Printing Direction Optimization Through Slice Number and Support Minimization

WEIMING WANG, HANLIU SHAO, XIUPING LIU, AND BAOCAI YIN
School of Mathematical Sciences, Dalian University of Technology, Dalian 116024, China
Corresponding author: Weiming Wang (wwmdlut@gmail.com)

This work was supported in part by the Natural Science Foundation of China under Grant 61702079 and U1811463, and in part by the Innovation Foundation of Science and Technology of Dalian under Grant 2018J11CY010.

ABSTRACT One major drawback of 3D printing technologies is the low printing efficiency. In general, it takes 8-12 hours to print a normal 3D model. 3D printing technologies manufacture objects layer by layer where each layer is composed of one or more closed 2D polygons (named as slices in this work). As a result, the number of slices will directly affect the printing time. In addition, most 3D printing technologies need extra supporting structures to support overhang regions during printing, such as Fused Deposition Modeling (FDM), which will largely increase manufacturing time. We observe that both the slicing number and the amount of supporting structures are affected by printing direction. In this work, a novel printing direction optimization algorithm is proposed based on the amount of slices and overhang areas. In the proposed method, Genetic Algorithm is used to obtain optimal printing direction where the fitness function is designed as the weighted slice number and overhang areas. Experimental results show that the proposed algorithm is able to obtain an optimal printing direction to reduce the number of slices and overhang areas.

INDEX TERMS 3D printing, slicing optimization, printing efficiency, genetic algorithm, supporting structure.

I. INTRODUCTION
3D printing (also named as additive manufacturing) is now widely used in small-scale prototype design and innovation projects due to its flexibility and low-cost. This new manufacturing technology largely changes the traditional subtractive manufacturing, and brings the change of manufacturing processes and production models. There are several 3D printing technologies, such as Fused Deposition Modeling (FDM), Selective Laser Sintering (SLS), Stereolithography (SLA), Digital Light Processing (DLP), etc. These technologies can easily manufacture arbitrary complex 3D models (such as free-form surfaces, complex internal runners and internal hollow structures) that cannot be produced by the traditional manufacturing technologies, for instance Stamping, which largely reduce the manufacturing cost of complex models and structures. Although 3D printing technologies have a plenty of advantages, they still have several defects that should be improved. First of all, the prices of many printing materials are still high. Secondly, the accuracy, structural strength, and chemical properties of the objects manufactured with 3D printing technologies cannot meet the practical application requirements. Thirdly, 3D printing technologies always need a long time to produce even a small project, which largely affects their application fields. For the current shortcomings of 3D printing technologies, researchers have carried out a lot of explorations. Among these, printing direction optimization is one of the important and hot research topics. As we know, printing direction affects many aspects in 3D printing, such as the amount of supporting structures, printing quality, structural strength, printing time, and so on. Many algorithms have been proposed to obtain optimal printing directions by considering overhang areas, structural strength, surface quality, etc. But none of them take into account the number of slices when optimizing the printing direction.

Printing time is directly affected by many factors, such as printing speed, printing paths, and the number of layers, etc. Printing speed is always specified according to different printers and materials. Printing path has been researched for a long time and there are existing a lot of effective algorithms. Some algorithms have been successfully integrated into 3D printing software. Therefore, they are out of the scope.
of this work. Through a large number of practical printing experiments, we observe that the number of slices has a large influence on the printing time when consuming the same amount of material which will be discussed in Sec. (III-A).

For a given 3D model, different printing directions may result in very different number of slices, see Figure. 1. The Desk model is sliced along z-axis in two different printing directions. The number of slices generated in these two directions are listed at the bottom which are 431 (left) and 46 (right), respectively. From the leg regions we can see that one layer of bottom-left has 4 independent slices, while the bottom-right has only 1 independent slice. As a result, bottom-left has larger number of slices than bottom-right. As analyzed above, the number of slices is determined by printing direction, which directly affects the printing time of the whole model. Therefore, one goal of the proposed method is to reduce the number of slices. Furthermore, for most 3D printing technologies, the amount of overhang areas directly affects the amount of supporting structures which directly affects the printing time and printing quality. Therefore, the other goal of the proposed method is to reduce the amount of overhang areas. Since the proposed optimization formulation is highly non-linear, Genetic Algorithm [1] is used to solve it where weighted the number of slices and overhang areas is used as fitness function.

