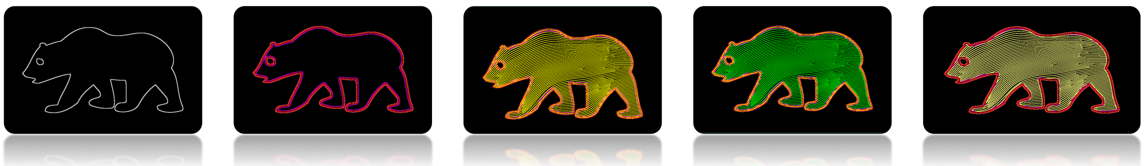
Milling Tool-Path based on Micrography

Youngwook Paul Kwon



I. Introduction

The objective of my research is to design and develop a unique tool-path planning system based on Micrography. CNC milling machines are extremely accurate, and fully automated; they are used widely in manufacturing industries. Thus, the some improvement of their tool-paths can have a widespread impact in this world.

Figure 1 shows that tool-path planning for a simple pocket leads to various different alternatives, which have different time and energy requirements. This leaves us a big room for improvement in energy efficiency. For example, CNC milling machines have different masses of table axes. Thus, movement in one direction consumes more energy than that is consumed in the other direction. This characteristic tells us that it is beneficial in tool-path planning to have more movements in the advantageous axis, as much as possible, from the available alternatives. In addition, considering that the acceleration from rest and deceleration consume more energy than when the milling machine is moving at a constant velocity, it is preferable to have longer paths and fewer stops during cutting (Figure 2).

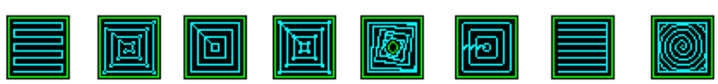


Figure 1: Different alternative tool-path strategies offered in popular CAM software tool MasterCAM

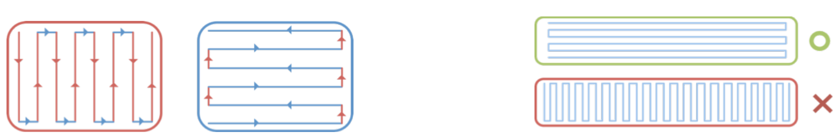


Figure 2: Alternative tool-path strategies with different performance characteristics

This tells us that a well-improved tool-path planning system can bring improvement in energy efficiency if the tool-path is appropriately designed. Common characteristics of energy efficient tool-paths include making fewer and gradual changes in tool-path cutting direction, avoiding sharp corners, having longer paths with near constant cutting load, and so on.

Surprisingly, an innovative approach for designing such energy efficient tool-path is based on an algorithm for digital micrography, a special type of calligraphy created from an input image and minuscule text (Figure 3). Designing textual layout for digital micrography has geometric constraints such as it should convey the input image, be readable and appealing and not be intersecting and overlapping. These conditions in micrography and those in energy efficient tool-paths for milling machines are actually very similar. Both problems need to generate lines filling the given space with some conditions on curvature, changes, length, spacing, etc.

Therefore, I think the tool-path planning based on micrography is beneficial enough to research. However, for energy efficient tool-path, quite deep research and tangible knowledge about digital micrography will be required. In this reason, this project is limited to come up with those innovative energy efficient tool-paths at this time. Thus this project, as a beginning step, will focus on developing a basic tool-path planning that can be derived from the result of micrography for a fundamental 2.5D pocket, and aim to conduct the actual cutting of it.

Figure 3: The examples of Digital Micrography

II. Methods for Tool-Path Processing

A. Input

The input image that I wish to cut is shown in Figure 4. It shows how streamlines are look like, as well. The vertices information of the input image and streamlines are the input of this project. The vertices of streamlines are generated by digital micrography and offed by Ron Maharik at University in British Columbia, who is the author of the digital micrography paper [1]. All these procedure are executed with OpenGL on Microsoft Visual C++ 2008.

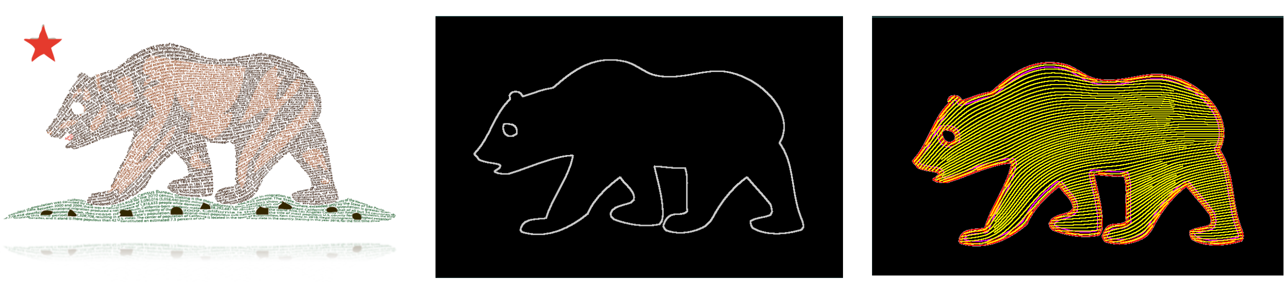


Figure 4: (a) Motivated image in the Digital Micrography paper. (b) Drawing of input image vertices (497 vertices). (c) Drawing of input streamlines (106 lines). Each lines has the diverse number of vertices.

