ELSEVIER

Computer-Aided Design 34 (2002) 299–308

COMPUTER-AIDED DESIGN

www.elsevier.com/locate/cad

Offset tool-path linking for pocket machining

S.C. Park*, Y.C. Chung

Cubic Technology Research Center, AceTechno-Tower 1101, 684-1, DungChon-dong, KangSeo-Ku, Seoul, South Korea

Received 25 October 2000; revised 30 January 2001; accepted 9 February 2001

Abstract

For die-cavity pocketing, contour-parallel offset (CPO) machining is the most popular machining strategy. CPO tool-path generation for pocketing includes geometrical and technological issues: (1) a 2D-curve offsetting algorithm; and (2) optimizing technological objectives, such as tool-path linking. The 2D-curve offsetting solution has been widely studied, because it has so many potential applications. However, though the tool-path linking may seriously affect the machining performance, there have been few reported investigations on optimizing the CPO tool-path linking. This paper presents a CPO tool-path linking procedure optimizing technological objectives, such as dealing with islands (positive and negative) and minimizing tool retractions, drilling holes and slotting. Main features of the proposed algorithm are as follows: (1) a data structure, called a 'TPE-net', is devised to provide information on the parent/child relationships among the tool-path-elements; (2) the number of tool retractions is minimized by a 'tool-path-element linking algorithm' finding a tour through the TPE-net; and (3) the number of drilling holes is minimized by making use of the concept of the 'free space' (negative islands or already machined region).

Keywords: Pocket; Tool-path linking; Offset; Optimization

1. Introduction

Pocket machining (or pocketing) is one of the widely used machining operations, particularly for the production of die-cavity [3,6]. Known approaches to pocketing can be distinguished by the tool-path strategy, such as contour-parallel offset (CPO) type and direction-parallel (DP) type [7]. As illustrated in Fig. 1, CPO type uses successive offsets of the pocket contour as the tool-path-elements (cutting paths of the tool). In this article, the term 'pocketing' is used to refer to the pocketing combined with the CPO type tool-path.

Offsetting 2D-curve [4,6–8,10] has been regarded as the key issue in generating the pocketing tool-path, because it plays an important role in the generation of the correct CPO tool-path-elements. However, generating NC tool-path for pocketing requires more than a 2D-curve offsetting solution. To optimize NC tool-path, it is necessary to consider not only geometrical problems, but also technological requirements, such as uncut region removal [5,9] and tool-path linking [1,7]. The tool-path linking, in particular, may seriously affect the machining time and quality. However, there have been few reported investigations on the CPO type tool-path linking problem. This serves as a motivation for exploring the possibility of finding an appropriate tool-path linking algorithm meeting most practical requirements on the shop floor.

Guyder [1] addresses some guidelines for CPO tool-path optimization and tool-path linking algorithm based on the guidelines. The guidelines—compiled over time in consultation with machining operators—are very practical. However, they are not properly accommodated by the presented tool-path linking algorithm oversimplifying the linking problem as a tool-path-elements ordering problem. Held [7] presents an algorithm for CPO tool-path generation for pocket machining based on proximity maps i.e. Voronoi diagram. To machine a pocket containing islands by consistent CPO tool-path, he discusses a linking procedure requiring a spanning tree of the planar graph of the monotonic pouches.

The objective of the paper is to develop a CPO tool-path linking procedure achieving technological objectives, such as dealing with islands (positive and negative) and minimizing tool retractions and slotting. The CPO tool-path linking problem can be informally specified as follows.

* Corresponding author. Tel.: +82-2-3663-2676; fax: +82-2-3664-4701.
E-mail address: psc@cubictek.com (S.C. Park).

0010-4485/02/$ - see front matter © 2002 Elsevier Science Ltd. All rights reserved.
PII: S0010-4485(01)00088-4

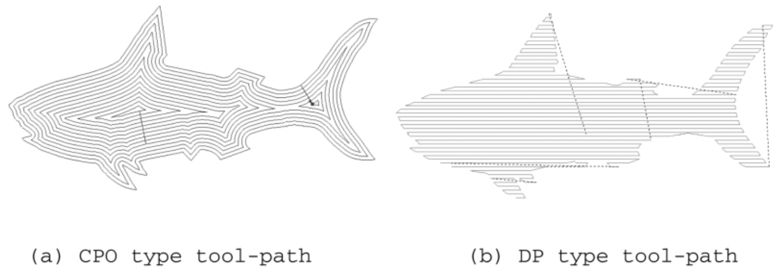


Fig. 1. CPO & DP type tool-paths.

1.1. CPO tool-path linking for pocketing

- Input: tool-path-elements (offset curves), technological objectives and constraints.
- Output: CPO tool-path for machining the pocket according to the technological objectives.

The generally accepted '**functional requirements**' (user requirements) of the pocketing (or general machining) may be summarized as follows: (1) no gouge; (2) safe machining without tool breakage or collision; (3) efficient machining to save the machining time; and (4) fine surface quality without tool-marks. To achieve the functional requirements, we employ the guidelines of Guyder [1] for the CPO tool-path linking. Fig. 2 shows the correspondence between the functional requirements and the guidelines.

1.1.1. Guyder's [1] guidelines

1. Gouge free milling is required;
2. Prevent milling both sides of a path before milling the path itself;
3. Prevent over-milling, i.e. cutting part or all of a path more than once;
4. Mill paths that are innermost first unless previous guidelines prohibit (minimize slotting);
5. Paths closest to a boundary of a part (finishing paths) should be milled with additional care, such as tangential approach and cutter parameter;
6. Tool intersects the path to-be-milled only at its starting point. Tool-paths can be reorganized to have a different starting point, if by so doing, a retraction is avoided;
7. Tool intersects no unmilled nest of tool-paths;
8. Tool may cross milled tool-paths.