The rest of this paper is arranged as follows: several related works will be introduced in Sec. (II), the details of the proposed algorithm will be described in Sec. (III), the implementation details of our algorithm with Genetic Algorithm will be given in Sec. (IV), several experimental results will be shown in Sec. (V), and the whole algorithm will be concluded in Sec. (VI).

II. RELATED WORK

3D printing is currently a hot research topic in many fields, such as Computer Graphics and Rapid Prototyping. So far, a plenty of algorithms have been proposed to handle different issues for 3D printing technologies. In this work, we mainly focus on the algorithms related to printing time and printing direction. The interested readers are recommended to read [2]–[6] for more algorithms.

A. PRINTING DIRECTION OPTIMIZATION

A lot of researchers propose algorithms to generate the optimal printing directions by considering certain goals, such as the structural strength, surface quality, and visual quality, etc. Umetani and Schmidt. [7] propose a novel cross sectional structural analysis technique to effectively detect the critical stress of a given model, then the analysis results are used as guidance to generate an optimal printing direction for structural strength. Wang et al. [8] propose an algorithm to improve the local printing quality of the object by decomposing a given model into parts with the consideration of building direction. Supporting structures always affect the fine details of the objects when removed. To prevent the visual impact of these artifacts, Zhang et al. [9] present a method to find printing directions that avoid placing supports in perceptually significant regions. Masood et al. [10] try to obtain an optimal building orientation by the consideration of the volumetric error.

The other researchers propose systems to find the optimal printing directions by combining several different aspects, such as build time, accuracy, stability, surface finish, support structure, etc. Byun and Lee [11], [12] aim to determine the optimal build-up direction when a part is built with the variable layer thickness for different PR system. The surface roughness, printing time, and the part cost using the variable layer thickness are considered in their works. Danjou and Köhler [13] present a generic system that performs a computer-aided optimization of part orientation by considering several factors. This method is possible to analyze convex as well as non-convex parts. Frank and Fadel [14] propose an expert system tool that considers the various parameters that affect the production of the prototype and interacts with the users, such as stair stepping effect, build time, and surface finish, to recommend the best printing direction. Xu et al. [15] take into consideration building time, accuracy, and stability of the part when determining the optimal orientation. Their method is proposed by the combination of variable thickness slicing techniques. Thrimurthulu et al. [16] attempt to obtain an optimal part deposition orientation for fused deposition modeling process for enhancing part surface finish and reducing building time. An evaluation of average part surface roughness and building time is presented, and the optimal solution is obtained by a real coded genetic algorithm. Son and Choi [17] search for the best building orientation for PR system by considering the material usage, resolution error, and support remainder visibility. The proposed algorithm is able to produce cost-effective and visually pleasing printing results. Since layer by layer building process introducing an error in the amount of material used compared to the volume specified by the computer-aided design model. Pandey et al. [18] describe and compare the various attempts made to determine part deposition orientation, and build time, support structure, dimensional accuracy, surface finish and
cost of the prototype are considered in their work. In this work, the number of slices and overhang area are considered to find an optimal printing direction.

B. BUILD TIME

1) PATH PLANNING

In 3D printing, sharp corners and the non-uniform spacing between adjacent path elements in the final planned path are two major causes of unevenness of the deposited surface. Jin et al. [19] propose an algorithm to decrease the number of sharp corners by using an implicit algorithm derived from the level sets of the input contours. Zhao et al. [20] develop a new kind of space-fill curves, connected Fermat spirals. They demonstrate that printing 2D layers following tool paths as connected Fermat spirals leads to efficient and quality fabrication, compared to conventional fill patterns. Dreifus et al. [21] propose a method for programming minimal time tool paths for single-bead-wide extrusion in additive manufacturing along an arbitrary lattice. They present a graphical model for 3D printing process and the Chinese Postman Problem (CPP) is used to obtain the solution on a given mesh. 3D printer with multiple extruders can effectively improve the printing efficiency. However, there is no efficient algorithm to schedule the extruders to avoid collisions during the printing operations. Wang et al. [22] propose a slicing algorithm which divides each slicing piece into conflicting areas and non-conflicting areas. The non-conflicting areas can be simultaneously printed with multiple extruders, while the conflicting areas must be printed separated with only one extruder.