B. Offset

As the first step, the offset by the tool radius should be obtained for the outer boundary of this bear pocket. For convenience, let call the radius offset as 'center boundary', and the diameter offset as 'inner boundary'. I developed this offset procedure with reference on some open source [2].

Figure 5: Outer (a given input), center (offset by radius), and inner (offset by diameter) boundary

C. Trim

If we want to use streamlines as tool-paths of pocket, streamlines should be trimmed by the inner boundary in order not to exceed the outer boundary. This trim algorithm consists of two steps. Above all, for each streamline, the starting vertex is tested whether it is inside of outside. Every vertex has boolean variable for in-out information, and 1 means it is inside of the given polygon, and 0 means outside.

Next, from the starting vertex, every line segment – the line between vertex i and $i+1$ - on the streamline is conducted intersection test with the inner boundary. If there is no intersection, the boolean value is maintained from vertex i to vertex $i+1$, otherwise the value is toggled.

The result is shown below. Yellow line means they are outside of the inner boundary and green means inside lines.

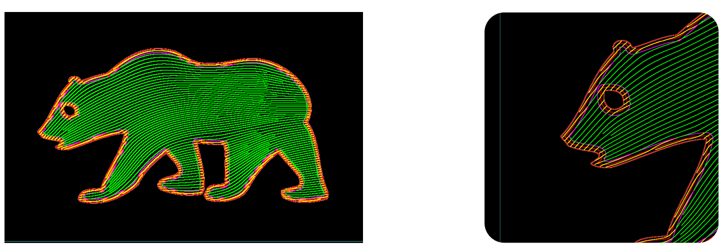


Figure 6: Streamlines trimmed by the inner boundary.

D. Connect

Instead of cutting with these individual inner streamline tool-paths, I aimed more connected tool-paths. Again, this procedure is not dealing with optimal or energy efficient tool-path connecting algorithm. If a streamline is trimmed by the inner boundary, they can be connected to their neighboring line. If it cannot find a possible neighbor line, the connecting path stops to find more line and pushed to tool-path stacks. As a result, 27 connected tool-paths are generated in this case.

Figure 7: Connected tool-paths.

E. NC code

Based on the connected tool-path, NC codes for CNC milling machine are automatically generated. User can decide the length and width in inches, and ratio of the bounding box to the size. Then this system transforms the vertices properly and prints NC codes. In this project, I aimed 6x4 inches.

Before actual cutting, I conducted simulation using *NCSim*.[3] In this process, I could find the range of tool diameter (15~20mm) that reflects the space between stream lines. Fortunately, the machine shop has 1/16 in diameter flat tool. [4] Finally I re-generated NC codes because the offset process reflects tool diameter.

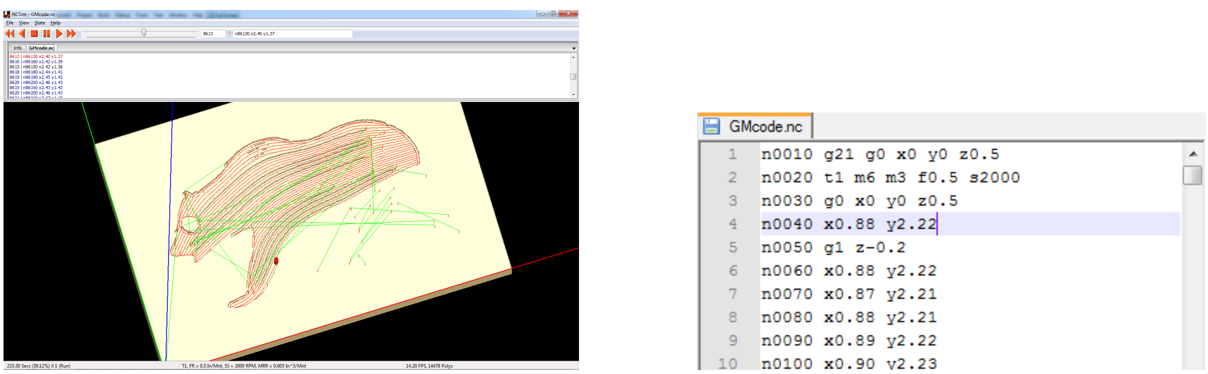


Figure 8: (a) *NCSim*, CNC milling machine simulator. (b) Automatically generated NC code.

