion relationship is reversed, i.e. the

internal raw offset curve contains the corresponding external raw offset curve. Thus, the overall procedure of the *expanded PWID offset algorithm* is as follows:

1. Remove all LIRs by applying PWID tests with convex vertices on the curves defining the offset area (Fig. 8a).
2. Raw offset curves are constructed from the resulting curves (Fig. 8b).
3. Find all intersections among the raw offset curves including self-intersections (Fig. 8b).
4. Remove raw offset curves belonging to GIRs by checking the inclusion relationship (Fig. 8c). Remove all GIRs by applying PWID tests with the intersection points (Fig. 8d).
5. Valid offset curves are obtained from the final curves.

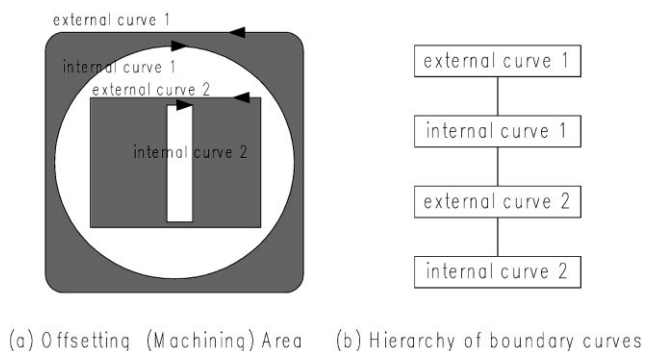


Fig. 7. Area defined by external and internal curves.

3. Tool paths without uncut regions

In this section, we present a solution based on the *expanded PWID offset algorithm* to remove the uncut regions. Note that the uncut problem appears when the tool-path interval (ω) is larger than the tool radius (ρ) and smaller than two times the tool radius ($\rho < \omega \leq 2\rho$). As the tool-path interval increases the total length of the tool paths decreases; however, the tool-path interval larger than the tool radius may leave uncut regions. The types of uncut region classified by Choi and Kim [3] are depicted in Fig. 1: corner uncut, center uncut and neck uncut.

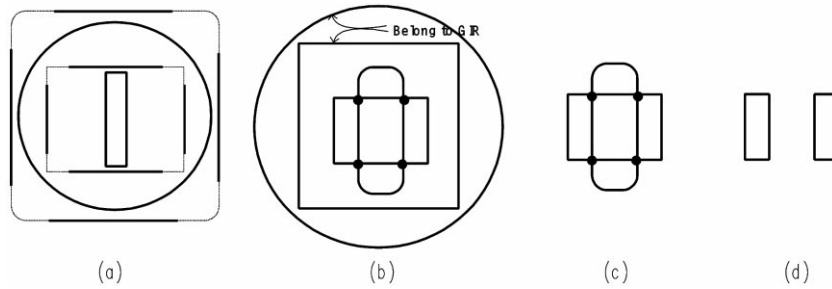


Fig. 8. Expanded PWID offsetting procedure for the area in Fig. 7.

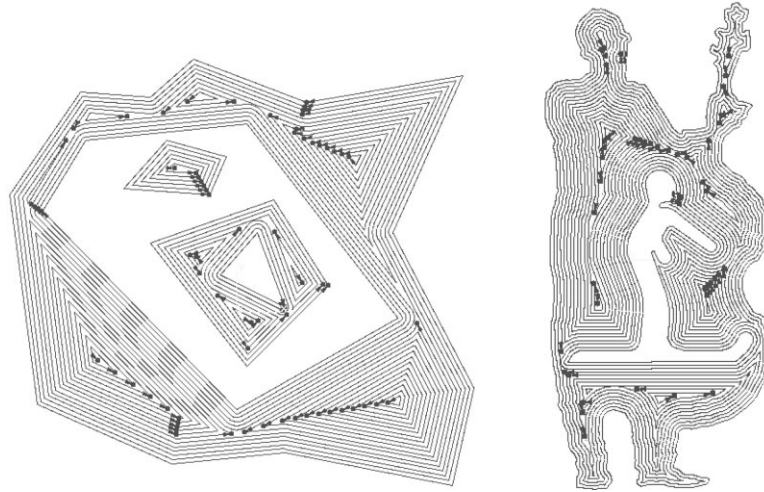


Fig. 9. Examples of the expanded PWID offsetting algorithm.