2) ADAPTIVE SLICING

Printing time is directly affected by the number of slices. To reduce printing time, the slicing thickness should be increased which may result in poor printing quality. To solve this problem, many researchers propose algorithms to slice 3D models with adaptive thickness according to geometric or visual errors. Mani et al. [24] describe a method for region-base adaptive slicing. In the presented algorithm, user has the flexibility to impose different surface finish requirements on different surfaces of the model. Pandey et al. [25] present an adaptive slicing algorithm for Fused Deposition Modeling based on real time edge profile of deposited layers rather than the traditional rectangular based edge profiles. Hayasi and Asabianpour [26] propose a new adaptive slicing method whereby the capability of cutting a 3D solid model at the predefined sheets. Hildebrand et al. [27] propose an algorithm to first partition the given model into parts with orthogonal bases, and then slicing each part with one of the three orthogonal directions. Wang et al. [28] observe that human beings are more sensitive to geometric errors in the region with high visual saliency. Therefore, they propose an adaptive slicing algorithm base a visual saliency based metric. In their work, the slicing thickness and positions are optimized to minimize the manufacturing time under a given build direction. Alexa et al. [29] propose an algorithm to generate printing sequences with minimal geometric error for any possible number of slices.

However, the total number of slices is not considered in the above algorithms. In this work, the build direction is optimized through the minimization of the total number of slices for the purpose of reducing the printing time.

III. OUR METHOD

A. RELATION BETWEEN SLICES AND PRINTING TIME

We observe that the amount of the slices significantly affects the printing time. Here a slice is a closed 2D polygon and a layer is generated by intersecting a slicing plane with a 3D model which may contain more than one slice. In Figure 2, we show three different layers. The layers in the left and middle images have only one slice while the layer shown in the right image has four slices.

To prove the above observation, we conduct two experiments. As shown in Figure 3, in the first experiment, we print a big cylinder whose diameter is 90 mm and height is 150 mm. The printing time is 6 hours and 23 minutes. In the second experiment, we simultaneously print 9 small cylinders whose diameters are 10 mm and heights are 150 mm. The total printing time is 7 hours and 46 minutes. Noticing that these models are only manufactured with shells (hollowed models) and the bottom and top surfaces are not printed to ensure these two experiments have the same surface area, that is the same amount of material. It is obvious that the second experiment has more number of slices. As a result, more than 21.7% printing time is wasted in the change of printing status between different slices, that is printing, movement, extrude, and non-extrude.

We can clearly see from the above analysis that when printing two models with the same surface areas, for instance printing a model in two different postures, the larger the number of slices, the longer the printing time. This issue is mainly caused by the printing principle of the most 3D printing technologies. We know that when printer finish printing a slice (closed polygon), the nozzle should slow down, and stop extruding material, then move to another slice, start extruding material, and printing. This sequence will waste several printing time. If a model is sliced into a large number of slices, many printing time will be wasted in this sequence. Therefore, one efficient method to reduce printing time is to reduce the number of slices which will be our major goal.
under the current posture, the number of vertex of the input model into an optimal posture. Furthermore, extra supporting structures are always needed for most 3D printing technologies at the overhang regions to avoid collapsing during the manufacturing process. However, the printing of these supporting structures will waste extra printing time. Different printing directions will result in very different overhang areas. If only the number of slices is considered in the optimization of printing direction, the generated printing direction may result in a large overhang areas. As a result, the whole printing time may not be reduced. Therefore, we should reduce the number of slices while not increasing overhang areas or even reduce the overhang areas which can better reduce the final printing time. Finally, the weighted number of slices and overhang areas is used as our objective function and the proposed optimization framework is given as follows:

\begin{align}
min_{r_x, r_y, r_z} & \quad L(M) + \lambda A(M), \\
\text{s.t.} \quad r_x & \in [-\pi, \pi], \\
r_y & \in [-\pi, \pi], \\
r_z & \in [-\pi, \pi],
\end{align}

where, \(L(M)\) represents the number of slices of \(M\) under the current posture, \(A(M)\) denotes the overhang areas of \(M\) under the current posture, \(r = [r_x, r_y, r_z]\), and \(r_x, r_y,\) and \(r_z\) indicate the rotation angle along \(x-, y-,\) and \(z-\)axis, respectively. To guarantee that all angles can be traversed, \(r_x, r_y,\) and \(r_z\) are restricted to \([-\pi, \pi]\). \(\lambda\) is a parameter to balance between the number of slices and overhang areas.

