Coverage Path Planning Application on FDM 3D Printer

Instructor: Dr. Yong Chen
Yuan Li, Mingjun Yao
Viterbi School of Engineering, University of Southern California
Date: May 6th, 2018

Abstract

This paper aims to apply one kind of grid-based coverage path planning method to Fused Deposition Modeling(FDM) 3D printer. In this paper, the conception and development of 3D printing and the most widely used technology-Fused Deposition Modeling(FDM) is introduced as background. Then the problem of one kind of traditional FDM nozzle path's low printing efficiency is proposed with two main reasons. As a solution to this problem, coverage path planning (CPP) is then recommended. As an effective kind of CPP, Path Cost Evaluation is selected as the algorithm to increase the printing speed and improve the boundary printing quality. The program implementation process and the results of simulation/experiment are presented then. Finally, the lesson learned from this study and final works are discussed.

Key words: *FDM 3D printing; Coverage Path Planning; Path Cost Evaluation; Time saving; Boundary Optimization;*

1. Introduction

1.1 Background

1.1.1 3D Printing Technology

3D Printing has been a popular field for many years. The concept of 3D printing, also referred to as additive manufacturing (AM), rapid prototyping (RP), or solid-freeform technology (SFF), was developed by Charles Hull [1]. It is the opposite of subtractive manufacturing which is cutting out or hollowing out a piece of metal or plastic with a machine. In 3D printing, the creation of a 3D printed object is achieved using additive processes in which an object is built by laying down successive layers of material. Each of these layers can be seen as a thinly sliced horizontal cross-section of the final object. Printing layer by layer, a 3D object can finally be created. The most popular 3D printing technologies include Fused Decomposition Modeling(FDM), Selective Laser Sintering (SLS), Stereolithography (SLA) and Laminated Object Manufacturing (LOM).

Breaking the shackle of subtractive manufacturing, 3D printing enables production of much more complex shapes with much less material. Due to its incomparable strength, 3D printing technology has been effectively applied to many areas. Medical applications of 3D printing rose in the early 2000s with the production of dental implants and prosthetics. Moreover, 3D printing has developed applications in automotive and aerospace industries for printing car and airplane parts models, in the architectural world for printing structural models and even in gun prototyping and manufacturing processes for the military.

1.1.2 Fused Deposition Modeling

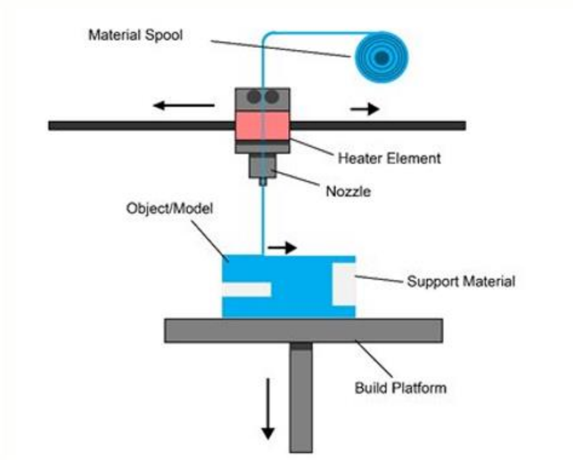

Figure 1. FDM 3D Printer [2]

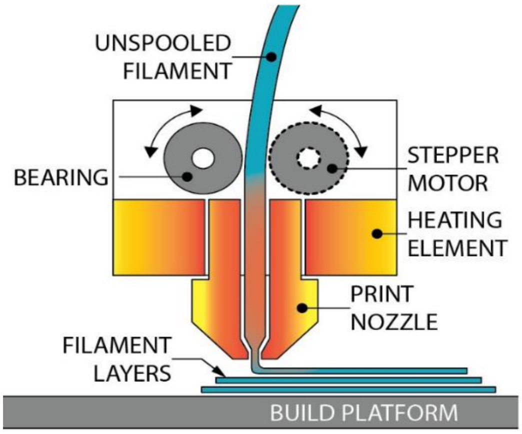

Figure 2. FDM Extruder [3]

Developed by Scott Crump of Stratasys, Fused Deposition Modeling(FDM) is one of the most widely used 3D printing technologies today. As is shown in figure 1, a typical FDM 3D printer consists of a build platform, the material filament and an extruder. The filament material most commonly used by a FDM 3D printer are polylactide(PLA) and acrylonitrile butadiene styrene(ABS). Figure 2 shows the detailed internal structure of a FDM extruder. It is composed of a bearing and stepper motor, a heating element and a nozzle. The filament from material spool is inserted through the extruder. When it's printing, the filament is fed in by the stepper motor to form the needed pressure for extrusion, then melted in the heating element and finally extruded out of the nozzle onto the build platform. Then the extrusion nozzle moves over the build platform horizontally and vertically (in XY direction), "drawing" a cross section of an object onto the platform. This thin layer of plastic cools and hardens, immediately binding to the layer beneath it. Every time the whole layer's printing is finished, the build platform will move a unit in Z direction, usually by about one-sixteenth of an inch, so that the extruder is able to begin the printing job for the next layer. Printing layer by layer, a 3D object is finally created. In essence, the process of FDM 3D printing is the coordinated work of X, Y, Z and E (the feeding of filament controlled by stepper motor) axis motion.