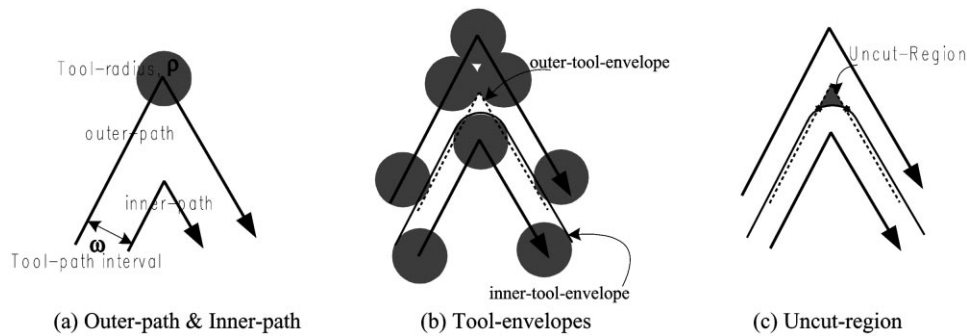


Fig. 10. Tool envelopes and uncut regions.

For generating contour-parallel tool paths, we first offset a machining area by employing the *expanded PWID offset algorithm* (Fig. 9). Uncut regions may exist between two successive tool paths, for example the inner path and outer path in Fig. 10. Let the inward offset curve of the outer curve with the offset distance p (tool radius) be the *outer tool envelope* and the outward offset curve of the inner curve with the offset distance p be the *inner tool envelope*, as shown in Fig. 10. Intuitively, we know that uncut regions exist, if and only if there are intersections between the outer tool envelope and the inner tool envelope. Based on this simple observation, we could say that detecting uncut

regions requires two offsetting operations for the envelopes (outer tool envelope and inner tool envelope) and one Boolean operation between the envelopes.

To improve the efficiency of detecting uncut regions, we make use of the concept of the interfering ranges, LIR and GIR, of the *expanded PWID offset algorithm*. As shown in Fig. 11a, the inner path is the offset curve of the outer path with the offset distance ω (tool-path interval). Let the range of the outer tool envelope corresponding to an interfering range of the outer path be a *clean-up curve* (Fig. 11b) and the vertex of the inner path corresponding to the interfering range be the *degen-*

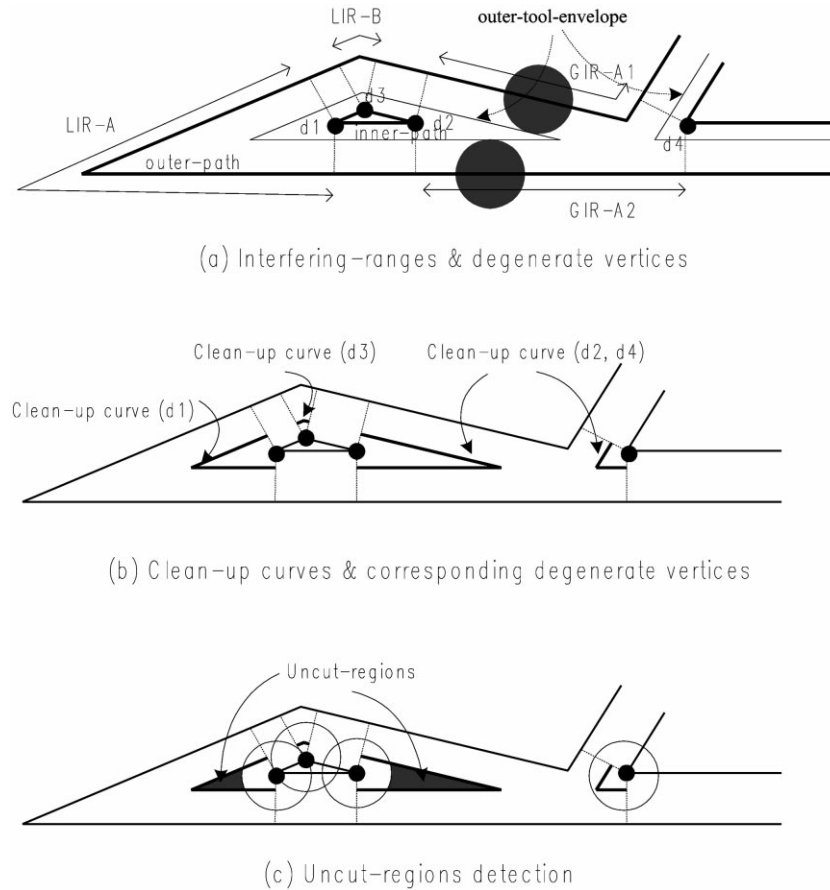


Fig. 11. Interfering ranges and uncut regions.

