

# An Algorithm to Generate Cutting Path for Micro-Fabrication\*

Jin Jin Zheng<sup>1</sup>, X P Wang<sup>1</sup>, H J Zhou<sup>2</sup>, W H Huang<sup>1</sup> and L G Shen<sup>1</sup>

<sup>1</sup>Precise Machinery and Precise Instrument Department,  
University of Science and Technology of China,  
Hefei, China 230026  
Email: jjzheng@ustc.edu.cn

<sup>2</sup>NSRL  
University of Science and Technology of China,  
Hefei, China 230029

*Abstract.* In a two-photon femtosecond laser system of three-dimensional micro-fabrication, a model of micro-device is fabricated layer by layer along its contours of cross-sections. So it is necessary to generate cutting path. In this paper, we present an efficient algorithm to generate cutting path which is a series of ordered contour points on each cross-section of a model.

**Keyword.** Micro-fabrication, cutting path, contour tracing, algorithm.

## 1 Introduction

Since a femtosecond laser machine was invented in 1980s, the technology of femtosecond laser has been widely applied in physics, chemistry, biology, and optoelectronics. At the beginning of 1990s, the mode-locked titanium sapphire femtosecond laser was invented and then the two-photon femtosecond laser system came into the field of micro-fabrication. Many recent researchers have shown the applications of femtosecond laser technology in three-dimensional micro-fabrication.

In a two-photon femtosecond laser system [1], a 3D micro-device model is fabricated point by point by two-photon excitation with a photochromic material. In order to efficiently fabricate a model, NC data should be prepared. One way to prepare the NC data includes the following steps: the model is first sliced into hundreds or thousands cross-sections by a set of parallel planes; and then each cross-section is output as a black-white bitmap image; coordinates of points on each cross-section are generated from a series of bitmap images. The model is fabricated layer by layer along its cross sections. It is not feasible to fabricate a cluster of disordered points one by one on its cross sections, so the cutting paths should be generated as a series of ordered points along the boundaries.

Contour tracing is a very popular technique [9,10,11]. There are usually two methods used to extract the boundaries: run-length based algorithms [2] and chain code based algorithms [3]. However, the well known and cited algorithm proposed by Pavlidis and Liow [3] may miss inner contours and trace one contour loop more than once. Ren [4] and Chang [5] improved Pavlidis and Liow's algorithm by labeling the inner contour or outer contour with different tokens. Still it also has two disadvantages: first, tracing some special contours more than once, such as single line with one pixel width; second, contours with some intersection loops may not be traced in one tracing process. Zingaretti's algorithm [6] is also based on run-length coding. By using "minimal runs", his algorithm obtains the boundary exactly and is of time efficiency. However, it uses a set of templates ( $3 \times 2$ ) to match pixel patterns, which may complicate the algorithm. To obtain the minimal runs separately may complicate the algorithm. Park and Choi's

---

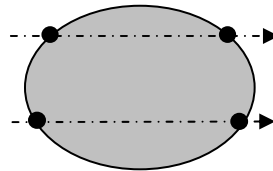
\* This work is supported by "Hundred Talents Project" of CAS and NSF of China (60473133, 10575097).

method [7] presents another simple algorithm. They drop the templates and link the contour pixels directly, but the inner contours are exacted incorrect. Both Zingaretti's and Park-Choi's method may also trace some contours more than once. Chia et al [8] presented a parallel algorithm to generate chain code and implement on hardware. It is hard to implement for software.

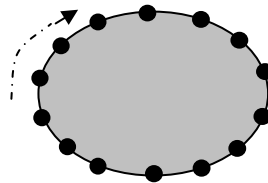
In this paper, an algorithm is presented to generate cutting path which is a sequences of ordered fabrication points based on the information within contour point-pairs. In this way, all the pixels on the contours are traced in sequence just only once. It doesn't distinguish the outer or inner contours and applies a novel and simple method to trace neighbor pixel. With registering the intersection pixel, any connected boundaries would be trace in one tracing process.

There are two steps in our algorithm to generate cutting paths:

- Parallel scan process. All the points on each cross section are scanned along parallel scan lines. Two end points on each scan line are registered as a contour point-pair. A model will be fabricated according to the contour point-pairs—beginning from one point and stopping at the other point (see Figure 1).
- Cutting path generation. Based on the point pairs generated in the first step, an efficient method is generated to obtain cutting path by tracing all contour points on each cross-section as ordered sequences, so that a model can be fabricated point by point (see Figure 2).



**Fig. 1.** Process along parallel scan lines.



**Fig. 2.** Process generates cutting path in terms of ordered contour points.

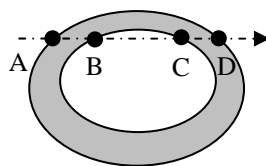
The first step is preparing one. The second step is the main one.