1.2 Problem Statement

In general, FDM printing time depends on the size of the object being manufactured. Small and tall, thin objects are printed quickly while larger, more geometrically complex objects usually take longer time. Compared to other 3D printing methods, such as SLA and SLS, FDM is very time-consuming. However, even within the same FDM printer, different nozzle path can generate significant difference in printing efficiency. Therefore, the time-consuming problem lies in the FDM printing system.

1.2.1 FDM printing system

The way FDM 3D printing system works in is shown in figure 3. To print a three-dimensional object, the first step is to create a 3D digital file of the object. This can be completed via CAD software such as Solidworks, Pro/E, etc. The file is then converted to an .STL file format. This file format is

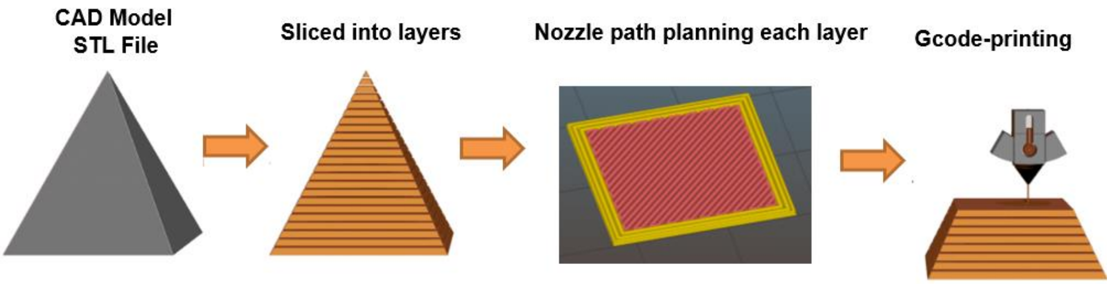

Figure 3. FDM Printing System

developed by Hull at 3D systems and it has been accepted as the standard for data transfer between the CAD software and a 3D printer. The .STL file stores the information for each surface of the 3D model in the form of triangulated sections, where the coordinates of the vertices are defined in a text file [1]. The next step is to slice the .STL model into a sequence of 2D horizontal cross sections which allows the nozzle path planning of each layer to be calculated. After that, gcode, a kind of code that can be identified and implemented by FDM 3D printer, is generated. The final step is to input the generated gcode into open-source FDM 3D printer and print the model. The time-consuming problem of FDM 3D printing has always been attributed to the nozzle path planning algorithm within each layer, which is the stage after slicing and before the generation of gcode. In fact, development of better nozzle path planning algorithms to improve the printing speed and quality is an active area recently.

1.2.2 Problem in a traditional nozzle path

One of the traditional nozzle path is firstly analyzed in this study. Given a layer pattern to be printed like figure 4, where the white space is the area to be printed, the red lines are the printing path of the nozzle and the blue polygon which is called "obstacle" in this study, is the area not to print. Therefore, when it's printing, the nozzle of the FDM 3D printer moves back and forth with constant intervals between two parallel path lines to cover all the points in the free space. When the nozzle

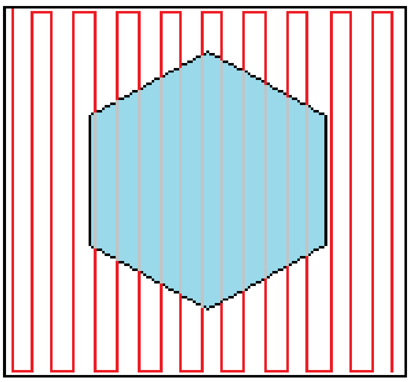

Figure 4. One kind of Traditional Nozzle Path

encounters the boundary of the obstacle, however, it will stop extruding filament, accelerate through this obstacle area and start printing again when getting out of the obstacle.

This method is time-consuming for two reasons. In the first place, nozzle going through the non-print segment is a waste of time, even though it is accelerating. It's just a way to reduce the time

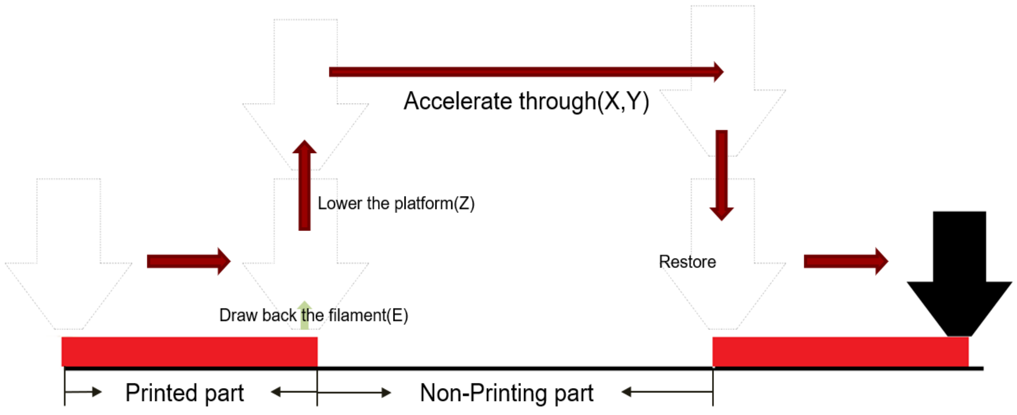

Figure 5. Nozzle Preparing

loss. Secondly, each time the nozzle meets the obstacle boundary, there will be a process called nozzle preparation. The whole process of nozzle preparing is shown in figure 5. When the nozzle prints with the planned path and encounters the obstacle, it will stop printing and accelerate through the non-printing area. However, before accelerating, the nozzle will first draw back the filament a little bit to avoid additional filament falling on the platform. This is the E axis motion controlled by the stepper motor. After that, to avoid unexpected touch with already printed part, there will be a Z direction movement of the build platform to increase the distance between nozzle and platform. After accelerating through this area, another Z and E movement will be implemented respectively to restore the nozzle to former state so that the printing can be resumed. Although this single nozzle preparing process takes very little time, when the nozzle encounters with the obstacle for many times, the accumulated time can be very long.