In this paper, we develop a CPO tool-path linking algorithm according to the guidelines. The overall structure of the paper is as follows. The next section gives the preliminary definitions followed by the explanation of the proposed CPO tool-path linking procedure in Section 3. Illustrative examples are given in Section 4. Finally concluding remarks are given in Section 5.

2. Preliminary definitions

As shown in Fig. 3, the CPO tool-path generation procedure can be split into three steps: (1) tool-path-elements generation (2D curve offset); (2) tool-path-element linking for sub-paths; and (3) sub-path linking for a tool-path. The tool-path-elements generation (Step 1) is a pure geometric problem, whereas the next steps (Steps 2 and 3) include many technological issues. As mentioned in the previous section, we assume that the correct tool-path-elements are available and focus on the tool-path linking problem, i.e. Steps 2 and 3.

Now we introduce some terms in order to provide a

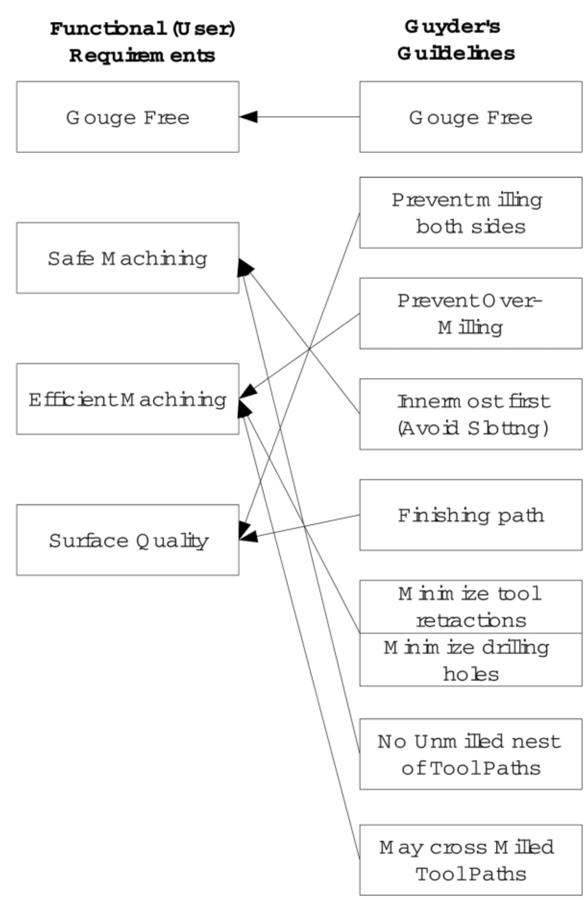


Fig. 2. Functional requirements and Guyder's guidelines.

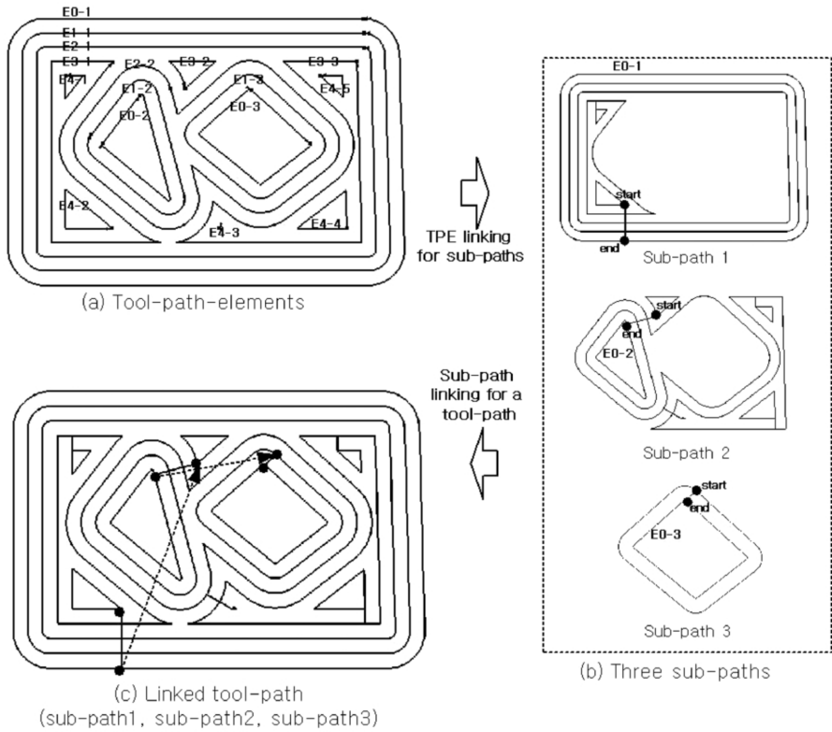


Fig. 3. CPO tool-path generation procedure.

systematic description of the proposed CPO tool-path linking procedure.

2.1. Definition 1 (offset number and initial boundaries)

Each tool-path-element has its **offset number** indicating how many times of offset was performed for the tool-path-element. Tool-path-elements having zero offset number are called **initial boundaries**. Initial boundaries also can be defined as curves enclosing the machining area on the CL-surface (cutter-location surface) [2,3].

Fig. 3(a) shows the example of tool-path-elements generated by repeated offsetting of the area bounded by three initial boundaries (E0-1, E0-2, E0-3). For instance, the **offset number** of the tool-path-element E3-1 is 3, because E3-1 requires three times of offsetting the initial boundaries with the tool-path interval δ (offset distance).