After solving formulation (1), we are able to obtain three rotation angles \(r_x, r_y,\) and \(r_z\). Then we rotate the model \(M\) into the optimal posture \(M'\) along \(x-, y-,\) and \(z-\)axis where the rotation matrix are represented as:

\[
R_x = \begin{bmatrix}
1 & 0 & 0 \\
0 & \cos(r_x) & \sin(r_x) \\
0 & -\sin(r_x) & \cos(r_x)
\end{bmatrix},
\]

\[
R_y = \begin{bmatrix}
\cos(r_y) & 0 & \sin(r_y) \\
0 & 1 & 0 \\
-\sin(r_y) & 0 & \cos(r_y)
\end{bmatrix},
\]

\[
R_z = \begin{bmatrix}
\cos(r_z) & \sin(r_z) & 0 \\
-\sin(r_z) & \cos(r_z) & 0 \\
0 & 0 & 1
\end{bmatrix}.
\]

Then, the final coordinate matrix of \(M'\) is obtained as:

\[
X' = X \ast R_x \ast R_y \ast R_z.
\]

Since formulation (1) is a highly non-linear optimization problem, most of the existing optimization algorithms cannot be used directly. In this work, Genetic Algorithm [1] is used to effectively solve this problem. The implementation details of our algorithm will be introduced in the next section.

IV. IMPLEMENTATION WITH GENETIC ALGORITHM

Genetic Algorithm (GA) is composed of several critical parts: initialization (gene coding), fitness function, selection operation, crossover operation, mutation operation, and termination. In the following, we will describe how to implement each step of GA for our formulation.

A. INITIALIZATION

The process begins with a set of individuals which is named as Population. Each individual in Population is a solution to the problem you want to solve. An individual is characterized by a set of parameters (variables) known as Genes.

B. THE PROPOSED ALGORITHM

For an input 3D triangle mesh \(M\), we use \(X\), a \(n \times 3\) matrix, to represent the 3D coordinates of \(M\)’s vertex where \(n\) is the number of vertex of \(M\), and the \(i^{th}\) row of \(X\) is the coordinate of the \(i^{th}\) vertex of \(M\). In addition, we assume that all models will be printed along \(z-\)axis of a global coordinate system.

As analyzed in Sec. (III-A) that an efficient way to reduce printing time is to reduce the number of slices. Since we fix the printing direction as the \(z-\)axis, that is \([0, 0, 1]\), to reduce the number of slices, we need to find a rotation matrix to put the input model into an optimal posture. Furthermore, extra supporting structures are always needed for most 3D printing technologies at the overhang regions to avoid collapsing during the manufacturing process. However, the printing of these supporting structures will waste additional printing time. Different printing directions will result in very different overhang areas. If only the number of slices is considered in the optimization of printing direction, the generated printing direction may result in a large overhang areas. As a result, the whole printing time may not be reduced. Therefore, we should reduce the number of slices while not increasing overhang areas or even reduce the overhang areas which can better reduce the final printing time. Finally, the weighted number of slices and overhang areas is used as our objective function and the proposed optimization framework is given as follows:

\begin{align}
\min_{r_x, r_y, r_z} & \quad L(M) + \lambda A(M), \\
\text{s.t.} \quad r_x & \in [-\pi, \pi], \\
r_y & \in [-\pi, \pi], \\
r_z & \in [-\pi, \pi],
\end{align}

where, \(L(M)\) represents the number of slices of \(M\) under the current posture, \(A(M)\) denotes the overhang areas of \(M\) under the current posture, \(r = [r_x, r_y, r_z]\), and \(r_x, r_y,\) and \(r_z\) indicate the rotation angle along \(x-, y-,\) and \(z-\)axis, respectively. To guarantee that all angles can be traversed, \(r_x, r_y,\) and \(r_z\) are restricted to \([-\pi, \pi]\). \(\lambda\) is a parameter to balance between the number of slices and overhang areas.