## 2 The generation Process

As mentioned in Section 1, there are two steps to prepare NC data for the three-dimensional micro-fabrication process: (1) parallel scanning—provides the information of the process area in the form of contour point-pairs; (2) contour searching—based on the contour point-pairs obtained in parallel scanning, generates ordered contour points in sequences.

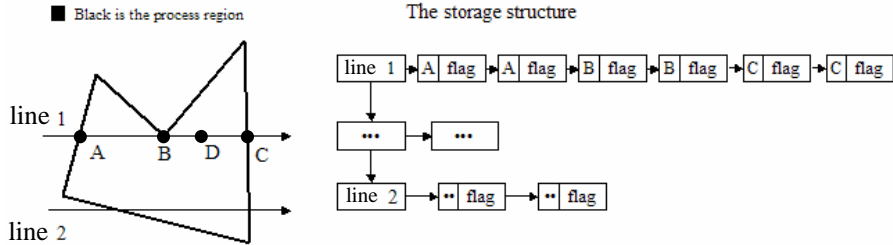
### Parallel Scanning (Preparing step)

Parallel scanning generates contour point-pairs on each cross section of a model. Each cross section is scanned through along a set of parallel lines, and two intersection points of a scan line (one enters the process area and one goes out ) are registered as a point-pair (see Figure 3).



**Fig.3.** A-B and C-D are registered as two contour point-pairs respectively.

If a scan line goes across a single pixel point, then the point will be registered as a pair itself (see Fig. 4).



**Fig. 4.** Scan lines go through a cross-section. A, B and C are registered respectively as contour point-pairs themselves. Corresponding data structure of registered point-pairs.

All the parallel scanning lines cover each pixel of the bitmap images, and we store the contour point-pairs in a two-dimensional array PP, called point-pair array, and are able to recover the process area with the array.

### Cutting path generation(main step)

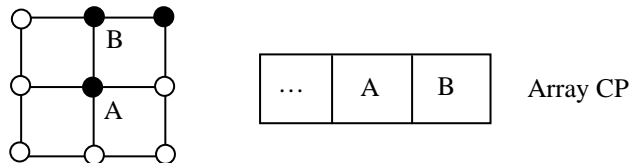
Although after parallel scanning, all contour points are registered, they are not arranged in a sequence. Contour tracing is needed to provide cutting path in terms of ordered contour sequence. To be convenient for process, the cutting path we get must be satisfied with the following requirements:

- The points in the contours are ordered in some consistent sequence;
- The contours obtained are accurate.

After parallel scanning, we have the information of contour point-pairs array PP on the contours of each cross-section. We will now in a position to generate order contour point sequences by point tracing with only contour point-pairs.

### Next Point Tracing

The general rule for tracing next contour point is as following. Starting from a point A in the contour point-pairs array PP, we call it current point and put it into an array called contour point array CP (which stores the contour point sequence). Now we are trying to search for its next contour point by clockwise rule. We first check its eight neighbor points to see which are in between a point pair in array PP, since all points in between a point pair are in process area. The first clockwise point B, which is in



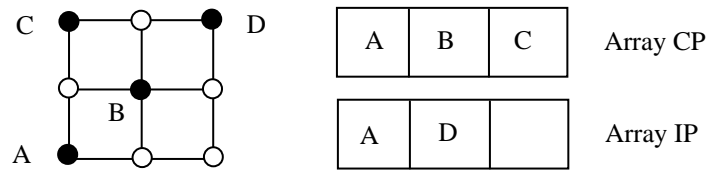
**Fig. 5.** A current point. Next point B registered.

between a point-pair, will be registered as the next contour point and be put into an array CP. In Figure 5, black points are in process area, and the middle point A is current point. The next point in the contours shall be one of the eight points around point A. Point B is registered as its next point and put into array CP. The procedure continues until we meet a visited point again.

However, there are some special cases. For example, when a point is the intersection of two or more contour lines, special rules must be set.

### Intersection Point

A current point is considered as intersection point if there are three or more neighbor points in process area and at least three points are separated by non-process points. In the Figure 6, preceding point is A, and the current point B has three neighbor points in process area and they are all separated by non process points. So B is an intersection point. For intersection point B, its next process point is registered as C, but all its remaining neighbor process points (except its preceding point A) will be recorded in an array IP. If a point in array IP is visited, then it will be flagged. After one tracing, next tracing begins from unvisited point in array IP. Then new tracing begins from unvisited point in contour point-pair array PP. In some complex cases, we generate the contours through several searches, each of which traces a loop of contour. All the loops cover the entire contour. A trace stops at a current point when all neighbor process points are visited. Following this rule, we can get all the loops without overlap.



