Trajectory Planning for 3D Printing: A Revisit to Traveling Salesman Problem

Nuwan Ganganath*, Chi-Tsun Cheng, Kai-Yin Fok, and Chi K. Tse
Department of Electronic and Information Engineering
The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong
e-mail: *nuwan@ganganath.lk

***Abstract*—Three dimensional (3D) printing can be used to manufacture many different objects range from toys to hi-tech robot parts. This paper investigates 3D printer trajectory planning to improve the speed of the printing process. The printing speed mainly depends on the motion speed and path of the printing nozzle. We use triangular and trapezoidal velocity profiles to minimize the transition time between print segments. In this work, several algorithms that were originally proposed as solutions for conventional traveling salesman problem are modified to adapt to the new problem. The proposed modifications are designed to obtain time-efficient trajectories for the printing nozzle.**

Keywords-3D printers, additive manufacturing, trajectory generation, path planning, motion control.

I. Introduction

Three dimensional (3D) printing is an additive manufacturing technique which uses many thin layers of materials to build 3D objects. It has sparked a revolution in the manufacturing industry by enabling users to create complex-shaped objects with both hollow and solid structures [1]. Some benefits of 3D printing over conventional manufacturing techniques are low manufacturing cost, ease of designing and customizing objects, automated manufacturing, and minimal wasted materials. As a result, it has been commonly used to manufacture customized products, prototypes, and replacement parts for many different applications [2].

A typical 3D printer consists of two essential components: a movable *printing nozzle* and a movable *print bed* (see Figure 1). We assume that the printing nozzle moves in horizontal plane (x-y plane) and the print bed moves only along vertical axis (z axis). Once a design is fed into the printer, the nozzle moves relative to the print bed to construct the design layer by layer. Each layer is made of multiple *print segments*, the thickness of which depends on the amount of materials deposited. The flow rate of printing materials and the motion speed of the nozzle can be considered as the main factors which control the amount of materials deposited on a unit length of a print segment. The printing duration of a given design depends on both the motion speed and the path of the printing nozzle.

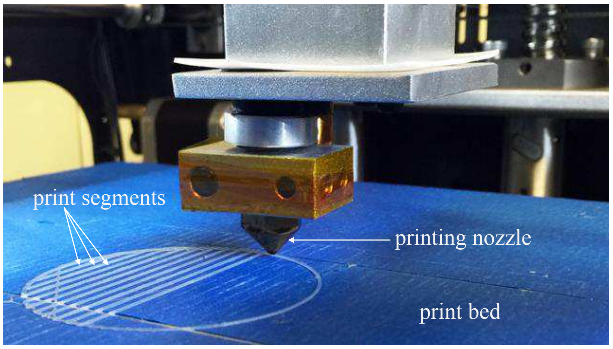

Figure 1. A printing nozzle and a print bed of a 3D printer.

This paper focuses on the *trajectory planning* for rapid 3D printing. In order to minimize the printing duration, trajectory planning algorithms used in 3D printing systems need to control both the position and speed of the nozzle efficiently. Since the problem is application specific, so far only few attempts have been reported in literature. Thompson and Yoon first proposed a trajectory planning algorithm based on two motion control methods: linear segments with parabolic blends and minimum time trajectory [3]. Recently, an improved algorithm was proposed by the same authors that can limit the speed fluctuations by predicting velocity errors beforehand [4]. These algorithms can only determine the desired motion speed of the nozzle and they do not optimize motion paths.

In order to minimize the printing duration, we propose to optimize both the motion speed and path of the nozzle. Section II describes a motion control model to calculate the nozzle speed at different segments of a path. In order to determine a motion path, we modify several heuristic algorithms which were originally proposed as solutions to *traveling salesman problem* (TSP) [5]. The proposed modifications to those algorithms are described in Section III. The modified algorithms are tested with the motion control model using extensive computer simulations. Simulations results are presented and analyzed in Section IV. Some concluding remarks and possible future research directions are highlighted in Section V.