III. Actual Milling

With this result, I started to cut actually. These works is done in the Mechanical Engineering Student Machine Shop, 166 Etcheverry. [4]

Figure 9: ME Student Machine Shop, 166 Etcheverry. The milling machine used actually.

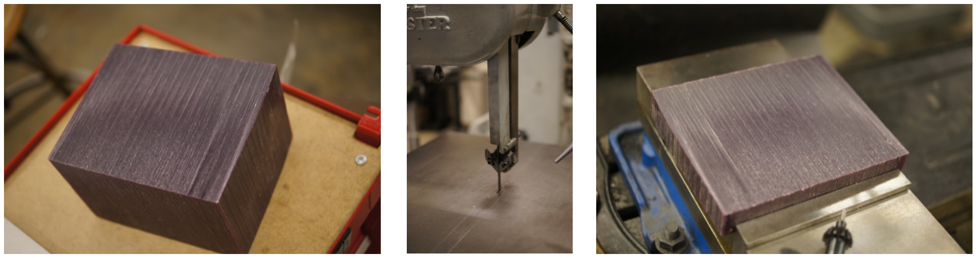
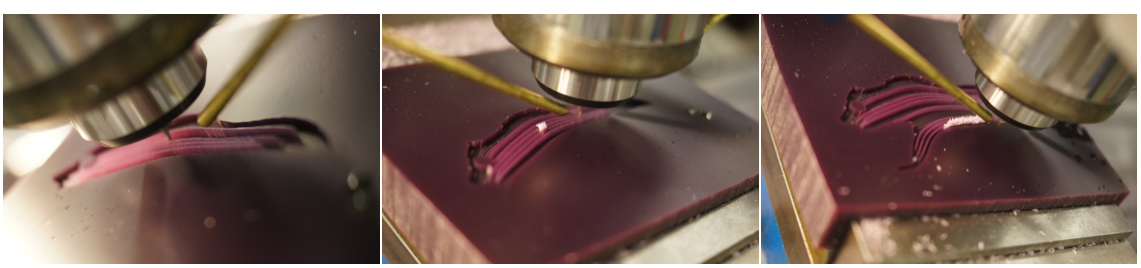


Figure 10: The original wax block is cut to obtain a 6x4x1 board. The board is located on the milling table.

Figure 11:Surface refining process.

Figure 12: (a) 1/16 diameter tool. (b) Origin-coordinate setting.

Figure 13: Monitor of milling machine and cutting simulation



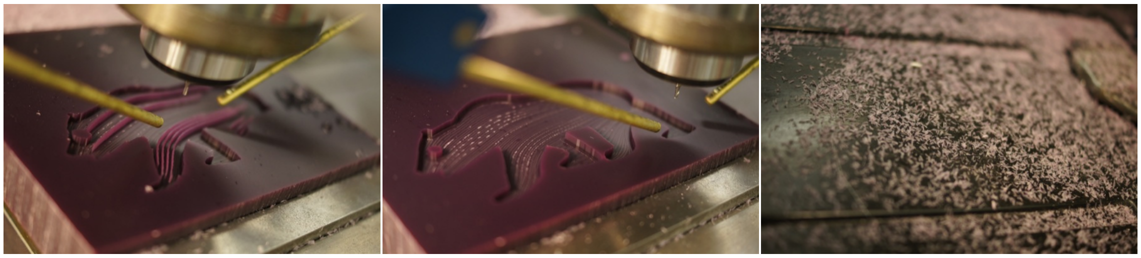


Figure 14: Cutting processing

IV. Results and Discussion

Finally, I could obtain a bear cut on the surface of a wax cube, whose cutting pattern is very unique based on digital micrography.

Figure 15: The Final product

Through this project, I had opportunity to design tool-paths based on digital micrography and to handle the real CNC milling machine. This tool-path is one of the diverse possible alternative tool-path planning strategies and has unique cutting pattern. I could understand more about the milling machine, and while developing offset, in-out test, and intersection test algorithms from the scratch, I found and learned about interesting method including the winding number method. Many papers that I read in the CS285 teaches how we can handle and solve the problems we face.

Even though, as a course project, this set of procedures are focusing on actual manufacturing, I would like to explore in direct input handling of digital micrography and developing more efficient algorithm in selecting and connecting streamlines in future. I believe this – the tool-paths based on the advantages of micrography streamline – will bring beneficial effect in saving energy in manufacturing, and offer more idea and improvement in 3D dimension cutting beyond 2.5 dimension.

V. Reference

[1] R. Maharik, M. Bessmeltsev, A. Sheffer, A. Shamir, and N. Carr, "Digital micrography," SIGGRAPH '11: SIGGRAPH 2011 papers, Aug. 2011.

[2] Polygon offset algorithm source code, http://kingkong.me.berkeley.edu/~xchen/

[3] CNC Milling machine simulator, http://www.cs.technion.ac.il/~gershon/NCSim/

[4] ME Student machine shop, http://www.me.berkeley.edu/new/Shop/enter.html