2.2. Definition 2 (parent/child relationship)

Let E_i and E_{i+1} be the two tool-path-elements having consecutive offset numbers i and $i+1$, respectively. Then, E_i (E_{i+1}) becomes the **parent** (**child**) of E_{i+1} (E_i) if the distance between the two tool-path-elements is the same with the tool-path interval δ.

In Fig. 3(a), the tool-path-element E0-1 is the **parent** of E1-1 (E1-1 is the **child** of E0-1, respectively) because they have consecutive offset numbers (0 and 1) and the distance is the same with the tool-path interval δ. A tool-path-element may have multiple parents and children. For instance, the tool-path-element E2-2 has two parents (E1-2 and E1-3) and the tool-path-element E2-1 has three children (E3-1, E3-2 and E3-3).

We believe it is reasonable to build up a data structure providing information on the **parent/child** relationships among the tool-path-elements generated by repeated offsetting. This goal has been realized by constructing a ***tool-path-elements net*** (**TPE-net**) consisting of tool-path-elements (offset curves) and their parent/child relationships. As a useful side-effect, performing this preprocessing step separates the computation of the tool-path-elements from the actual construction of the tool-path. The availability of this **TPE-net** will give us the possibility to perform some optimizations of the tool-path, thereby working entirely on the **TPE-net**. Fig. 4 shows the example of the TPE-net of the tool-path-elements shown in Fig. 3(a). In the TPE-net, shaded rectangles represent the parent/child relationships between tool-path-elements.

3. CPO tool-path linking

As discussed earlier, we employ the Guyder's guidelines to develop a CPO tool-path linking procedure meeting the requirements on the shop floor. The proposed tool-path linking procedure consists of three modules: (1) TPE-net construction; (2) tool-path-element linking for sub-paths; and (3) sub-path linking for a tool-path. Each module is designed to accommodate the technological requirements suggested by Guyder [1]. Fig. 5 shows the correspondence between the three modules and the guidelines.

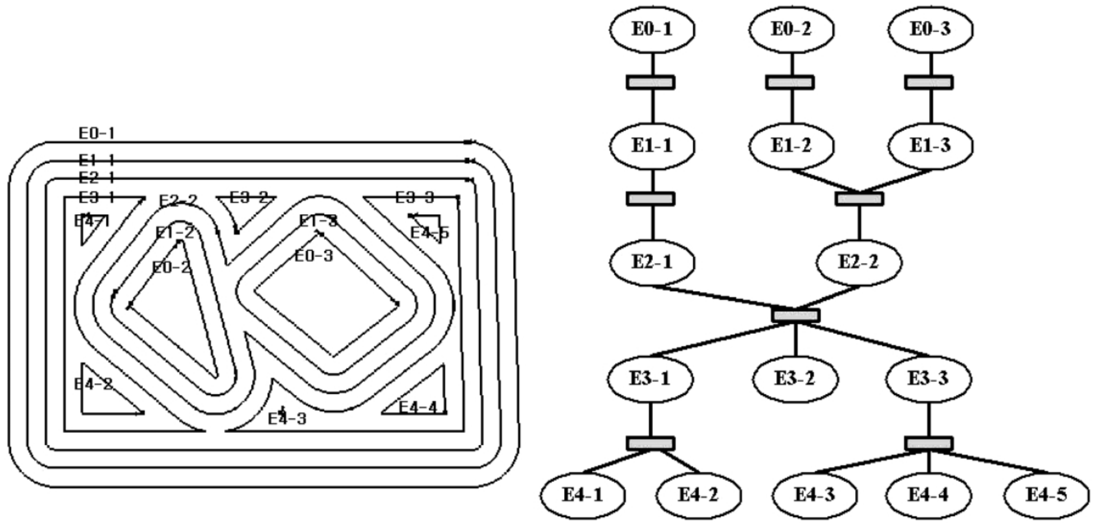


Fig. 4. Tool-path-elements and TPE-net.

3.1. TPE-net construction

According to the definition of the parent/child relationship, we can intuitively construct the TPE-net with the tool-path-elements, as shown in Fig. 4. However, it should be noted that the TPE-net may be constructed more efficiently by making use of the information available during the offsetting operations, such as **point correspondence** between offset curves (tool-path-elements). As shown in Fig. 6, a pair of **corresponding points** is defined as two points satisfying following conditions: (1) the two points should be on offset curves having successive offset numbers; and (2) the distance between the two points should be the same with the offset distance (tool-path interval). Therefore, we can say that two tool-path-elements having the point correspondence have the parent/child relationship. Fortunately the information on the point correspondence is usually available in offsetting algorithms (Voronoi [7], Pair-wise [4]), the parent/child relationships among the tool-path-elements can be obtained without additional computation.

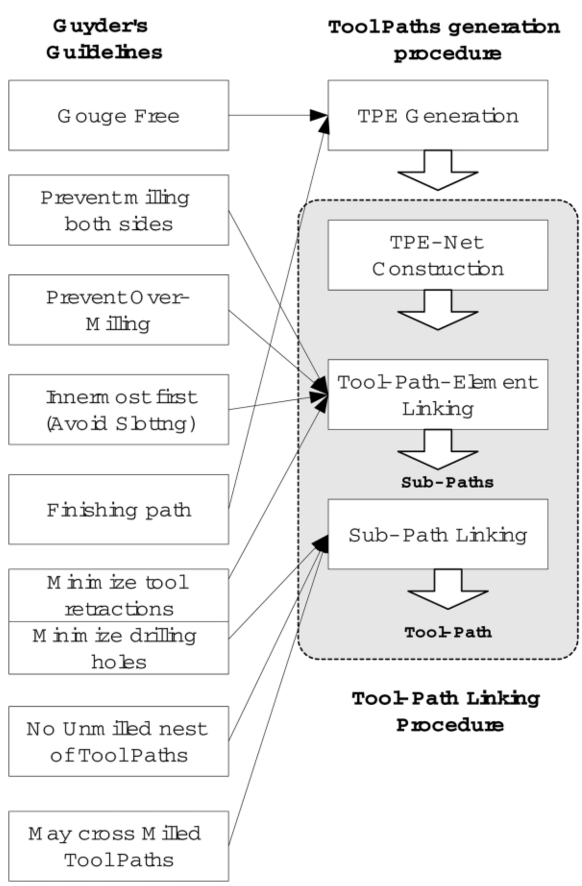


Fig. 5. Three modules and Guyder's guidelines.

