Proceedings of the 2005 International Conference on Simulation and Modeling
V. Kachitvichyanukul, U. Purintrapiban, P. Utayopas, eds.

IMAGE PROCESSING FOR RAPID PROTOTYPING TECHNOLOGY

Paphakorn Sunanon
Pisut Koomsap
Suchart Nachaisit

Design and Manufacturing Engineering
School of Advanced Technology
Asian Institute of Technology
Pathumthani, Thailand 12120

ABSTRACT

In order to minimize the construction time of rapid prototyping while maintaining prototype accuracy at high level, the thickness should be determined for each layer. This paper presents a contour identification algorithm to determine layer thickness for prototype construction by applying image processing technique. This approach considers the image data to search for a junction point. The maximum thickness that its contours contain no junction will be used to construct that particular layer.

1 INTRODUCTION

Rapid prototyping process (RP) is a manufacturing process that manufactures products directly from CAD model. This process converts 3D CAD models to stack of 2D contours which are used to generate machine commands to build object layer by layer. Normally, 3D CAD models are converted to STL format before they are sliced. In the STL file conversion step, the triangular facets are patched to form surface of 3D model. With the limitation of triangle facets, they cannot exactly represent curved surfaces. Those errors can be minimized by using small triangular facets, but this solution enlarges file size which unavoidably increases conversion time that conflicts with the purpose of RP. The direct slicing method has been proposed to minimize error caused by STL conversion (Kunwoo, 1999; Jamineson and Hacker, 1995; Chua and Leong, 1997; Cao and Miyamoto, 2003). This method slices layers directly from a CAD model. Since the model that composes of minimum number of layers requires less construction time; therefore, layers with maximum thickness are desirable. However, the maximum layer thickness will reduce prototype accuracy if the top and bottom contours of a layer are different. In order to minimize the construction time while maintaining prototype accuracy at high level, the thickness should be determined for each layer and a technique that will be considered is called adaptive direct slicing algorithm.

Presented in this paper is a new approach for identifying layer thickness by applying image processing technique. Next section is about previous work. Section 3 is about contour identification algorithm and local search method follows by the application of this algorithm in rapid prototyping technology. The last section is conclusion and future work.

2 PREVIOUS WORK

From the beginning of rapid prototyping technology, the stereolithography (STL) file format is used to convert a CAD model to a tessellated model before slicing it to layers for prototype construction. This STL conversion only approximates the original model with triangular facets. Therefore, accuracy of the tessellated and the size of STL file are depended upon the number and sizes of triangle facets. (Kunwoo, 1999; Jamineson and Hacker, 1995; Chua and Leong, 1997; Cao and Miyamoto, 2003). Compressed STL (CS) file was proposed to reduce the STL file size (Zhang, Han and Huang, 2003). CS file can be used to convert back and forth between CAD model and tessellated model. Since the size of CS file is much smaller than the STL file; it is faster and easier to convert. Also, the idea of direct slicing was presented to eliminate STL conversion step and at the same time to minimize the error from STL conversion. This direct slicing concept has been implemented with many available commercial software including Unigraphic (Samieson and Hacker, 1995), PowerSHAPE (Chen et al., 2001) and AutoCAD (Cao and Miyamoto, 2003). Beside, using these commercial softwares, there is also an attempt to implement the direct slicing with STEP file (Starly et al., 2004). The aforementioned implements assume equal thickness.

To improve the prototype accuracy, adaptive layer slicing algorithm was introduced. Samieson and Hacker (1995) presented model slicing by geometric checking. If geometric is not satisfied, the thickness is reduced by haft. Sabourin et al. (1996) proposed the adaptive slicing algorithms by using stepwise uniform refinement that is related with the cusp height of each layer. This approach equally slices the model by slabs with maximum thickness. Then each slab is subdivided again until it satisfies the minimum cusp height. Also, Hope et al. (1997) presented the approach which encourages for minimizing the cusp height by slicing model with sloping layer.

There is possibility of applying image processing technique in rapid prototyping technology, specifically, chain code concept. The chain code was introduced for recording a shape contour using straight line (Awcock and Thomas, 1996; Gonzalez and Woods, 1993). This code concept was used in active chain algorithm to detect region boundaries (Karin and Bulut, 2001). By converting chain code to pcode, the tracing algorithm was proposed to trace the outer and inner boundaries (Ren, Yang and Sun, 2002). Chain code was also applied to simultaneously trace the outer and inner contours (Tsorng, Kuang and Lay, 2003).