After solving formulation (1), we are able to obtain three rotation angles \(r_x, r_y,\) and \(r_z\). Then we rotate the model \(M\) into the optimal posture \(M'\) along \(x-, y-,\) and \(z-\)axis where the rotation matrix are represented as:

\[
R_x = \begin{bmatrix}
1 & 0 & 0 \\
0 & \cos(r_x) & \sin(r_x) \\
0 & -\sin(r_x) & \cos(r_x)
\end{bmatrix},
\]

\[
R_y = \begin{bmatrix}
\cos(r_y) & 0 & \sin(r_y) \\
0 & 1 & 0 \\
-\sin(r_y) & 0 & \cos(r_y)
\end{bmatrix},
\]

\[
R_z = \begin{bmatrix}
\cos(r_z) & \sin(r_z) & 0 \\
-\sin(r_z) & \cos(r_z) & 0 \\
0 & 0 & 1
\end{bmatrix}.
\]

Then, the final coordinate matrix of \(M'\) is obtained as:

\[
X' = X \ast R_x \ast R_y \ast R_z.
\]
Genes are joined into a string to form a Chromosome (solution). In Genetic Algorithm, the set of genes of an individual is represented using a string, in terms of an alphabet. Usually, binary values or symbols are used (string of 1s and 0s). In this work, binary values are used to encode the rotation angle vector \( \mathbf{r} = [r_x, r_y, r_z] \).

Generally, a binary code sequence of a certain length can only represent floating point numbers of a certain precision. In our algorithm, the precision is set as \( 1^\circ \), that is separating \([-\pi, \pi]\) into 360 intervals. Since \( 256 = 2^8 \times 360 < 2^9 = 512 \), so a rotation angle should be encoded with at least 8 binary numbers. As we have three rotation angles, the final length of our Chromosom is 24 which is denoted as \((b_0, b_1, \ldots, b_{23})\).

Given a Chromosom \((b_0, b_1, \ldots, b_{23})\), we need to transform it into rotation angles \(r_x, r_y, r_z\) following the next three steps: First of all, we separate \((b_0, b_1, \ldots, b_{23})\) into three sub-binary sequences with the same length. According to the encoding order, \((b_0, b_1, \ldots, b_7)\), \((b_8, b_9, \ldots, b_{15})\), and \((b_{16}, b_{17}, \ldots, b_{23})\) correspond to \(r_x, r_y, r_z\), respectively.

Secondly, for a sub-binary sequence \((b_0, b_1, \ldots, b_7)\), it is transformed into a decimal, that is rotation angle, with the following equation:

\[
 r_x = (b_0, b_1, \ldots, b_7)_2 = (\sum_{i=0}^{7} b_i 2^i)_{10}, \quad (9)
\]

where, \((.)_2\) and \((.)_{10}\) represent binary and decimal number, respectively. \(r_x\) and \(r_z\) can be obtained with the same way.

Finally, rotation angles generated with Equation (9) cannot be guaranteed to be located in interval \([-\pi, \pi]\). Therefore, we should project the obtained angles into this interval as:

\[
 r_x^* = -\pi + r_x - \frac{(\pi - (-\pi))}{2^8 - 1}. \quad (10)
\]

\(r_x^*\) and \(r_z^*\) can be obtained in the same way.

Through the above three steps, we are able to transform a Chromosom into rotation angles. Then, the model is rotated, and the number of slices and overhang areas are calculated.

### B. FITNESS FUNCTION

The fitness function determines how fit an individual is (the ability of an individual to compete with other individuals). It gives a fitness score to each individual. The probability that an individual will be selected for reproduction is based on its fitness score. In the proposed algorithm, fitness function is the weighted slice number and overhang areas.

1) SLICE NUMBER

The number of slices directly affects the printing time. For a given 3D triangle model, the more the number of slices, the longer the printing time. For each rotated model, the modified Cura code [30] which is an open source code, is used to calculate the number of slices.

2) OVERHANG AREAS

In 3D printing process, overhang regions always need extra supporting structures to help print (see Figure. 4). The larger the overhang areas, the more the supporting structures, and the longer the printing time. Therefore, we should try to reduce the overhang areas as well as reducing the number of slices. In this work, we mainly focus on 3D triangle mesh. The outer normal of a triangle is denoted as \(n_f = (n_{fx}, n_{fy}, n_{fz})\). If \(n_{fc} < -\frac{\sqrt{2}}{2}\), this triangle is considered as overhang triangle, as the printing direction \(d\) is fixed as \([0, 0, 1]\). The area of all overhang triangles are summed as the overhang areas of the model \(M\).