After the construction of the TPE-net, the tool-path linking procedure is performed entirely on the TPE-net. The CPO tool-path linking based on the TPE-net corresponds to finding a tour though the TPE-net so that every node (tool-path-element) has been traversed.

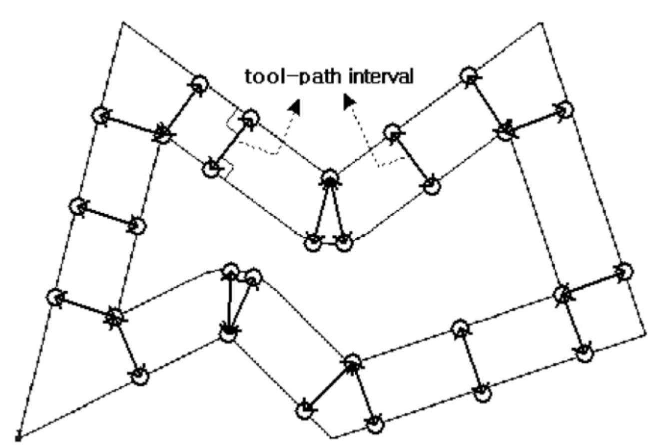


Fig. 6. Point correspondence between offset curves.

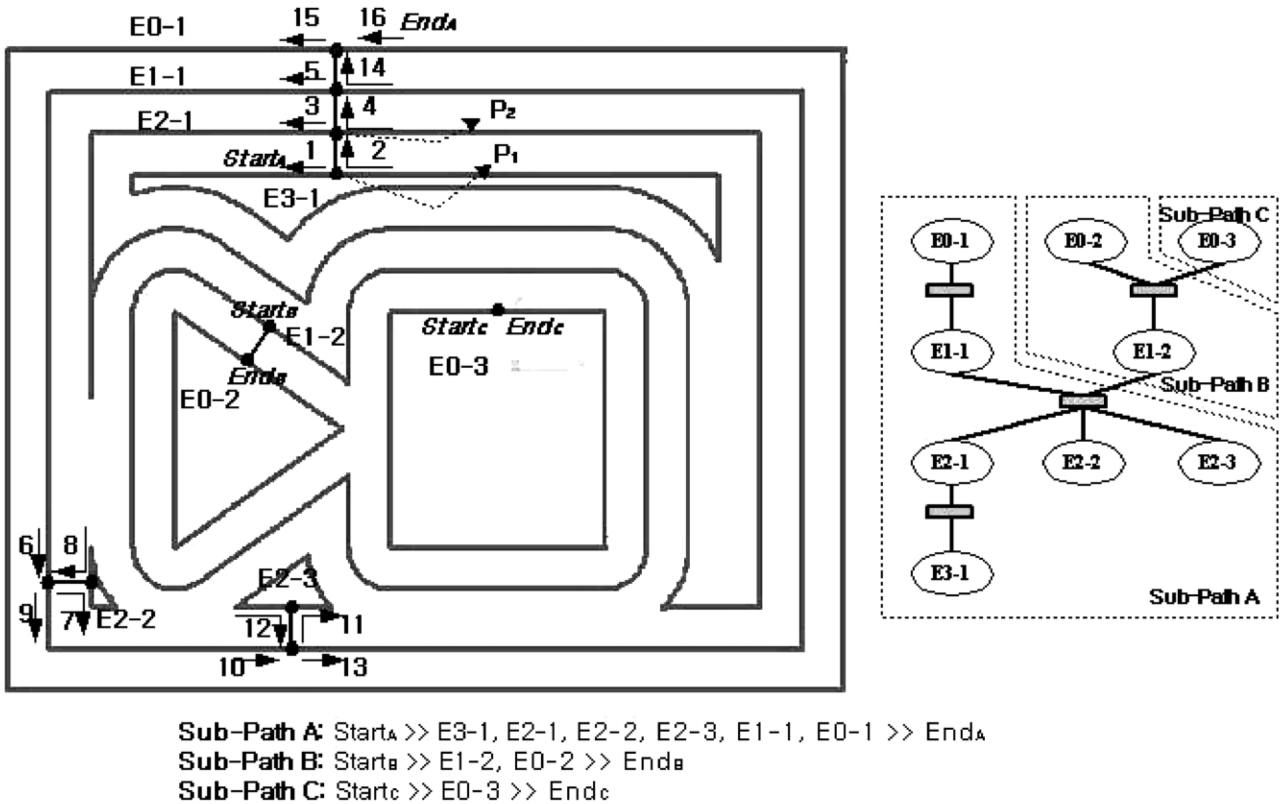


Fig. 7. An example of tool-path-element linking for sub-paths.