2. Approaches used in the project

2.1 Desired Algorithm-Coverage Path Planning

This study aims at finding time-saving algorithm regarding to the two aspects of the problem. Thus, the desired algorithm should be one that is able to bypass the obstacle area in the printing layer and reduce the nozzle preparing times. After literature review, coverage path planning(CPP) is selected as the solution.

Coverage path planning is the task of determining a path that passes over all points of an area or volume of interest while avoiding obstacles [4]. The general criteria for CPP are listed as following.

- The path must cover all the point in the area.
- The path must avoid all obstacles.
- The path should be formed by simple motions like straight lines and circles.

Aside from meeting all the criteria listed above, an optimal CPP should have less repetition and less turns. CPP can be classified mainly into two different types, "on-line" and "off-line". "On-line"

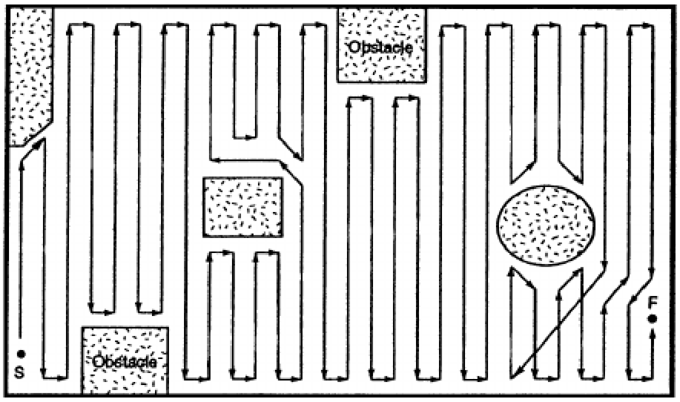

Figure 6. 2D Off-line CPP [5]

means that the environment is unknown so that it is necessary to use sensors to detect the real-time environment and decide the path dynamically. On the contrary, "off-line" in this problem means the CPP in the 2D environment known ahead of time. In this study, the main study objective is 2D off-line CPP. An example of 2D off-line CPP is shown in figure 6 where in the two-dimensional area, all the free space and obstacles are known.

2.2 Path Cost Evaluation

The solution to the problem proposed before is, specifically, Path Cost Evaluation, which is a kind of grid-based decomposition. Grid-based decomposition method is one type of 2D off-line CPP. As is shown in figure 7, it divides the whole target area into square cells and can easily decide which parts are required to visit. The rectangle area is the working space and the red ellipse inside is the obstacle to be avoided. When the square cell contains only part of the obstacle, it’s also marked as obstacle. Otherwise, the cell is marked as the area to be visited. The advantage of this method is obvious: it’s easy to construct and can be generally applied to all kinds of obstacle shape. But the biggest limitation is clear as well: it has resolution problem, which can be vital sometimes.

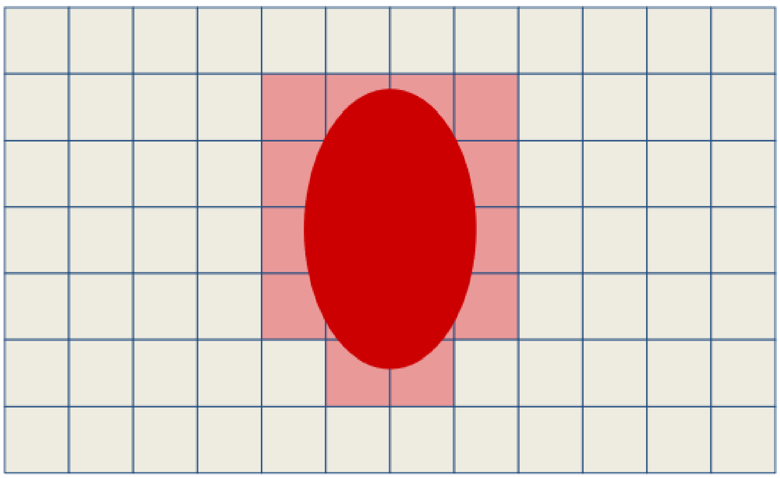

Figure 7. An Example of Grid-based Decomposition

As is mentioned before, Path Cost Evaluation(PCE) is one kind of effective grid-based CPP method. This algorithm labels every cell before doing path planning. As is shown in figure 8, each cell is initially labeled a cost number, which equals to 8 minus the number of uncovered cells in all its