II. Motion Control Model

Here, we consider a trajectory $q(t)$ which is a path that a moving nozzle follows through x-y plane as a function of time t. As shown in Figure 2(a), such a trajectory consists of print segments and transition path segments. The motion control model utilized in this work is based on two basic principles: On print segments, the nozzle should move at

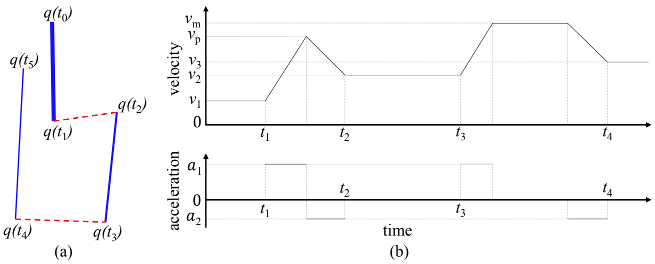

Figure 2. (a) A trajectory of a nozzle. Print segments and transition path segments are represented by solid and dashed lines, respectively. (b) Velocity and acceleration profiles of the trajectory.

a constant speed to facilitate consistent material deposition. On the transition path segments, the nozzle should move as fast as possible to minimize the transition time. Therefore, triangular and trapezoidal velocity profiles are utilized to model the transition motion [3]. Figure 2(b) illustrates corresponding velocity and acceleration profiles for the trajectory given in Figure 2(a).

In the given example, velocities of the nozzle while it is moving on the three print segments are set to v_1, v_2, and v_3 accordingly. Let the length of the second print path segment be denoted as p_2. The printing duration of the second print segment can be obtained as

$$t_3 - t_2 = \frac{p_2}{v_2}.$$

Similarly, the printing duration of other print segments can also be obtained. Since the length of the print segments and the corresponding speed of the nozzle are constants, the actual printing time cannot be improved. Hence, the printing process can only be accelerated by minimizing the transition time.

It is assumed that the nozzle follows a triangular velocity profile during the time period $[t_1, t_2]$ and reach a peak velocity of v_p. It is also assumed that the acceleration and deceleration values of the nozzle are set to a_1 and a_2, respectively. Hence, the transition time can be obtained as

$$t_2 - t_1 = \frac{v_p - v_1}{a_1} + \frac{v_2 - v_p}{a_2}. \tag{1}$$

Let the length of the first transition path segment be denoted as d_1. Hence, we have

$$d_1 = \frac{v_p^2 - v_1^2}{2a_1} + \frac{v_2^2 - v_p^2}{2a_2}.$$

Since v_1, v_2, a_1, a_2, and d_1 are known for a given path, v_p can be obtained as

$$v_p = \pm\sqrt{\frac{2a_1a_2d_1 + a_2v_1^2 - a_1v_2^2}{a_2 - a_1}}. \tag{2}$$

Note that $a_1 \neq a_2$. The transition time $t_2 - t_1$ given in (1) can be minimized by maximizing v_p, thus by maximizing acceleration and/or deceleration values [3].

According to (2), $|v_p|$ increases with the transition distance. If $|v_p|$ exceeds a maximum allowed speed $|v_m|$, such a transition motion is modeled using a trapezoidal velocity profile as illustrated in Figure 2(b) during the time period $[t_3, t_4]$. Let the length of the second transition path segment be denoted by d_2. Hence, we have

$$d_2 = v_m\left[(t_4 - t_3) - \frac{v_m - v_2}{a_1} - \frac{v_3 - v_m}{a_2}\right] + \frac{v_m^2 - v_2^2}{2a_1} + \frac{v_3^2 - v_m^2}{2a_2}. \tag{3}$$

Using (3), the second transition time can be obtained as

$$t_4 - t_3 = \frac{d_2}{v_m} + \frac{v_m - v_2}{a_1} + \frac{v_3 - v_m}{a_2} - \frac{v_m^2 - v_2^2}{2v_ma_1} - \frac{v_3^2 - v_m^2}{2v_ma_2}.$$

Similar to the previous case, the transition time can be minimized by maximizing acceleration and/or deceleration values.