This paper presents the application of image processing technique for rapid prototyping technology. The chain code concept is applied to trace layer contour for determining layer thickness that algorithm will be described in next section.

3 CONTOUR IDENTIFICATION ALGORITHM

This section presents the algorithm of contour identification when two contours are projected on the same plane. The concept of this algorithm is a contour searching by pixel consideration. Pixels, belonging to the contours on a plane are evaluated whether or not they are junction points. If one junction point is found, that means the two contours are not identical.

As illustrated in the Figure 1, the steps of contour search algorithm are the following:

1. Identify the minimum boundary that contains all contours projected on a plane to reduce searching time.
2. Seek for the first pixel on the contours. In this step, the algorithm searches for the first pixel, belonging to the contours.
3. Evaluate pixels. In this step, the algorithm evaluating whether or not a current pixel is a junction. Unless it is the junction, the search will continue with determining the next pixel to be evaluated. This search process will be repeated until it arrives at the staring pixel. The details of this step are presented in the next section.
4. Output the algorithm decision. In this step, if a junction is found, the two contours are different. On the other hand, if no junction is found, the two contours are identical.

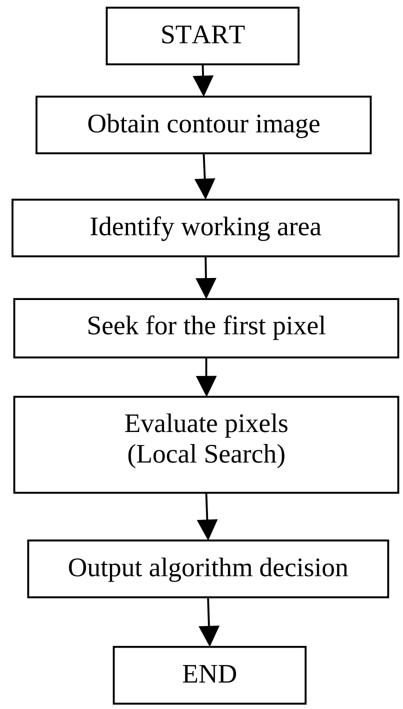


Figure 1: Contour identification steps

4 LOCAL SEARCH METHOD

A local search method is developed for seeking a junction on the contours. This method considers 9 pixels, where the middle pixel is a considering point as shown in Figure 2. Information from eight adjacent pixels is used to determine whether or not the middle pixel is a junction. If it is not a junction, one of these eight adjacent pixels will be the next pixel to be evaluated. The eight adjacencies are assigned with numbers 0 to 7, by using the chain code concept of image processing (Awcock and Thomas, 1996; Gonzalez and Woods, 1993). The numbers are assigned in a counterclockwise direction, staring from the pixel on the right hand side, as shown in Figure 2.

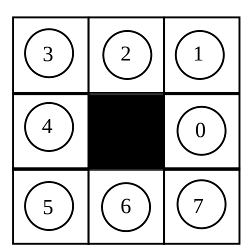


Figure 2: Number of 8 adjacent pixels around considering pixel.

In the searching process, the algorithm checks all adjacent pixels which there values are assigned to be binary numbers. The values of the pixels belonging to the two contours are 1s. Otherwise their values are 0s. Traveling direction is used along with the information of the eight adjacencies to identify a junction. Traveling direction is the direction from previous considered pixel, which is one of those eight pixels. The previous considered pixel and two pixels on its left and right sides can be neglected because they have already been evaluated in the previous step. Consequently, the algorithm will consider only the remaining five pixels and by doing this, searching time and memory usage are decreased.

To identify a junction, there are two cases: one “1” pixel case and more than one “1” pixels case. If there is only one pixel with the value of 1, the considering pixel is not a junction, as shown in Figure 3. On the other hand, when there exists more than one pixel with the value of 1, traveling direction is used in consideration. If the traveling is on the diagonal directions, a considering pixel will be a junction when there is at less one empty pixel between the two “1” pixels, as shown in Figure 4. In case of traveling on the horizontal or vertical direction, the similar rule can be applied, except 3 special cases, presented in Figure 5. These 3 cases will require information from all eight adjacent pixels and pixels in the neighborhood of “1” pixels to determine a junction.