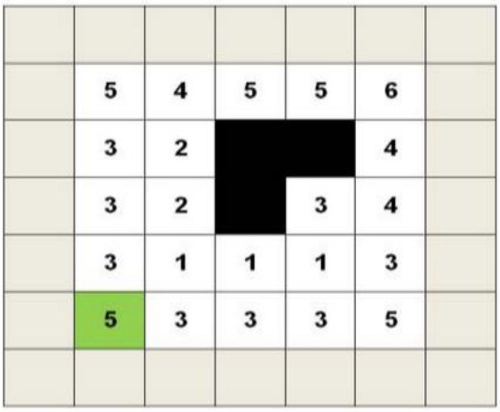

Figure 8. PCE Initial Label

eight neighbor cells. For instance, the green cell has 8 neighbor cells, and only three of them are uncovered. Therefore, the ‘coverage cost’ of the start cell is 5. Then a starting cell is set, like the green cell in figure 8. When it’s doing path planning, the pre-determined start cell is firstly marked as covered. Then, the uncovered neighbor cell with the highest coverage cost will be chosen as the next one to cover. After that, this chosen cell will be marked as covered and all the neighboring cells’ cost number will be updated, as is shown in figure 9. This process will be looped until all the cells’ cost number becomes 8, which means the whole area has been completely covered, just as figure 10 shows. Another feature is that when all the neighboring cells of the current cell has already been covered and the area still has uncovered cell somewhere else, it requires the back tracing. This situation is called "singularity", which is shown as the double arrow in figure 8.

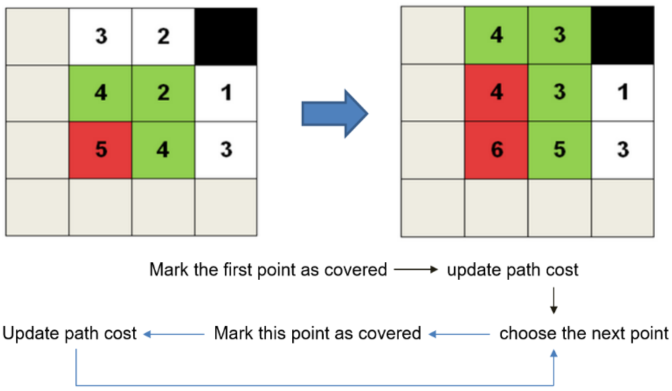

Figure 9. PCE Path Developing Algorithm

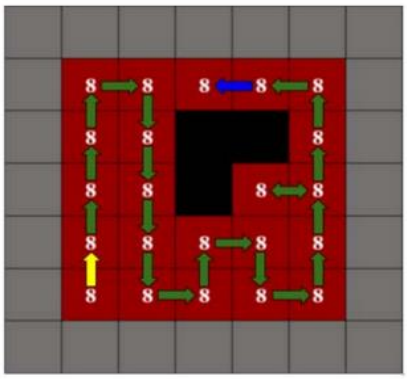

Figure 10. PCE Coverage Result

This study uses PCE as the solution to the time-consuming problem of FDM 3D printing. The pattern of each printing layer is considered as PCE’s target area where the covered area represents the already printed area, the obstacles is the non-printing area and the covering order is the nozzle path of FDM 3D printer.

3.Objectives and Plan

3.1 Project Objectives

The objective of this study is sub-divided into three parts:

- Write code for Path Cost Evaluation algorithm and successfully apply it to FDM 3D printer
- Compare the printing time of two different path planning algorithms by doing theoretical calculations, simulations and experiments. Analyze the results.
- Modify the original algorithm to improve the boundary printing quality and minimize the resolution effect of grid-based algorithm.

3.2 Project Plan

Table 1 shows the detailed plan of this study.

	Start Date	End Date	Duration(days)
Find an available open-source FDM 3D Printer	3/19	3/26	7
Understand the printing code inside	3/27	4/2	7
Write the code for coverage path planning algorithms	4/3	4/16	14
Combine different methods	4/17	4/23	7
Do experiments and edit the code	4/24	4/30	7
Do comparison and testing experiments	5/1	5/1	1
Prepare for the demo footage and slides	5/2	5/3	2

Table 1. Project Plan

4. Implementation Notes

This study uses MATLAB to develop the program. In the program, a black-and white jpeg is used to represent the pattern of printing layer. White space is printing area and black is non-printing area, as is shown in figure 11.

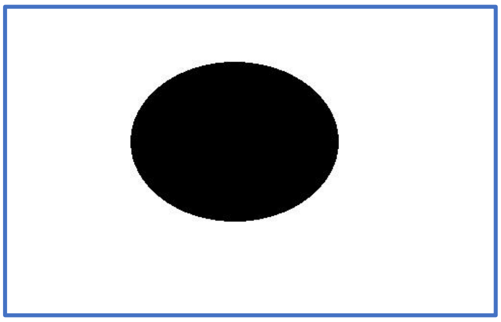

Figure 11. Printing pattern representation in the program

Firstly, the figure is read in. Inside MATLAB, jpeg is saved as 4 groups of numbers representing different colors at each point. Since this study uses black and white figure, the numbers in each group are the same. Secondly, the pixel size should be decided for the algorithm. For example, if the pixel size is set as two, the 4 pixels within a two by two square in the original figure will become a new pixel for the algorithm. Then, with the path planning number group prepared, each cell will be labelled by the original coverage cost value as the theory defines. After labelling every point, a number group is ready for coverage path planning. Also, to improve the boundary quality of the path planning, the precondition is to find the boundary. Therefore, the program is split into two parts. The first part is applying the coverage cost theory, and the second part is to find every boundary. After the above process, two separated paths are generated, the printing path and the boundary path. Apparently, they need to be combined. The most important part is to combine the two paths at the right point. Finally, the ultimate path planning will be implemented, and it will be converted to gcode which can be printed on the FDM 3D printer. The implementation process is shown in figure 12.