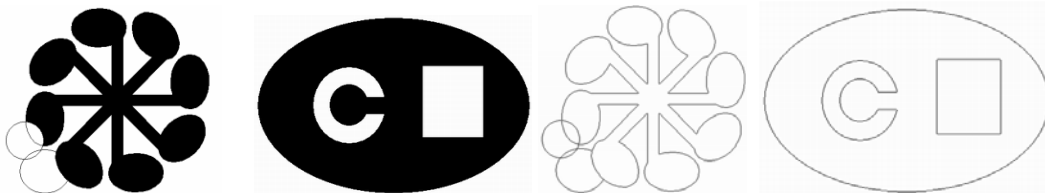
**Fig. 6.** The current point B is an intersection point. The next point is point C and remaining process point D is recorded in array IP.

### Description of Algorithm and Implementation

We trace the contour points based on the points in the contour point-pair array PP, which is generated by parallel scanning. Every point in array PP has a flag, which earmarks whether it is visited or not. All those points in array PP will be visited when all traces are completed. The algorithm is as following:

- A tracing begins from an unvisited point A in the contour point-pairs PP. Set A the current point, put it into Array CP, and flagged it as visited;
- Check A's eight neighbor points in the bitmap image to see which are process points. A process point is in between a point-pair in array PP, since all points in between a point pair are in process area. The first clockwise point B, which is in between a point-pair, will be registered as the next contour point and be put into array CP. Check whether A is an intersection point. If yes, recorded all its remaining process points in array IP. Set B the current point.
- Continue tracing the next point of B until all neighbor points of B are either non-process points (not in between a point-pair) or visited. Then stop this tracing.
- Begin new tracing from an unvisited point in array IP.
- Begin new tracing from an unvisited point in array PP.
- If all points in IP and PP are visited, then stop new tracing.
- Array CP registers a series of contour points which describe all the contour points in sequence.

Implementation is taken on a pc machine with p4 CPU 2.0GHz, 256MB of RAM. Figure 7 shows cross-sections of two models. The first two pictures are the bitmap images of cross-sections and the last two are all their contours traced.



**Fig. 7.** Cross-section of models and their contours traced.

### 3 Conclusions and Future Work

In this paper, we have presented an algorithm to generate cutting path for micro-fabrication in terms of a series of ordered contour points by tracing the full contours in bitmap images of cross-sections of a model. The algorithm includes two steps:

- Preparing step: parallel scan a bitmap image is scanned once and point-pair information on the contours of process area is recorded in an array PP.
- Main step: based on the array PP, we generate cutting paths by a series of ordered contour points.

Considering thousands of cross-sections are needed to be traced, the efficiency is important. In the algorithm, all the neighbor process points are visited and registered only once, which guarantees the efficiency and integrity the cutting paths.

It would be more accurate if the cross-sections are represented by vector images, especially when the images are scaled and transformed. We are undertaking research on applying vector images to prepare NC data for micro-fabrication.

### References

- [1] Z.W. Jiang, Y.J. Zhou, D.J. Yuan, W.H. Huang and A.D. Xia, A Two-Photon Femtosecond Laser System for Three-Dimensional Microfabrication and Data Storage, *Chinese Physics Letters* 20(12), 2126-2129, 2003.
- [2] S.D. Zenzo, L.Cinque and S.Levialdi, Run-based algorithms for binary image analysis and processing, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18(1), 1996.
- [3] T. Pavlidis, Y.T. Liow, Integrating Region Growing and Edge-detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12(3), 225-233, MAR 1990.
- [4] M.W. Ren, J.Y. Yang and H. Sun, Tracing boundary contours in a binary image, *Image and Vision Computing* 20, 125-131, 2002.
- [5] F. Chang and C.J. Chen, A component-labeling algorithm using contour tracing technique, *Proceedings of the Seventh International Conference on Document Analysis and Recognition( ICDAR'03)*, 2003.
- [6] P. Zingaretti, M. Casparoni and L. Vecchi , Fast chain coding of region boundaries(2003). *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20 (4), 407-415, 1998.
- [7] S.C. Park and B.K. Choi, Boundary extraction algorithm for cutting area detection, *Computer-Aided Design* 33, 571-579, 2001.
- [8] T.L. Chia, K.B. Wang, L.R. Chen and Z. Chen, A parallel algorithm for generating chain code of objects in binary images, *Information Sciences* 149, 219-234, 2003.
- [9] T. Pavlidis, *Algorithms for Graphics and Image Processing*, Computer Science Press, Rockville, Maryland, 143-146, 1982
- [10]. X.J. Yu and L.Z. Peng, An algorithm for getting the curve border from a black-white image, *Journal of Image and Graphics* 7A(3), 2002.
- [11]. B. Kamoun, D. Afungchui, and A.A. Chauvin, A wind turbine blade profile analysis code based on the singularities method, *Renewable Energy* 30 (3), 339-352, MAR 2005.