III. Path Planning Algorithms

This section focuses on a problem of finding a fast path from a predefined start point to a predefined end point such that it travels through all print segments. This problem is closely related to TSP which asks to find a shortest path that travels through each city exactly once and returns to the origin [5]. If we consider a print segment as an edge which connects two nodes in a given graph, the path planning problem considered in this work differs from TSP in three ways: First, it is a problem of connecting existing edges, instead of nodes. Second, it does not require a path to return to the start node (origin). Finally, its objective is to minimize the total traversal time, instead of path length. Thus, the edge costs are specified in terms time durations using the motion control model explained in the preceding section. Here we modify several TSP algorithms for the 3D printer path planning problem as explained below.

A. Random Selection

The path planning process initiates from a given start node and randomly selects a node from a set of nodes which excludes the end node. Once a node is selected, another node which is connected to the selected node through a print segment is also selected. Then, both nodes are removed from the set and a new node is randomly selected from the remaining set. This process continues until the set becomes empty. Finally, the end node is selected and the path is constructed from the start node to end node follow the order of selection. Here, we use results generated by random selection as a reference to evaluate the other algorithms.

B. Nearest Neighbor Selection

The modified nearest neighbor path planning process is similar to the path planning process explained under random selection, except it selects the nearest node to the current node as the next node of the path instead of randomly selecting a node.

C. Christofides Algorithm

In TSP, Christofides algorithm [6] begins with creating a minimum spanning tree (MST) from isolated nodes of a given graph. Instead, here it obtains an MST starting from a forest which consist of pairs of nodes that are connected with print segments and the start and end nodes that are connected using a virtual edge. Once the MST is obtained using Kruskal's algorithm [7], it performs a minimum weight matching for the set of odd degree nodes in the MST. Then the MST is combined with the matching graph to generate another graph that consists of only even degree nodes. After that, it finds an Eularian circuit on the combined graph and the virtual edge between start and end nodes is disconnected. If the resulted Eularian path consists of two consecutive transition edges (path segments), the common node to those two edges is removed and a shortcut between the two remaining nodes is created. The final path is obtained by visiting the resulting node sequence from the start node to the end node while skipping already visited nodes if there exist any repeating nodes.

D. k-opt Heuristics

k-opt heuristics are proposed as a local improvement technique for TSP solutions [5]. For a given feasible tour, it deletes k mutually disjoint edges and reconnects the remaining fragments such that resulting tour is shorter than the original tour. This process iteratively looks for all possible combinations until no further improvements can be made. The running time of k-opt technique increases considerably with k. Thus, we only consider 2-opt and 3-opt techniques (*i.e.* $k = 2$ and 3). In 3D printer path planning, it can only swap the transition path segments as the print segments cannot be changed.

IV. Simulations

A. Simulation Parameters

The print bed is assumed to be a square with an area of 100×100 mm^2. The desired velocity for printing each print segment is set to 3 mm/s. The maximum allowed velocity of the nozzle is 4 mm/s. It is assumed that the nozzle move at its maximum acceleration and deceleration while following triangular and trapezoidal velocity profiles. The absolute value of the maximum acceleration/deceleration is set to 30 mm/s^2. In each simulation, print segments are distributed uniformly at random on the print bed provided that they do not intersect with each other. The length of the print segments are selected uniformly at random from a range of $[10, 40]$ mm. Start and end points are also randomly selected within the print bed.

B. Results and Discussion

The first set of simulations were performed with 6 print segments. Simulation results are illustrated in Figure 3. The trajectory generated using random selection requires the longest time (110.09 s) to complete the printing process. The trajectories found by Nearest neighbor and Christofides algorithms report significantly improved operation times of 94.49 s and 85.04 s, respectively. The second row in Figure 3 shows improved trajectories obtained using 2-opt swap. 2-opt swap has improved the previously solutions by exploiting poor sections of the paths, such as path crossings.