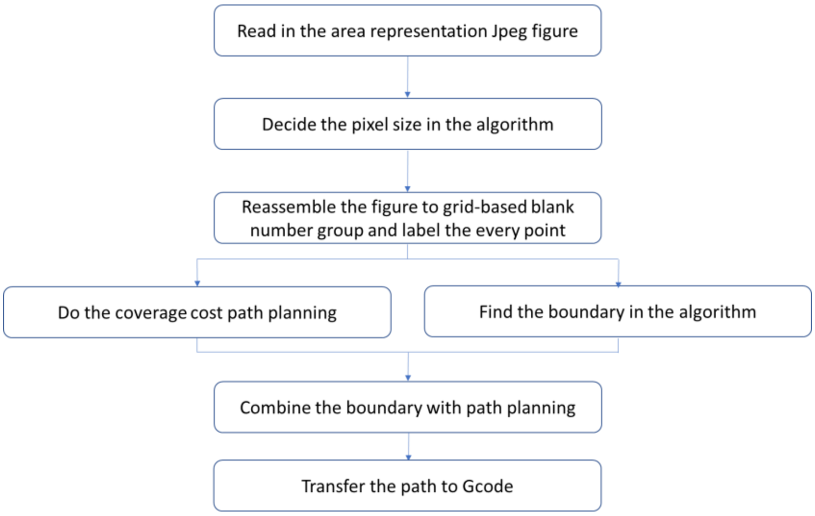

Figure 12. Overview of Implementation Process

5. Project Results and Conclusions

5.1 Simulation and Experiment comparison

Since real world 3D printing requires multiple settings like nozzle temperature, warm platform temperature, the thickness of each layer, the understand of printing materials and so forth, the open

Figure 13. Simulation Test Case

source printing is a tough and time-consuming task. Therefore, for most test cases, simulation within the program is run to calculate the time consumption. The test case used is shown in figure 13. For this case, the simulation calculates that PCE algorithm will save 26.57% overall printing time, and the real-world printing test presents a 26.37% time-saving. Therefore, the simulation is reliable. In the printing test, the normal printing method took 455 seconds to finish building for a single layer, while PCE only took 335 seconds.

5.2 Simulation result test case presentation

Before the test case presentation, some definitions should be explained first. "Overall area" (OA) represents the combination of both printing and non-printing area. "Non-printing area" (NA) represents the area not to print, which is also the black area in each test case. "CPP theoretical printing time" (CPPT) represents the simulated shortest printing time based on CPP criteria and optimal path criteria. "Normal algorithm printing time" (NAT) represents the simulated printing time applying normal printing method. "Practical algorithm printing time" (PAT) represents the simulated printing time applying PCE algorithm. With the basic definition, comparison definitions can be introduced. "Blank percentage" (BP) represents the ratio between non-printing area and overall area, which is NA/OA. "Theoretical time-saving percentage"(TTP) represents the theoretically highest time-saving percentage comparing with the normal printing method, which is equal to 1 – CPPT/NAT. "Real time-saving percentage" (RTP) represents the time-saving percentage applying PCE algorithm comparing with the normal printing method, which is 1 – PAT/NAT. The simulation result for all 18 test cases is shown in figure 14.