erate vertex of the *clean-up curve*. We should observe that if there are no interfering ranges of the outer path then uncut regions will not exist; in other words, uncut regions are caused by interfering ranges: an LIR may cause a corner or center uncut region and a GIR may cause a neck or center uncut region. Based on this observation, it is enough to consider the *clean-up curves* (the ranges of the outer tool envelope corresponding to the interfering ranges of the outer path) for detecting uncut regions. A clean-up curve causes an uncut region, if and only if the maximum distance from the corresponding degenerate vertex is larger than the tool radius ρ , as shown in Fig. 11c. Thus, detecting uncut regions requires offsetting operations for the clean-up curves and distance checking operations between the clean-up curves and the corresponding degenerate vertices.

If there are uncut regions, then we append clean-up tool paths to remove the uncut regions; otherwise, the offset curves can be directly employed for the contour-parallel tool paths. The clean-up curves causing uncut regions may be employed for the clean-up tool paths. Note that the ranges of the clean-up curves having a distance larger than ρ (tool radius) from the corresponding *degenerate vertex* are enough for the clean-up tool paths. Fig. 12 shows the appending of the clean-up tool paths according to the types of the uncut region. Because a

rapid change of feed direction may cause machining errors, we append the clean-up tool paths to relieve the rapid change of feed direction.

4. Illustrative examples and tests

The proposed pocketing tool paths generation procedure was implemented and tested with various examples. Fig. 13 shows stamping dies for the door and fuel tank of a passenger car. The area curves (the thick curves in Fig. 13a and b) were obtained by intersecting the CL-surfaces with a horizontal plane and they have 3500 and 4000 points, respectively.

Fig. 14 shows the generated tool paths guaranteeing no uncut regions, with the tool-path interval 15 mm and the tool radius 10 mm in an engineering work station with the computation times of the tool paths 1.02 and 1.87 s in Fig. 14a and b, respectively. The computation time includes the offsetting time of the expanded PWID offsetting algorithm and the calculation time of the clean-up tool paths for removing uncut regions. The time complexity of the suggested procedure is overwhelmed by the PWID offsetting algorithm whose time complexity is near $O(n)$, where n is the number of points in the boundary curves. For a detailed documentation of the time complexity of the

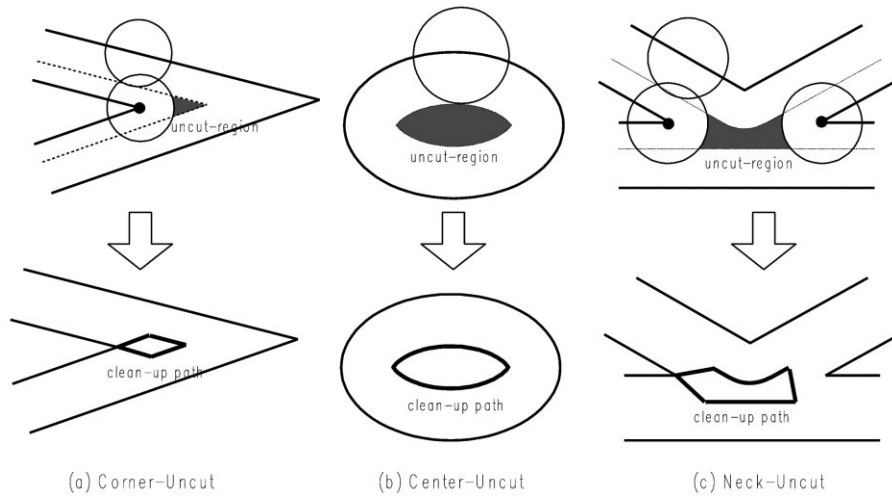


Fig. 12. Uncut regions and clean-up tool paths.