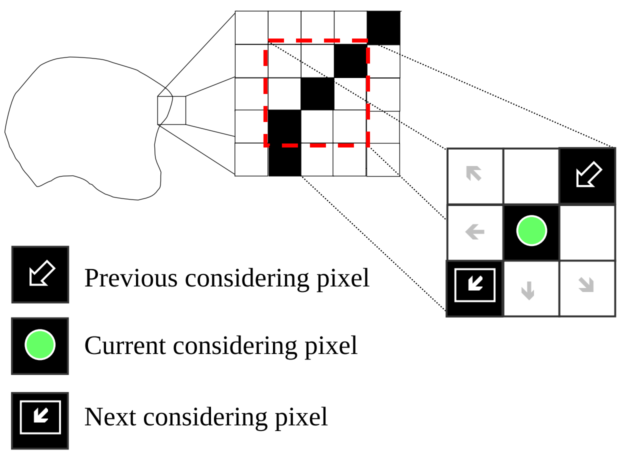


Figure 3: Contour with no junction point

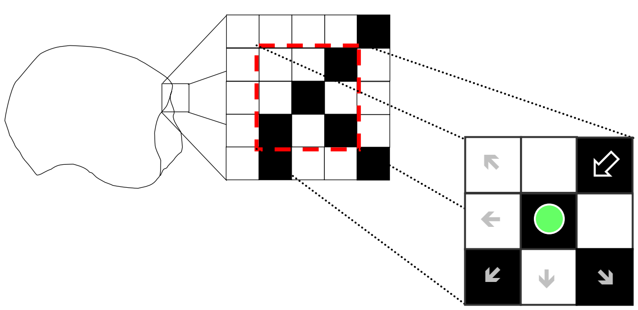


Figure 4: Contour with junction point

Unless the junction is found, the algorithm will determine the next pixel to be evaluated. To determine the next pixel, there are two cases: First case is when only one pixel of value 1 is found. That pixel will be the considering pixel, referring to Figure 3. Second case is when more than one “1” pixel exists. Candidates for the next considering pixel, in this case, are the pixels at both ends of “1” pixels chain. If one end of the “1” pixel chain is at one of even position (0, 2, 4 or 6 referring to Figure 2), that pixel will be the next considering pixel, as shown in Figure 6. If neither ends of the “1” pixels chain is at the even position, the end pixel that is in the traveling direction will be the next considering pixel. However, if the end pixels are not in the traveling direction, adjacent pixels of both ends are required to determine the next considering pixel. Also, if both ends of the chain are at the even positions, adjacent pixels of both ends are required to identify the next considering pixel, as shown in Figure 5. Figure 7 illustrates the flow diagram of local search algorithm.

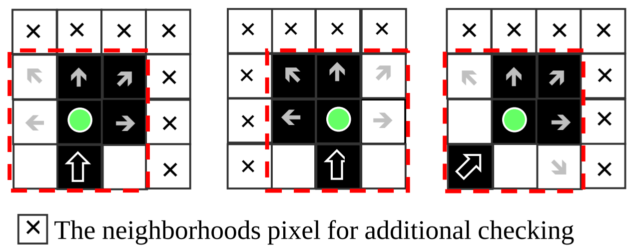


Figure 5: Three special cases

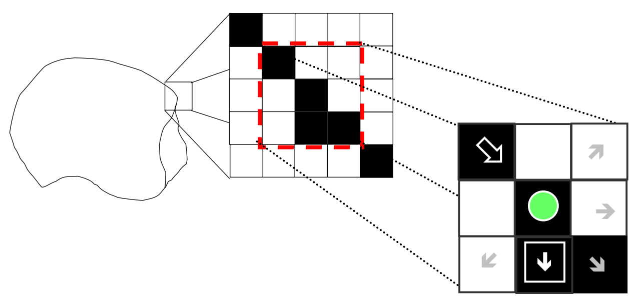


Figure 6: The case of more than one “1” pixels

5 APPLICATION OF CONTOUR IDENTIFICATION ALGORITHM FOR RP

The contour identification algorithm determines identical contours by searching for a junction point on contours. This algorithm can be applied to determine the layer thickness for adaptive direct slicing technique of RP. According to rapid prototyping concept, the maximum layer thickness building is expected. The maximum layer thickness is assigned when the top and bottom contours are identical. That means no junction point is found, as shown in Figure 8.