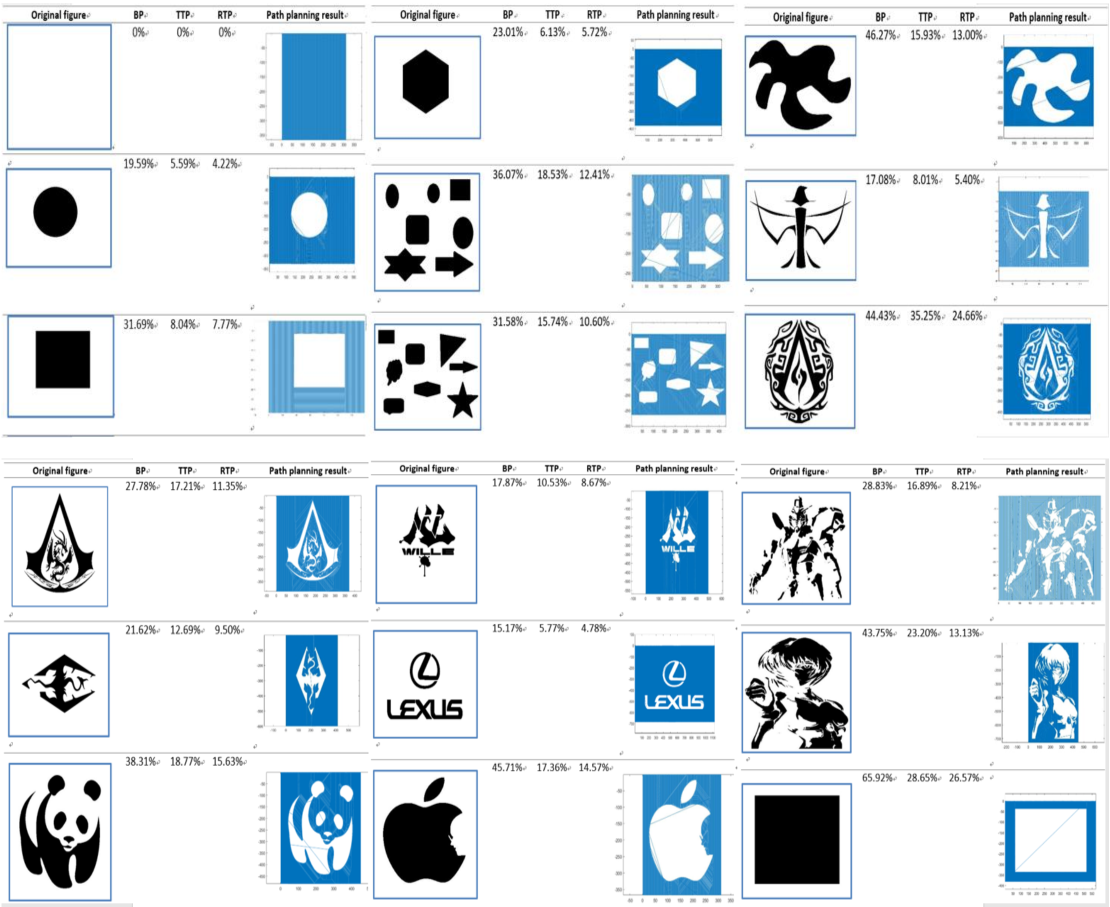

Figure 14. Simulated test cases

5.3 Simulated result analysis

Following the order of Black Percentage, we reorder these 18 results and form a plot in figure 15.

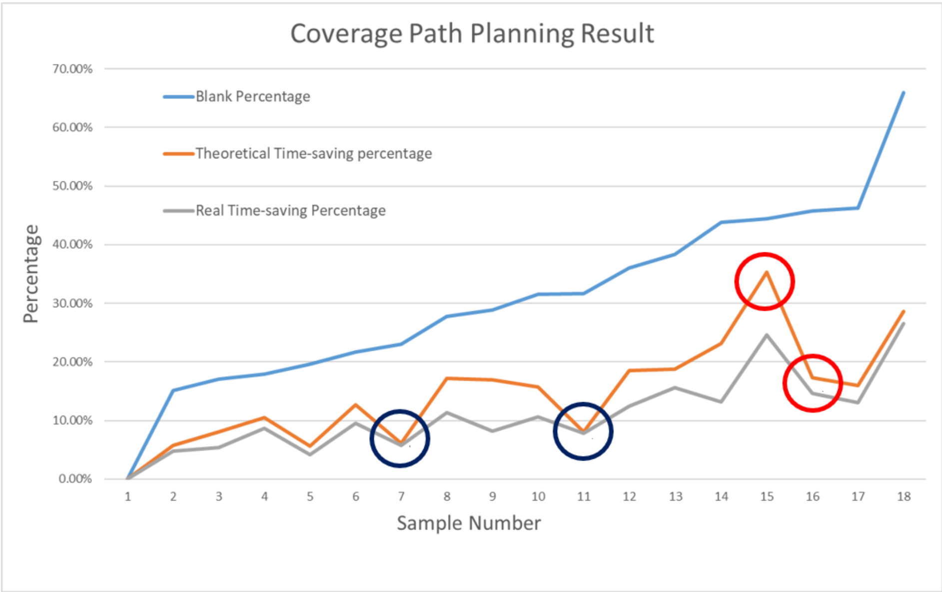

Figure 15. Simulated results plot

From the figure, it’s easy to conclude that both TTP and RTP are following the tendency of BP, which is obvious and the same with coverage path planning theory. What’s more, the relationship between TTP and BP should be discussed. Based on the first conclusion, it is natural to ask the question: when the TTP will not follow the tendency of BP? According to the plot, the best example should be test cased 15 and 16 labeled by red circle, which have similar BP. The two test cases are shown in figure 16. Apparently, using the traditional path planning method to print, if the figure is scanned horizontally, test case 15 will require much more "nozzle preparing time" than test case 16.

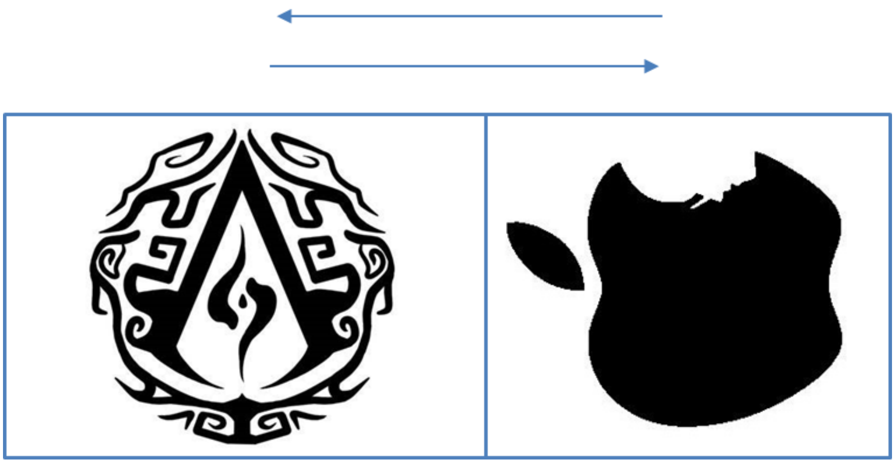

Figure 16. Test case 15(left) and 16(right) result

Finally, the relationship between TTP and RTP is worth discussing in this study. Generally, RTP will be lower than TTP. However, test cases 7 and 11 are exceptions, which are shown in figure 17.

