A 3D Printing Path Optimizer Based On Christofides Algorithm

Kai-Yin Fok[†], Nuwan Ganganath, Chi-Tsun Cheng, and Chi K. Tse
Department of Electronic and Information Engineering
The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong
Email: [†]zerofky@gmail.com

***Abstract*—Rapid prototyping and product customization have become more convenient with the emergence of 3D printing technologies. In extrusion deposition based 3D printing, objects are built by connecting many lines of filament, layer by layer. The efficiency of the printing process can be improved by optimizing motion paths of the printing nozzle. In this paper, a 3D printing path optimizer based on Christofides algorithm is proposed. Experiment results show that the proposed optimizer can significantly reduce the length of motion paths compared to a nearest neighbor-based optimizer using in consumer 3D printers.**

***Index Terms*—3D printers, path planning, motion control, additive manufacturing, traveling salesman problem**

I. INTRODUCTION

Thanks to recent advances in 3D printing technologies, 3D printers can now be manufactured at affordable prices and be compact enough to be treated as ordinary office equipment. A 3D model of an object to be printed is usually fed into a 3D printer as G-code [1], a numerical control programming language for most off-the-shelf 3D printers. It carries instructions for controlling the mechanical movements of a printer, including motions of its printing nozzle, print bed positions, filament feeding speeds, etc.

This paper proposes a path optimizer for the printing nozzle, which can ultimately help in reducing the printing time. As pointed out in [2], a print segment in 3D printing can be considered as an edge between two nodes. The objective of the optimizer is to find a tour that can visit all the print segments. This circumstance is similar to the well-known travelling salesman problem (TSP) which aims to search a shortest tour that travels every node exactly once and return to the starting node. The major difference is that the former focuses on connecting exiting edges while the latter focuses on joining nodes. TSP is proved to be MAX SNP-hard [3]. A polynomial time algorithm proposed by Christofides has an approximation factor of 1.5 in solving TSP [4]. Simulations in [2] have shown that modified TSP algorithms are capable to find fast trajectories for the printing nozzle.

In this paper, the 3D printing path optimization problem is formulated similar to that as in [2] and solved using a variant of Christofides algorithm. The optimizer is implemented and tested using Cura-15.04.03 [5], a common open-source software for converting 3D models in STereoLithography (STL) [6] file format into G-code. The simulations reported in [2] only consider printing of randomly-generated non-intersecting straight print segments. Furthermore, it only considers print segments on a single layer. However, real-world 3D printing tasks can be far more complicated. Hence, this paper evaluates the performance of the proposed optimizer using real 3D printing models which comprises multiple layers. Results show that the proposed optimizer can significantly reduce the tour length when comparing with the built-in nearest neighbor-based optimizer in Cura,

The rest of the paper is structured as follows. The build-in nearest neighbor-based optimizer and the proposed optimizer are described in Section II. Experiment setup is introduced in Section III followed by results and analyses. Concluding remarks are given in Section IV.

II. 3D PRINTING PATH OPTIMIZERS

The 3D printing path optimizing algorithms under test are briefly discussed in this section.

1) Nearest neighbor-based algorithm: Cura uses a nearest neighbor-based algorithm in its built-in path optimizer [7]. It is a heuristic algorithm which has a low computational complexity. However, on each layer, it does not globally consider all print and transition segments in its optimization process. Instead it first traverses all print segments inside each polygon, then moves the nozzle to the nearest polygon. The process repeats until all polygons on same layer are printed.

2) Modified Christofides algorithm: Christofides algorithm used in the proposed path optimizer begins with constructing a minimum spanning tree (MST) using a Krushal algorithm [8], that operates on a set of print segments. The ending node of the previous layer is treated as the starting node of the current layer. This starting node is connected to itself to form a virtual segment. After that, a minimum cost perfect matching on odd degree nodes is performed. An Eularian circuit can be obtained by combining the MST and the matching graph. The virtual segment that formed earlier is then disconnected. The final path is obtained by traversing from the staring node while skipping nodes that have been visited before.

III. EXPERIMENTS

A. Experiment Setup

A total of seven 3D models [9] are selected for evaluating the proposed optimizer. Some models are scaled down in order

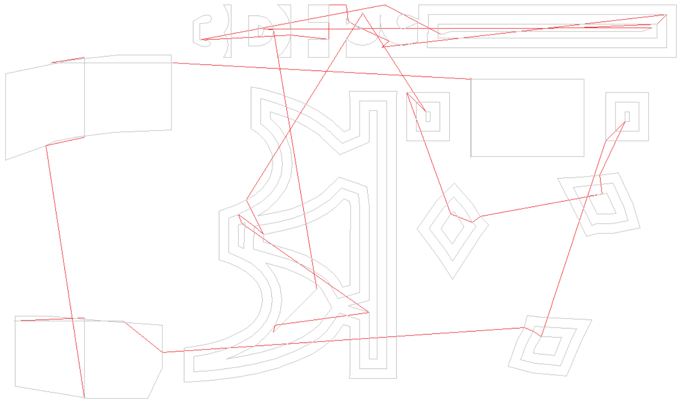

Fig. 1. An illustration of print (red color) and transition (grey color) segments on a single layer when paths are obtained using the built-in path optimizer in Cura.