To further justify these results, more simulations were performed using different number of print segments. Statistical results of 100 individual realizations are provided in Table I. According to the simulation results, Christofides algorithm outperforms random and nearest neighbor selections in terms of both the path length and the operation time, regardless of the number of print segments. For 100 print segments, the trajectories obtained by nearest neighbor selection and Christofides algorithm can shorten the printing process by more than two times than the trajectories obtained by random selection. Nevertheless, 2-opt and 3-opt techniques have improved the quality of the solutions obtained by all the algorithms under test. For 20 and 100 print segments, random selection followed by 2-opt swap has achieved faster completion time compared to nearest neighbor selection and Christofides algorithm without refinements. Since 3D printing models often consists of large number of print segments within a single layer, 2-opt and 3-opt techniques can effectively shorten the printing process.

V. Conclusion and Future Work

3D printing speed can be improved by optimizing the motion path of the nozzle and selecting effective velocity profiles for each segment of the path. A motion path usually consists of print segments and transition path segments. This work utilizes a motion control model which can ensure uniform material deposition on print segments and short transition time on the rest of the path. Some previously proposed TSP algorithms are modified to find fast trajectories for the nozzle. Further experiments using 3D printers need to be carried out to verify the applicability of the proposed trajectory planning techniques. This work only consider trajectory planning over a single layer of a 3D object. Therefore, future research should also focus on optimizing trajectory of the nozzle across multiple layers.

Acknowledgment

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

References

[1] H. Lipson and M. Kurman, *Fabricated: The new world of 3D printing*. John Wiley & Sons, 2013.

[2] B. Berman, "3-D printing: The new industrial revolution," *Business horizons*, vol. 55, no. 2, pp. 155–162, 2012.