3.2. Tool-path-element linking for sub-paths

We define a sub-path as the group of tool-path-elements, which can be machined without tool retractions. As shown in Fig. 7, 'sub-path A' consists of six tool-path-elements, ⟨E3-1, E2-1, E2-2, E2-3, E1-1, E0-1⟩, linked by straight lines connecting a pair of corresponding points among the tool-path-elements. We can see that the 'sub-path A' can be machined without tool retractions by following the arrow lines (1–16) along the tool-path-elements.

Remember that there are four guidelines for the tool-path-element linking module to achieve the functional requirements (see Fig. 5). Our approach to accommodate the guidelines is specified as follows.

3.2.1. Four guidelines for the sub-path linking module

1. *Minimize the number of tool retractions*: If we have more than two sub-paths, then it is necessary to link sub-paths by inserting additional paths (tool retractions) which do not contribute to the actual cutting. Therefore, we can improve the productivity of the tool-path by reducing the number of tool retractions. For the minimum number of tool retractions, we have to minimize the number of sub-paths and the method will be explained later.
2. *Prevent over milling*: During the sub-path linking, we flag tool-path-elements that have already been traversed as visited to prevent machining a tool-path-element twice.
3. *Innermost path first (minimize slotting)*: Because slotting causes drastic increases of chip-load [3] (the rate of chip-generation during machining) and chip evacuation problems, it is desirable to minimize slotting. However, in general, we have to slot at least a tool-path-element in a sub-path. To minimize the length of slotting path, we select the innermost tool-path-element first having the smallest path length (see 'sub-path A' in Fig. 7, it starts from the inner most tool-path-element E3-1.).
4. *Prevent milling both sides of a path before milling the path itself*: Milling both sides of a path before milling the path itself may cause some problems (bad surface quality) in the real machining. To avoid this case, we link the tool-path-elements sequentially, outward (innermost path first). Going outward also allows consistent up (conventional)/down (climb) milling [3].

Among these guidelines, the only non-trivial issue is the minimizing the number of tool retractions, i.e. the number of sub-paths. One clear fact is that the minimum number of sub-paths is the same as the number of initial boundaries having the zero offset number, because it is not possible to traverse initial boundaries without tool retractions as long as the innermost path have be machined first. We present a tool-path-element linking algorithm for minimal number of sub-paths first, and then additional explanations are provided later. The proposed tool-path-element linking algorithm is as follows.

3.2.2. Tool-path-element linking algorithm// Input: TPE-net including n initial boundaries.// Output: n sub-paths (linked paths).

Step 1. TPE_c = Get the innermost tool-path-element (having the biggest offset number), among unvisited tool-path-elements in the TPE-net;
Step 2. If TPE_c is NULL, then stop;
Step 3. TPE_p = Get a tool-path-element among the parents of TPE_c;
Step 4. Find a pair of corresponding points between TPE_c and TPE_p, then link them by connecting the corresponding points.
Step 5. Flag TPE_c as visited; Go to Step 1.

The algorithm will be explained with the example shown in Fig. 7. At first, the tool-path-element E3-1 is selected as the innermost tool-path-element in Step 1. In Step 3, TPE_p becomes the tool-path-element E2-1 because E2-1 is the parent of the E3-1 (TPE_c). Then, we can arbitrary select a pair of corresponding points ($\boldsymbol{P_1}$ and $\boldsymbol{P_2}$) between the two tool-path-elements, TPE_c (E3-1) and TPE_p (E2-1), and link them by connecting the corresponding points. In this way, every tool-path-element is inserted into one of its parents. As a result, we obtain three sub-paths (sub-path A, B and C) as many as the number of the initial boundaries (E0-1, E0-2 and E0-3).

3.3. Sub-path linking for a tool-path

What remains is to make a tool-path by linking the sub-paths with tool retractions. As depicted in Fig. 6, there are three guidelines for the sub-path linking module. Our approach to accommodate the guidelines is specified as follows.

3.3.1. Three guidelines for the tool-path linking module

1. *Minimize the number of drilling holes*: To link multiple sub-paths, it is necessary to lift the tool, move it in the air, and plunge it down to continue the next sub-path. However, in general, it is not possible to plunge the tool down onto the material without having a hole drilled previously. To avoid the additional production step (drilling), we employ the concept of **free space**, 2D regions having a surface lying below the actual machining level. (Note that machining a sub-path widens the free space.) We can make use of the free space to plunge the tool down near the starting point of the next sub-path as near as possible. Then, the tool moves to the starting point of the next sub-path by slotting. Although slotting is not desirable, the process planning engineer may prefer slotting to an extra drilling operation. The detailed explanation of a method, utilizing the free space to minimize the number of drilling holes, will be provided later.
2. *Tool intersects no unmilled nest of tool-path*: Because milling an unmilled nest of tool-path means slotting, it is not desirable. However, a small amount of slotting is inevitable to avoid drilling operations. We minimize the length of slotting by making use of the free space information.
3. *Tool may cross a milled tool-path*: As discussed earlier, a milled tool-path widens the free space and we utilize the information to minimize drilling operations.

For the formal description of the **free space**, we introduce two kinds of regions: (1) **Initial free space** is the union of negative islands (among the initial boundaries); and (2) **Machined free space** is the union of the tool-envelopes of the machined tool-path-elements. Intuitively the free space is defined by the sum of the initial free space and the machined free space. Note that we may remove the drilling operation for the first tool plunge by making use of the initial free space (negative islands) information, and other drilling operations for the tool plunges between sub-paths can be removed by making use of the machined free space. We present a sub-path linking algorithm minimizing the number of drilling operations first, and then additional explanations are provided later. The proposed sub-path linking algorithm is as follows.