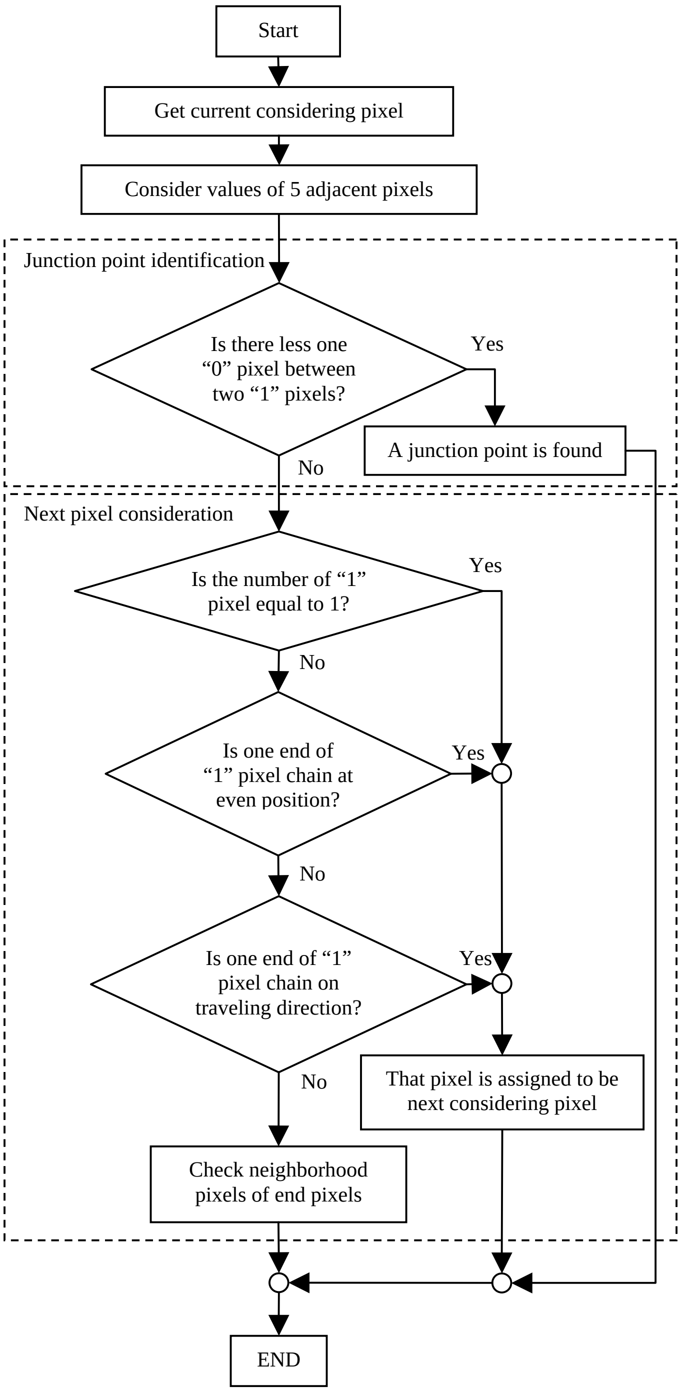
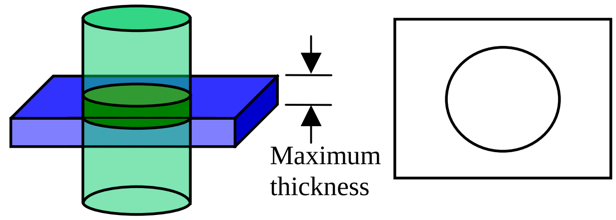


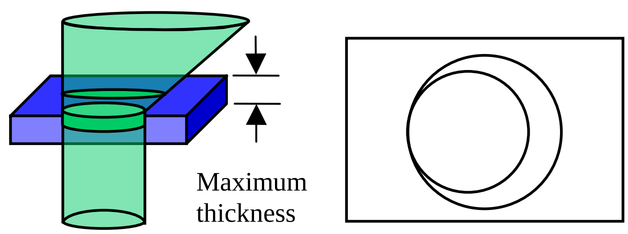
Figure 7: Local search algorithm



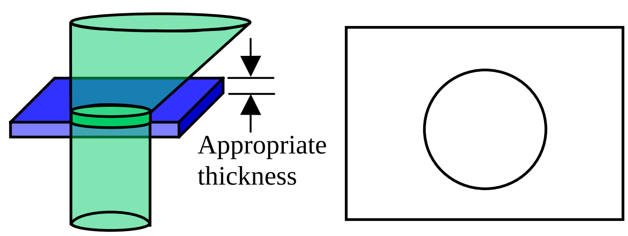
8a. Model slicing 8b. Layer contour

Figure 8: The identical contour of the maximum layer thickness

However, if a junction exists, as shown in Figure 9b, the layer thickness will be reduced until no junction is detected. The maximum thickness that contains no junction will be used to construct that layer. In case that the junction exists even with the minimum thickness, the outer contour will be used in construction at the minimum thickness.