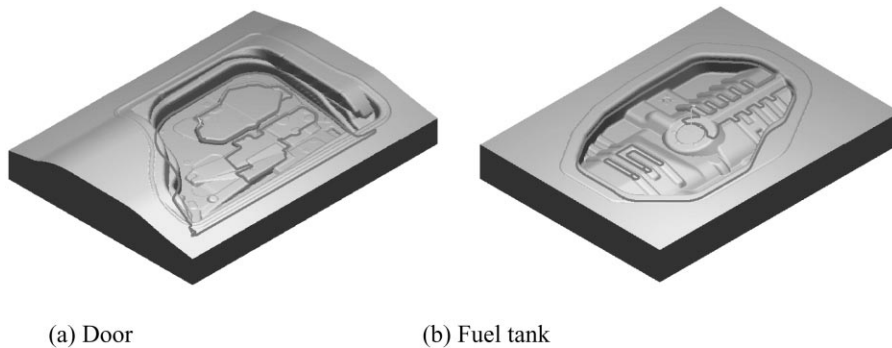
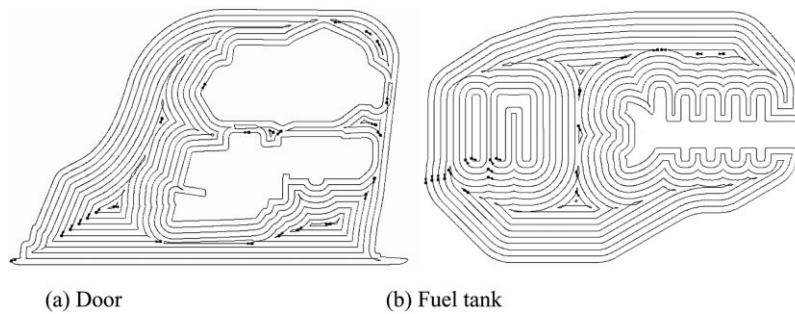


Fig. 13. Areas from CL-surfaces.

Fig. 14. Tool paths without uncut regions ($\omega = 15$, $\rho = 10$).

PWID offsetting algorithm, we refer to the authors' paper [1].

Fig. 15 shows the lengths of the tool paths plotted against the tool-path intervals when the tool radius is 10 mm for the areas in Fig. 13. Note that the lengths of the tool paths decrease with an increase in the tool-path interval, which means an improvement in productivity. In Fig. 15, we observe that the length of the tool paths when the tool-path interval is near 20 mm is about 50% of that when the tool-path interval is near 10 mm.

5. Discussions and conclusions

For die-cavity pocketing, we present a procedure which guarantees the generation of contour-parallel tool paths with no uncut regions. There are two main issues for contour-parallel tool-paths generation: (1) a robust 2D-curve offsetting algorithm; and (2) detecting and removing uncut regions. The 2D-curve offsetting solution has been widely studied, because it has many potential applications. However, though the importance of the uncut problem in

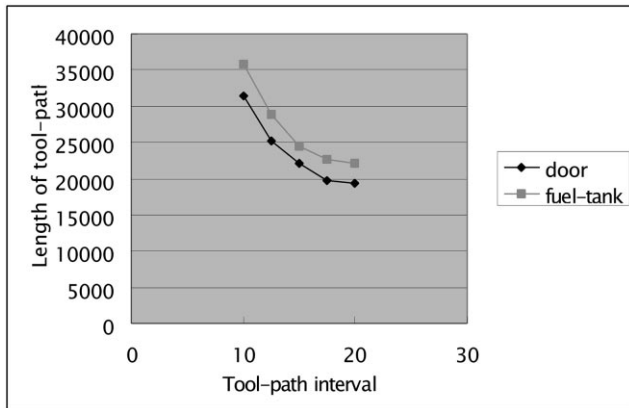


Fig. 15. Plotting of the lengths of the tool paths ($\rho = 10$).

contour-parallel machining has long been recognized there have been few reported investigations on detecting and removing uncut regions.