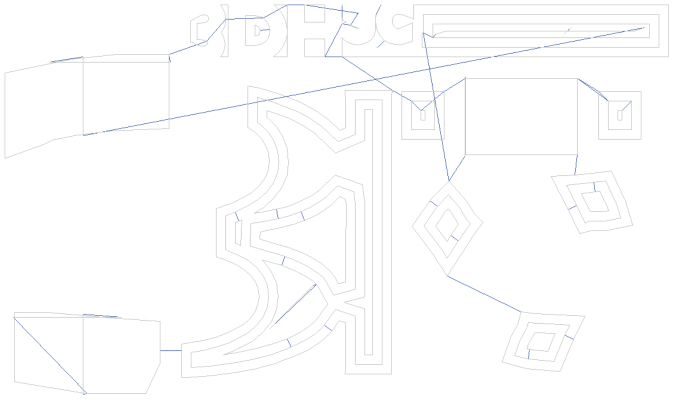

Fig. 2. An illustration of print (blue color) and transition (grey color) segments on a single layer when paths are obtained using the proposed path optimizer.

to fit in a print bed with a dimension of 250×250 mm^2. The proposed method is implemented and integrated into Cura as an alternative print segment optimizer. G-code files generated using the built-in and proposed optimizers are then evaluated using GCode Print Simulator-1.32 [10].

B. Experiment Results and Analyses

Figs. 1 and 2 illustrate the motion path of the printing nozzle on 50th layer of the model "3DHackerTest.stl" [9]. It can be observed that the total distance of the blue segments is much shorter than the red segments. Similar results are observed in other layers of the 3D models under test.

According to the results given in Table I, the proposed optimizer is able to generate shorter tours for all the models under test. On average, paths generated using the proposed optimizer are 8.58% shorter than paths generated using the built-in algorithm. Since the total printing time is based on the total path length and the velocity of the nozzle, it implies a shorter printing time. Comparing with the built-in optimizer, a slight increase in computation time is observed when the proposed optimizer is utilized. However, when comparing with the printing time being saved, such an increase is insignificant.

TABLE I
EXPERIMENT RESULTS

3D Models	Total Path Lengths (mm)	
	Built-in Algorithm	Christofides Algorithm
3DHackerTest	112548.03	103534.68
ctrlV_3D_test	135552.84	127231.76
Debailey_x10	161282.28	145309.73
dragon_65_tilted_large	118466.03	106417.77
testModel	183172.72	170849.90
TortureTestV2	177335.86	160966.66
UltimakerRobot_support_2015	143080.47	128966.29

IV. CONCLUSION

This work proposes essential modifications to Christofides algorithm for optimizing motion paths in 3D printing and the proposed optimizer is evaluated using actual 3D models. Results show that the proposed method can significantly reduce the length of motion paths comparing to a path optimizer used in a common 3D model slicing software. Printing time can be greatly reduced by adopting the proposed optimizer and that can be further optimized by considering the velocities and accelerations of the printing nozzle.

ACKNOWLEDGMENT

This work is supported by the Department of Electronic and Information Engineering, the Hong Kong Polytechnic University (Projects RU9D, RTKL and G-YBKH) and the Hong Kong PhD Fellowship Scheme.

REFERENCES

[1] "G-code - RepRapWiki," (Accessed: 2016-01-2). [Online]. Available: http://reprap.org/wiki/G-code

[2] N. Ganganath, C.-T. Cheng, K.-Y. Fok, and C. K. Tse, "Trajectory planning for 3D printing: a revisit to traveling salesman problem," in *2016 International Conference on Control, Automation and Robotics (ICCAR)*, Apr 2016.

[3] C. H. Papadimitriou and M. Yannakakis, "The traveling salesman problem with distances one and two," *Mathematics of Operations Research*, vol. 18, no. 1, pp. 1–11, 1993.

[4] N. Christofides, "Worst-case analysis of a new heuristic for the travelling salesman problem," DTIC Document, Tech. Rep., 1976.

[5] Ultimaker, "Cura 3D Printing Slicing Software," (Accessed: 2016-01-2). [Online]. Available: https://ultimaker.com/en/products/cura-software

[6] D. Pham and R. Gault, "A comparison of rapid prototyping technologies," *International Journal of Machine Tools and Manufacture*, vol. 38, no. 10, pp. 1257–1287, 1998.

[7] T. Kuipers, "CuraEngine/pathOrderOptimizer.cpp at master Ultimaker/CuraEngine GitHub," (Accessed: 2016-01-4). [Online]. Available: https://github.com/Ultimaker/CuraEngine/blob/master/src/pathOrderOptimizer.cpp

[8] J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proceedings of the American Mathematical society*, vol. 7, no. 1, pp. 48–50, 1956.

[9] T. Kuipers, "CuraEngine/tests at master Ultimaker/CuraEngine GitHub," (Accessed: 2016-01-3). [Online]. Available: https://github.com/Ultimaker/CuraEngine/tree/master/tests

[10] M. Dietz, "GCode Print Simulator by mdietz - Thingiverse," (Accessed: 2016-01-7). [Online]. Available: http://www.thingiverse.com/thing:44286