*3.3.2. Sub-path linking algorithm// Input: n sub-paths (linked paths), TPE-net.// Output: a linked tool-path (***TOOL-PATH***).*

Step 0. Preprocessing: If there is a sub-path including a negative island (i.e. accessible from a negative island), reverse the sub-path to make the starting point accessible from the negative island; // use the initial free space
Step 1. S_m = Arbitrarily select a sub-path among unlinked sub-paths. (If there is an unlinked sub-path including a negative island, select the sub-path first.);
Step 2. Flag S_m as linked (machined) and append S_m to the **TOOL-PATH**;
Step 3. S_n = select an unlinked sub-path having the nearest starting point to the end point of S_m;
Step 4. If S_n is NULL then STOP;
Step 5. P_t = find the nearest point to the starting point of S_n in the free space (on the sub-paths already machined); // P_t: tool plunging point;
Step 6. Append following movements to the **TOOL-PATH**: (1) lift the tool; (2) move it in the air; (3) plunge the tool down on the point P_t; and (4) move the tool to the starting point of S_n by slotting;
Step 7. Assign S_n to S_m, and go to Step 2.

The sub-path linking algorithm will be explained with the sub-paths of Fig. 7 which is shown again in Figs. 8 and 9. Fig. 8 assumes that the two islands (E0-2 and E0-3) are

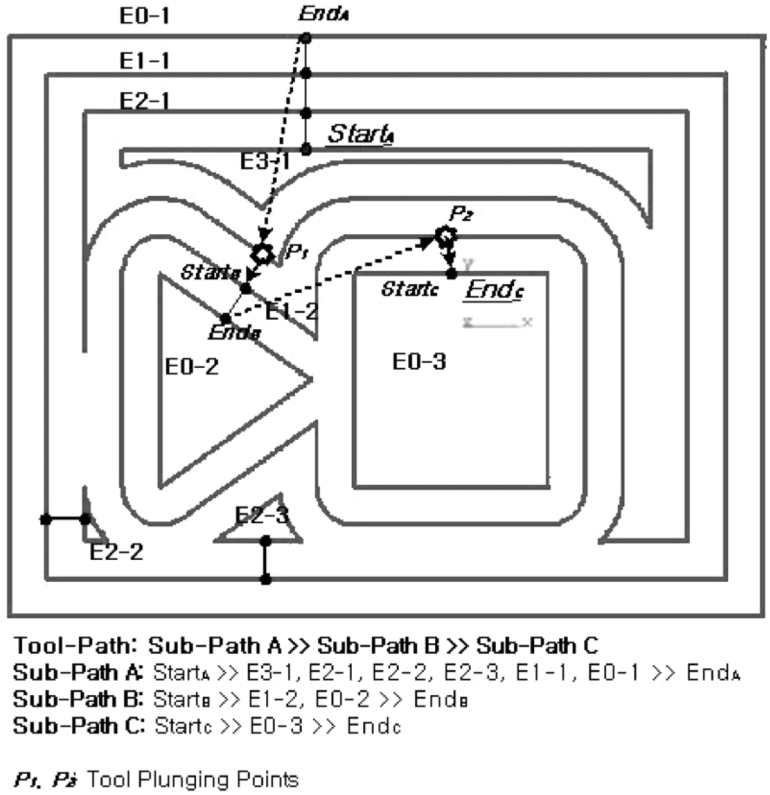


Fig. 8. An example of sub-path linking (no negative island).

positive islands, whereas Fig. 9 deals with negative islands (assume E0-2 is a negative island).

At first, we explain the sub-path linking example in Fig. 8. The 'sub-path A' is selected as the S_m in Step 1. Then, the 'sub-path A' (S_m) is flagged as linked (machined) and appended to the tool-path in Step 2. In Step 3, we select the 'sub-path B' as the S_n, because the 'sub-path B' has the nearest starting point ($Start_B$) to the end point (End_A) of the 'sub-path A' (S_m). To move the tool from the machined 'sub-path A' (S_m) to the next 'sub-path B' (S_n), we have to find the tool plunging point near the starting point of 'sub-path B' (S_n). As discussed earlier, the tool plunging point should be found in the free space (on the sub-paths already machined) to avoid drilling operations. In Step 5, the point P_1 is selected as the tool plunging point P_t for the 'sub-path B'. It remains to append the tool movements from the end point of the 'sub-path A' to the starting point of the 'sub-path B' in Step 6. In the same way, we can link the 'sub-path C' as shown in Fig. 8.

Sub-path linking algorithm dealing with negative islands will be explained with the help of Fig. 9 (assume the island E0-2 is a negative island). Initially, we have to preprocess the 'sub-path B' including the negative island (E0-2) in Step 0. Because the tool can plunge in a negative island without a drilled hole, the 'sub-path B' is reversed to make the sub-path accessible from the negative island, i.e. $Start_B$ changes positions with End_B (compare Figs. 8 and 9). Then the 'sub-path B' is selected as the S_m in Step 1, and Step 2 flags the 'sub-path B' as linked (machined) and appends to the tool-path. In Step 3, we select the 'sub-path C' as the S_n, because the 'sub-path C' has the nearest starting point ($Start_C$) to the end point (End_B) of the 'sub-path B' (S_m). Then, the point P_2 is selected as the tool plunging point P_t for the 'sub-path C' in Step 5. The tool movements from the machined 'sub-path B' (S_m) to the next 'sub-path C' (S_n) are appended to the tool-path in Step 6. As shown in Fig. 9, the tool-path can be linked in this way.

4. Illustrative examples

The proposed CPO tool-path linking procedures have been implemented and tested with various examples. Fig. 10 shows two examples and the area curves were obtained by intersecting the CL-surfaces with a horizontal plane.