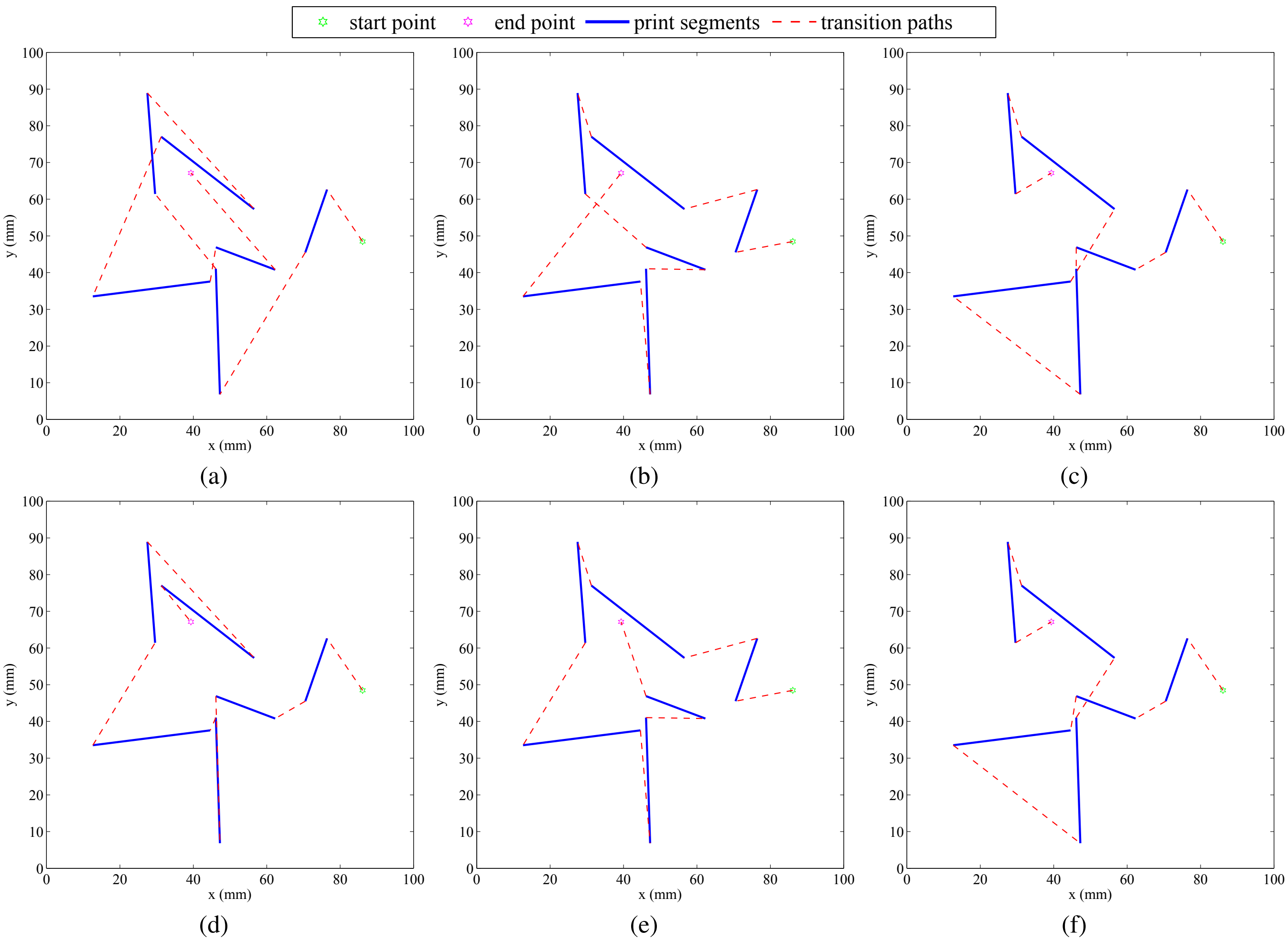

Figure 3. Path planning of 6 randomly distributed print segments using (a) random selection, (b) nearest neighbor selection, (c) Christofides algorithm, (d) random selection followed by 2-opt swap, (e) nearest neighbor selection followed by 2-opt swap, and (f) Christofides algorithm followed by 2-opt swap.

Table I
SIMULATION RESULTS.

Path planning algorithms	Average path length (mm)			Average operation time (s)		
	$N = 5$	$N = 20$	$N = 100$	$N = 5$	$N = 20$	$N = 100$
Random	344.4421	1279.7178	6324.6440	96.4203	358.2532	1737.7589
Random followed by 2-opt	267.9284	713.3363	2359.5340	77.2918	216.6573	746.4605
Random followed by 3-opt	266.7053	705.9704	2348.4699	76.9860	214.8158	743.6945
Nearest neighbor	265.8711	742.567	2471.0373	76.7775	223.9646	774.3394
Nearest neighbor followed by 2-opt	249.9634	687.1246	2321.8534	72.8006	210.1038	737.0403
Nearest neighbor followed by 3-opt	249.5605	684.6447	2316.7929	72.6998	209.4839	735.7752
Christofides algorithm	264.4541	719.4376	2417.0162	76.4233	218.1821	760.8402
Christofides algorithm followed by 2-opt	248.7739	679.4121	2301.9392	72.5032	208.1757	732.0613
Christofides algorithm followed by 3-opt	248.6097	677.3959	2298.3651	72.4621	207.6717	731.1678

N = number of print segments.

[3] B. Thompson and H.-S. Yoon, “Efficient path planning algorithm for additive manufacturing systems,” *Components, Packaging and Manufacturing Technology, IEEE Transactions on*, vol. 4, no. 9, pp. 1555–1563, 2014.

[4] B. Thompson and H.-S. Yoon, “Velocity-regulated path planning algorithm for aerosol printing systems,” *Journal of Manufacturing Science and Engineering*, vol. 137, no. 3, pp. 031 020–031 020–7, 2015.

[5] D. S. Johnson and L. A. McGeoch, “The traveling salesman problem: A case study in local optimization,” *Local search in combinatorial optimization*, vol. 1, pp. 215–310, 1997.

[6] N. Christofides, “Worst-case analysis of a new heuristic for the travelling salesman problem,” Carnegie-Mellon University, Pittsburgh, PA, Tech. Rep. Management Sciences Research Report No. 388, Feb. 1976.

[7] 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.