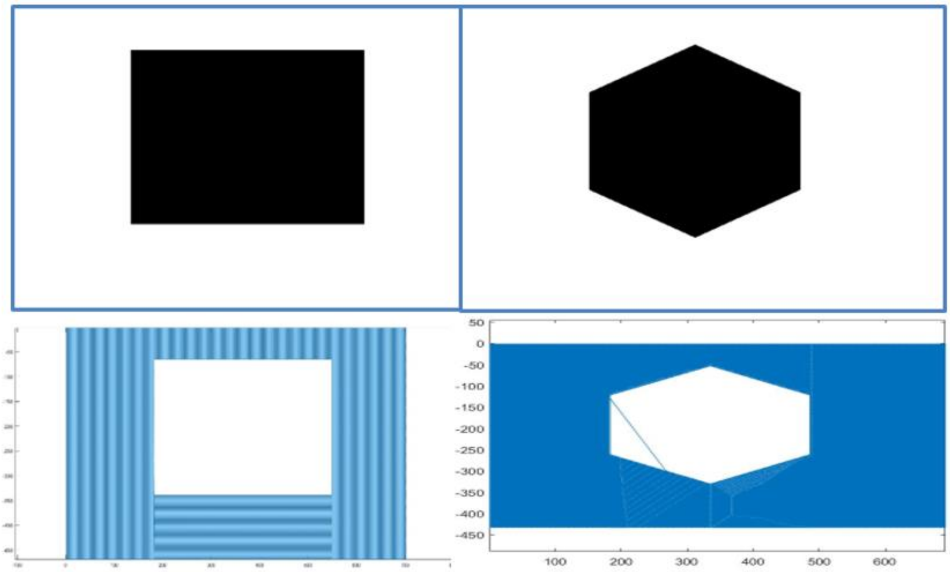

Figure 17. Test case 7(left) and 11(right) results

For test cases 7 and 11, their difference from other cases is that they apparently have less singularities, where all the neighbors are already covered. As mentioned before, when there is singularity in PCE, the nozzle has to stop printing and back trace the next uncovered cell, which forms a line cross the non-printing area and wastes time in nozzle preparing. Therefore, the less singularity a case has, the smaller difference between RTP and TTP will be.

5.4 Boundary and pixel size effect

To illustrate the boundary and pixel size effect, the simulation is done in the case below in figure 18 and 19. From figure 18, conclusion can be made that adding a boundary will sometime lead to loss of the original information. Comparing figure 18 and 19, it’s obvious that increasing pixel size will also lose information, and in this case, adding a boundary can retrieve some information. Therefore, increasing pixel size or adding a boundary when pixel size is one will have bad influence in terms of accuracy of original information representation, while adding a boundary when pixel size is higher can retrieve some information.

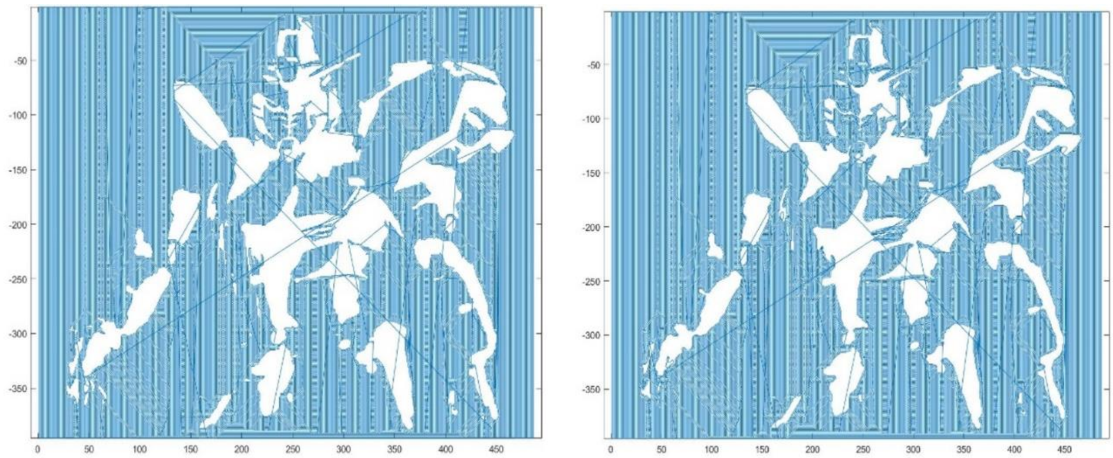

Figure 18. Test case of pixel size one simulation with(right) and without(left) boundary