For the 2D-curve offsetting algorithm, we employ the *PWID offset algorithm* because of its robustness and efficiency. The original PWID offset algorithm was developed for offsetting a closed PS-curve having no “islands” (island PS-curves have to be manually bridged to the boundary PS-curve to make them a single closed PS-curve). For contour-parallel tool-paths generation, we expand the PWID offset algorithm to offset areas having islands.

Our solution based on the expanded PWID offset algorithm, detects and removes the uncut regions. To improve the efficiency of detecting uncut regions, we use the concept of the interfering ranges, LIR and GIR, of the *expanded PWID offset algorithm*. Empirical tests show the usefulness of the proposed algorithm for the improvement of machining productivity.

Applications of contour-parallel machining include area milling of the finishing stage as well as pocket machining of the roughing stage [5]. Although area milling of the finishing stage does not have the uncut problem, the expanded PWID offset algorithm may be useful to generate the contour-parallel tool paths for area milling.

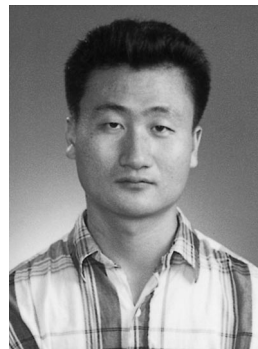
Acknowledgements

The research was supported by the Ministry of Science and Technology of Korea, through its National Research Lab Program.

References

- [1] Choi BK, Park SC. A pair-wise offset algorithm for 2D point-sequence curve. *Computer Aided Design* 1999;31(12):735–45.

- [2] Park SC, Shin H, Choi BK. A sweep line algorithm for polygonal chain intersection and its applications. *Proceedings of IFIP WG5.2 GEO-6 Conference in Tokyo University*, December 7–9, 1998, p. 187–95 (also available from bkchoi@bezier.kaist.ac.kr).
- [3] Choi BK, Kim BH. Die-cavity pocketing via cutting simulation. *Computer Aided Design* 1997;29(12):837–46.
- [4] Choi BK, Kim DH, Jerard RB. C-space approach to tool-path generation for die and mould machining. *Computer Aided Design* 1997;29(9):657–69.
- [5] Choi BK, Jerard RB. *Sculptured surface machining*. Dordrecht: Kluwer, 1998.
- [6] Hansen A, Arbab F. An algorithm for generating NC tool path for arbitrary shaped pockets with islands. *ACM Transactions on Graphics* 1992;11(2):152–82.
- [7] Tiller W, Hansen EG. Offset of two-dimensional profiles. *IEEE Computer Graphics and Applications* 1984;36–46.
- [8] Held M, Lukacs G, Andor L. Pocket machining based on contour-parallel tool paths generation by means of proximity maps. *Computer Aided Design* 1994;26(3):189–203.
- [9] Persson H. NC machining of arbitrary shaped pockets. *Computer Aided Design* 1978;10(3):169–74.
- [10] Kim DS. Polygon offsetting using a Voronoi diagram and two stacks. *Computer Aided Design* 1998;30(14):1069–76.
- [11] Kokichi S. Degeneracy and Instability in Geometric Computation. *Proceedings of IFIP WG5.2 GEO-6 Conference in Tokyo University*, December 7–9, 1998, p. 5–15.
- [12] Chiang CS, Hoffman CM, Lynch RE. How to compute offsets without self-intersection. *SPIE Conference Proceedings on Curves and Surfaces in Computer Vision and Graphics II*, Boston, MA, 1991, p. 76–87.
- [13] Remfeld RF. IGB-offset for plane curves-loop removal by scanning of interval sequences. *Computer Aided Geometric Design* 1998;15(3):339–75.



Sang C. Park is a PhD student in the Department of Industrial Engineering at KAIST. He received his BS and MS degrees from KAIST in 1994 and 1996, respectively. His research interests include geometric algorithms in CAD/CAM, sculptured surface NC machining, simulation and virtual manufacturing system.



Byoung K. Choi is a professor of manufacturing systems engineering in the Department of Industrial Engineering at KAIST, which he joined in 1983. He received a BS from Seoul National University, an MS from KAIST and a PhD from Purdue University, all in Industrial Engineering. His research interests are in the area of sculptured-surface modeling and machining, CAPP for die-cavity machining, discrete event system modeling and simulation, and virtual manufacturing system.