9a. Model slicing with the maximum thickness 9b. Different layer contours



9c. Model slicing with the appropriate thickness 9d. Identical layer contours

Figure 9: Adaptive direct layer slicing by using contour identification algorithm

6 CONCLUSION

Contour identification algorithm is developed by applying the image processing technique. This algorithm considers the image data to search for a junction point. In the searching process, pixels on the contours are evaluated whether or not there are junction points. The search will continue until the first junction is found or the next considering pixel is the starting pixel. By searching the junction point on the contour, this algorithm can be applied to support the adaptive direct slicing technique in RP. The maximum thickness that contains no junction will be used to construct that particular layer. This algorithm can also be applied to detect a junction on multiple contours.

7 FUTURE WORK

Currently, the contour identification algorithm can identify various contours, including primitive contours and arbitrary contours. However, there is a special case, the severe acute angle contour, which will require additional work. The next generation of this algorithm will also be able to detect two different contours that do not intersect (e.g., donut shape and two contours completely apart).

REFERENCES

Kunwoo L. 1999. *Principles of CAD/CAM/CAE*. Addison Wesley Longman. USA.

Chua C. K. and Leong K. F. 1997. *Rapid Prototyping: Principles & Applications in Manufacturing*. John Wilet & Sonc. Singapore.

Jamieson R. and Hacker H. 1995. Direct slicing of CAD models for rapid prototyping. *Rapid Prototyping Journal* 1 (2): 4-12.

Zhang L. C., Han M. and Huang S. H. 2003. CS file – An improved interface between CAD and rapid prototyping system. *International Journal of Advanced Manufacturing Technology* 21: 15-19.

Cao W. and Miyamoto Y. 2003. Direct slicing form AutoCAD solid models for rapid prototyping. *International Journal of Advanced Manufacturing Technology* 21: 739-743.

Chen X., Wang C., Ye X., Xiao Y. and Huang S. 2001. Direct slicing form PowerSHAPE models for rapid prototyping. *International Journal of Advanced Manufacturing Technology* 17: 543-547.

Sabourin E., Houser S. A. and Bohn J. 1996. Adaptive slicing using stepwise uniform refinement. *Rapid Prototyping Journal* 2 (4): 20-26.

Starly B., Lau A., Sun W., Lau W. and Bradbury T. 2004. Direct slicing of STEP based NURBS models for layered manufacturing. *Computer-Aided Design* xx: 1-11.

Hope, Roth R. L., R. N. & Jacobs P.A. 1997. Adaptive Slicing with sloping later surfaces. *Rapid Prototyping Journal* 3 (3): 89-98.

Awcock G.J. and Thomas R. 1996. *Applied Image Processing*. McGraw-Hill. Singapore.

Gonzalez C. R. and Woods E. R. 1993. *Digital Image Processing*, Addison – Wesley.

Kaygin S. and Bult. 2001. A new one-pass algorithm to detect region boundaries. *Pattern Recognition Letters* 22: 1169-1178.

Ren M, Yaung J and Sun H. 2002. Tracing boundary contours in a binary image. *Image and Vision Computing* 20: 125-121.

Tsorng L. C., Kuang B. W. and Lay R. C. 2003. A parallel algorithm for generating chain code of objects in binary images. *An international journal of information sciences* 149: 219-234.

AUTHOR BIOGRAPHIES

PAPHAKORN SUNANON is a doctoral student in the Design and Manufacturing Engineering, School of Advanced Technologies, Asian Institute of Technology, Thailand. She received her M.Eng. in Manufacturing Systems Engineering from King Mongkut's University of Technology Thonburi, Thailand and B.Eng. in Industrial Engineering from Suranaree University of Technology, Thailand. Her doctoral research interest is adaptive direct slicing for layer manufacturing. Her email address is < Paphakorn.Sunanon@ait.ac.th >

PISUT KOOMSAP is currently an Assistant Professor in the Design and Manufacturing Engineering, Asian Institute of Technology. He received his Ph.D. in Industrial and Manufacturing Engineering from the Pennsylvania State University. His research interests are sensing and control for manufacturing, rapid prototyping, and laser applications. His email address is < pisut@ait.ac.th >

SUCHART NACHAISIT is a doctoral student in the Design and Manufacturing Engineering, School of Advanced Technologies, Asian Institute of Technology (AIT), Thailand. He received his M.Eng. in Manufacturing Systems Engineering from AIT and B.Eng. in Industrial Engineering from Rajamangala Institute of Technology, Thailand. His doctoral research interest is focus on rapid prototyping technique. His email address is < Suchart.Nachaisit@ait.ac.th >