Fig. 11 shows the linked tool-paths machining the areas of Fig. 10, with the tool-path interval 23 mm [Fig. 11(a)] and 15 mm [Fig. 11(b)], respectively. We can observe that the number of sub-paths of each tool-path is the same with the number of its initial boundaries.

Remember that, the suggested tool-path-element linking algorithm generates a single sub-path when the machining area does not have any island, i.e. the number of initial boundary is one. That means we can get a tool-path having no tool retraction if the machining area can be represented

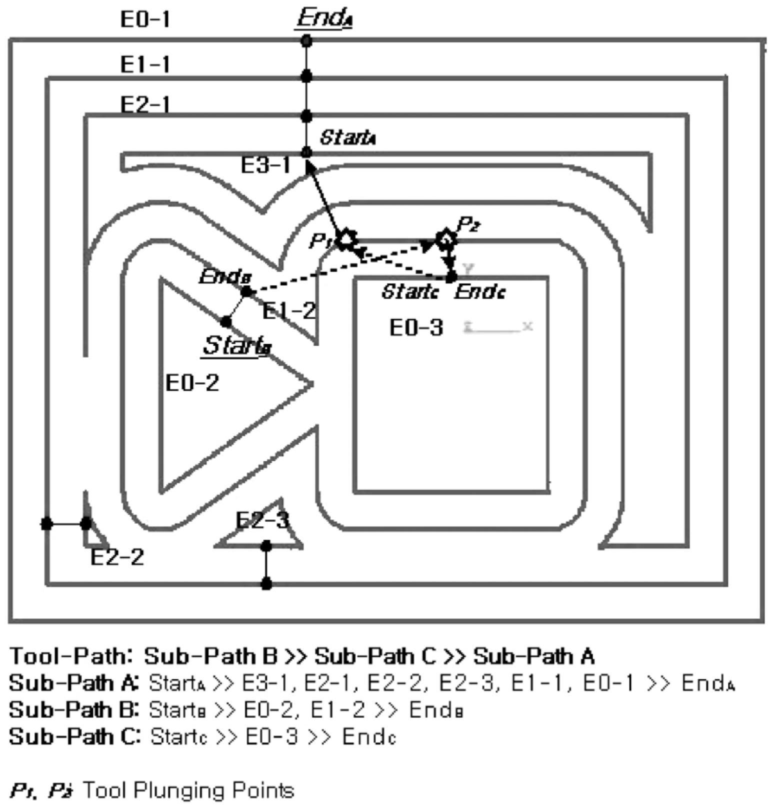


Fig. 9. An example of sub-path linking (E0-2 is a negative island).

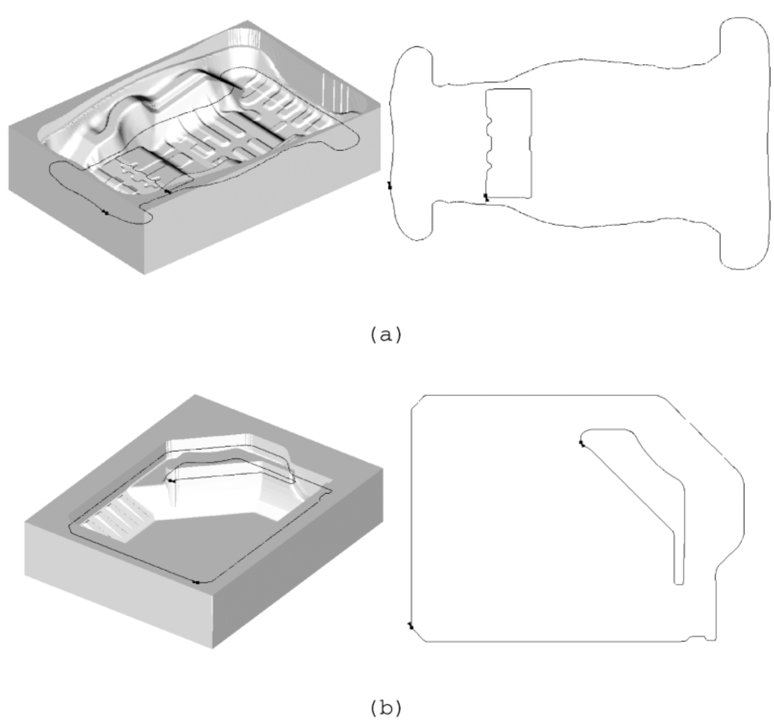
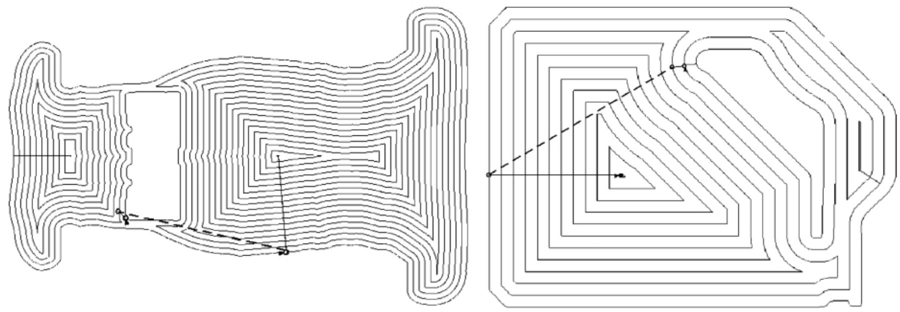


Fig. 10. Initial boundaries from CL-surfaces.



(a) (b)

Fig. 11. Examples of tool-path linking.