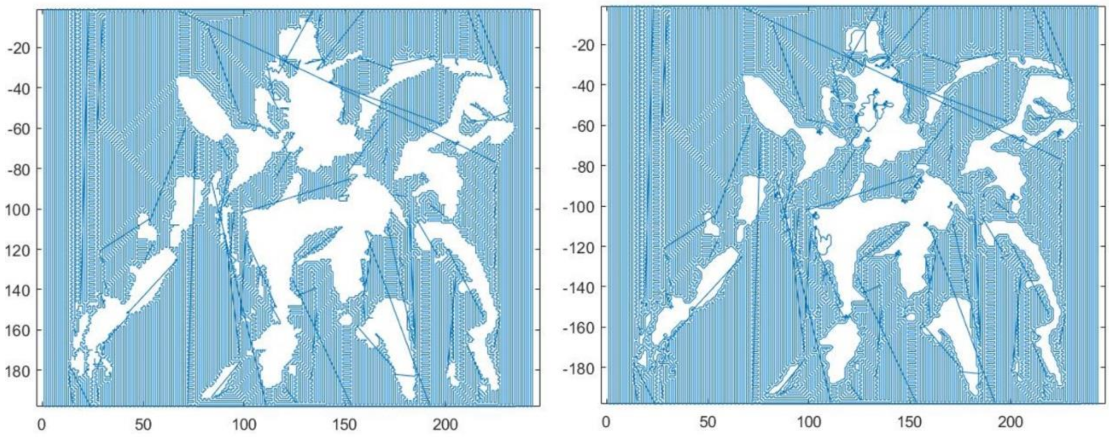

Figure 19. Test case of pixel size two simulation with(right) and without(left) boundary

A sample is printed to test the boundary representation effect of adding a boundary, which is the original reason to add a boundary. The result is shown in figure 20. In this figure, the right one is the sample of pixel size one with boundary, the middle one is the sample of pixel size one without boundary, and the left one is the sample of pixel size two without boundary. The effect of boundary is clearly shown in the samples.

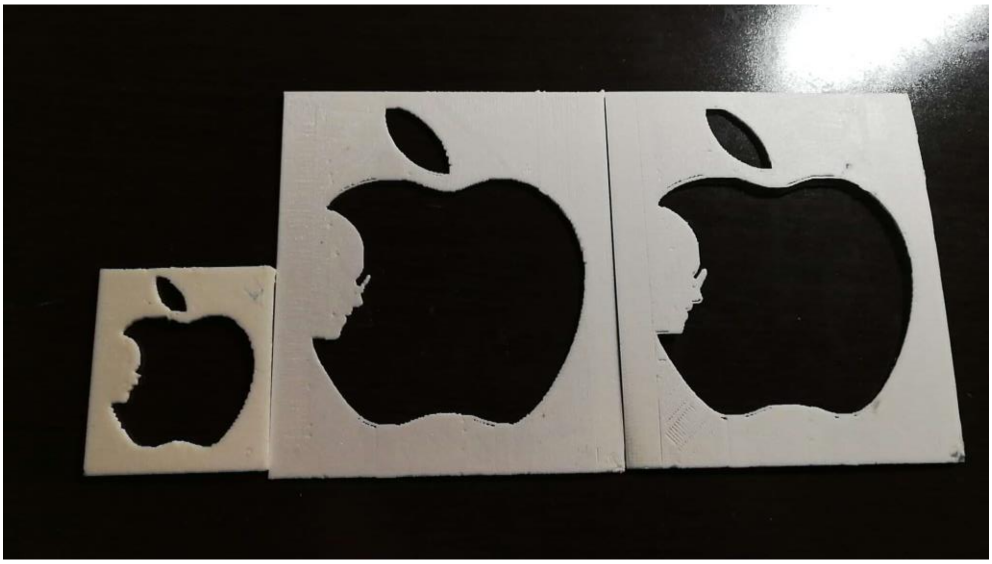

Figure 20. Boundary quality printing test

6. lessons learned

- The CPP algorithm (PCE in this study) is effective, but the time-saving percentage will change with different cases. Therefore, it is not a stable time-saving method in FDM.
- Adding a boundary, at least using boundary algorithm developed in this study, cannot always guarantee a better result. Whether to add it is based on the specific situation of different cases.
- The path planning algorithm itself only considers the time-saving, but not the printing quality. Sometimes the path will affect the printing quality in printing, and this should be considered in programming.
- In this application project, the closest four neighbor cells are defined as neighbors. In fact,

some simulations have been done using eight neighbors. For some cases, singularities are reduced, printing quality is increased, but the simulated printing time is increased. The relationship between 4-neighbor and 8-neighbor algorithm may require more discussion.

7. Future Work

- Combine the path cost evaluation with the exact decomposition path planning methods like Morse-based decomposition.
- Upgrade the algorithm to reduce singularity.
- Try other solutions for locating next point after singularity to reduce the calculation time.
- Do more testing on 8-choice neighbors algorithm and compare it with 4-neighbor one.

Reference

[1] Gross, Bethany C., et al. "Evaluation of 3D printing and its potential impact on biotechnology and the chemical sciences." (2014): 3240-3253.

[2] https://www.tth.com/3d-printing/fdm-prototyping/.

[3] http://www.instructables.com/id/3D-Printing-101-1-Introduction/.

[4]Galceran, Enric, and Marc Carreras. "A survey on coverage path planning for robotics." Robotics and Autonomous systems 61.12 (2013): 1258-1276.

[5] S. Gajjar, J. Bhadani, Complete coverage path planning algorithm for known 2d environment, 2017, IEEE