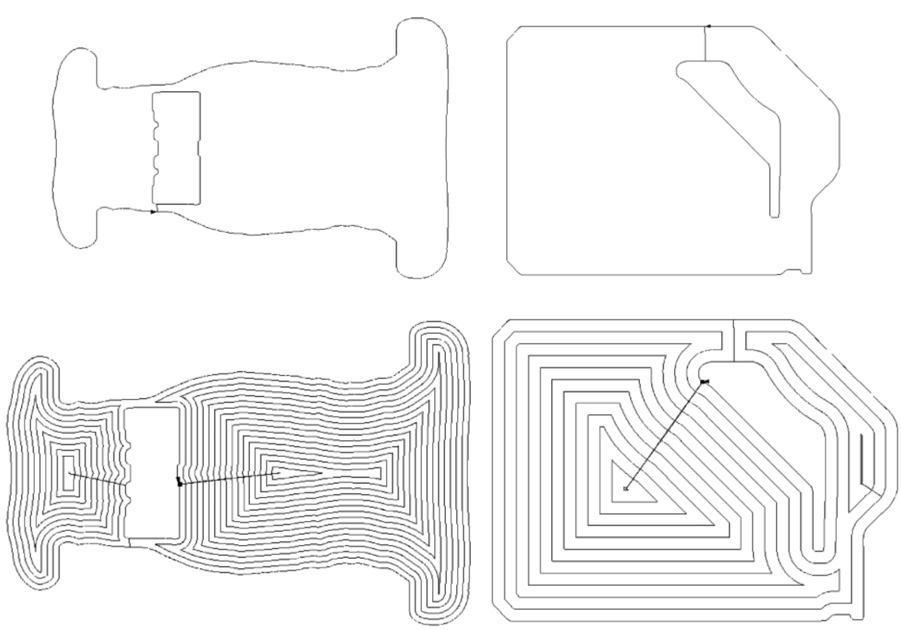
as a single closed curve. Fig. 12 shows the tool-paths without tool retractions, the machining area is represented as a single closed curve by bridging the islands to the outer boundary.

5. Discussions and conclusions

For pocket machining, we present a CPO tool-path linking algorithm to achieve technological objectives, such as minimizing tool retractions, drilling holes and slotting and handling islands (positive and negative). The suggested tool-path linking procedure consists of three modules: (1) TPE-net construction; (2) tool-path-element linking for sub-paths; and (3) sub-path linking for a tool-path. For the technological objectives, each module is designed by following the Guyder's [1] guidelines.

The TPE-net is devised to provide information on the **parent/child** relationships among the tool-path-elements generated by repeated offsetting. By employing the TPE-net, we can separate the computation of the tool-path-elements from the actual construction of the tool-path and it is possible to perform some optimizations of the tool-path entirely on the TPE-net.

In order to improve the productivity of the tool-path, it is necessary to minimize the number of tool retractions, because a tool retraction requires additional paths which do not contribute to the actual cutting. To minimize the



(a) (b)

Fig. 12. Removing tool retractions by bridging.

number of tool retractions, we suggest a tool-path-element linking algorithm minimizing the number of sub-paths (as many as the number of initial boundaries).

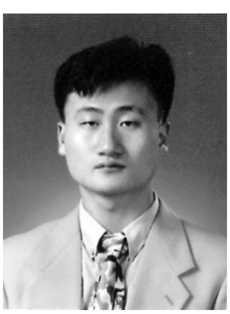
One of the important issues in the sub-path linking is the minimizing the number of drilling holes. To avoid the additional production step (drilling), we employ the concept of **free space (= Initial free space + Machined free space)**, 2D regions having a surface lying below the actual machining level. A drilling hole for the first tool plunge can be removed by making use of the initial free space and other drilling holes for tool plunges between sub-paths can be removed by making use of the machined free space.

Acknowledgements

This research was supported by the Ministry of Science and Technology of Korea, through a 'National Research Lab Project' grant to Cubic Technology Research Center.

References

[1] Guyder MK. Automating the optimization of 2 1/2 axis milling. Computers in Industry 1990;15:163–8.
[2] 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.
[3] Choi BK, Jerard RB. Sculptured surface machining. Kluwer, 1998.
[4] Choi BK, Park SC. A pair-wise offset algorithm for 2D point-sequence curve. Computer Aided Design 1999;31(12):735–45.
[5] Park SC, Choi BK. Uncut-free pocketing tool-paths generation using pair-wise offset algorithm. Computer-Aided Design 2001 (in press).
[6] Choi BK, Kim BH. Die-cavity pocketing via cutting simulation. Computer-Aided Design 1997;29(12):837–46.
[7] 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.
[8] 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.
[9] Manuel D, Liang M, Kolahan F. A dynamic offsetting approach to tool path generation for machining convex pockets. Computers Ind Engng 1996;31(1/2):135–8.
[10] Persson H. NC machining of arbitrary shaped pockets. Computer-Aided Design 1978;10(3):169–74.



Sang C. Park is a senior researcher of Cubictek Co., a CAD/CAM software company in Korea. He received his BS, MS and PhD degrees from KAIST in 1994, 1996 and 2000 respectively, all in Industrial Engineering. His research interests include geometric algorithms in CAD/CAM, sculptured surface machining and system modeling and simulation. He can be reached via email at psc@cubictek.com.

Yun C. Chung is a chief engineer of Cubictek Co., a CAD/CAM software company in Korea. He received his BS degree from Hanynag University, and his MS and PhD from KAIST, all in Industrial Engineering. He was a research engineer at Tech. Center of DaimlerChrysler. His research interests are surface and solid modeling, tool-path generation and verification, CAD/CAM system, and computer graphics. He can be reached via email at cyc@cubictek.com.