As a result, the fitness function of our algorithm is represented as:

\[
 f(M) = L(M) + \lambda A(M). \quad (11)
\]

### C. SELECTION OPERATION

Selection operation is to select the fittest individuals and let them pass their genes to the next generation. Individuals with high fitness have more chance to be selected for reproduction. In this work, roulette selection method is used to select individuals to generate the next generation. Assuming the current population has \(n\) individuals, and the population is denoted as:

\[
 P = \{p_i|i \in [1, n]\}, \quad (12)
\]

where, \(p_i\) is the \(i^{th}\) individual.

The fitness of the \(i^{th}\) individual is represented as \(f_i\). The sum of the fitness for all individuals is \(s = \sum_{i=1}^{n} f_i\). Therefore, the possibility of each individual is calculated as:

\[
 s_i = f_i/s. \quad (13)
\]

### D. CROSSOVER OPERATION

Crossover is one of the most significant phase in Genetic Algorithm. For each pair of parents to be mated, a crossover point is chosen at random from within the genes. In our implementation, middle point is selected as crossover point.
TABLE 1. Statistics by only considering the number of slices. From the first column to the last column are the model names, the initial number of slices, the optimized number of slices, the number of evolutions, the heights of the models, and the total number triangles, respectively.

| Name  | Initial slice number | Optimized slice number | Evolutions | Height(mm) | Triangles |
|-------|----------------------|------------------------|------------|------------|-----------|
| Ant   | 336                  | 237                    | 11         | 134        | 16695     |
| Bird  | 173                  | 39                     | 8          | 13         | 124       |
| Bunny | 206                  | 121                    | 8          | 158        | 20183     |
| Chair | 284                  | 74                     | 10         | 117        | 17306     |
| Desk  | 323                  | 110                    | 6          | 86         | 27848     |
| Hand  | 221                  | 83                     | 9          | 140        | 13210     |
| Human | 277                  | 114                    | 5          | 145        | 11348     |
| Jar   | 196                  | 128                    | 5          | 140        | 7800      |
| Plane | 222                  | 51                     | 5          | 192        | 11234     |
| Piers | 163                  | 30                     | 7          | 135        | 8952      |
| Spider| 308                  | 182                    | 8          | 152        | 14498     |

TABLE 2. Desk model is tested with five different parameter $\lambda$. The second row lists the number of slices after optimization, while the third row lists the overhang areas after optimization.

| $\lambda$ | 0   | 0.1 | 0.5 | 5   | 10  |
|-----------|-----|-----|-----|-----|-----|
| Number of slices | 110 | 114 | 318 | 320 | 348 |
| Overhang areas   | 3047.8 | 2820.3 | 1027.8 | 1001.8 | 861.4 |

E. MUTATION OPERATION
For an individual, some of its genes can be subjected to mutation with a low random probability. This implies that some of the bits in the bit string can be flipped. Mutation operation can effectively prevent the local optimal. In this work, a small probability $\omega$ is given to change some points from 0 to 1, and change some points from 1 to 0.

F. TERMINATION OPERATION
The algorithm terminates if the population has converged (does not produce offspring which are significantly different from the previous generation). Then it is said that the genetic algorithm has provided a set of solutions to our problem. In this work, if four consecutive generations have the same fitness, our algorithm is terminated.

V. EXPERIMENTAL RESULTS
A. SETTINGS AND TESTED MODELS
Our algorithm is implemented in mixed C++ and Matlab, and it run on a PC with Intel(R) Core(TM) i7-7700 HQ @ 2.80 GHz and 16 GB memory. For all the tested 3D models, we use the same set of standard parameters. The number of populations is set as 100, invariable individuals in each generation is set as 25, mutation possibility $\omega$ is 0.1, and the maximal evolution is set as 100.

We test our algorithm on a large number of 3D triangle meshes and 11 of them are shown in this paper. The names of the models are listed in the first column of Table 1, Table 3, Table 4, and Figure 7. The initial posture of each model is randomly generated, see Figure 7. To accelerate the calculation, the slicing thickness is set as 1mm for all models, and the height and the number of triangles of the models are reported in Tab.1.

B. RESULTS
The fitness function of our algorithm is the weighted slice number and overhang areas. The weight $\lambda$ is used to balance between the number of slices and overhang areas. If one printing technology itself is support-free, we do not need to consider the overhang areas for it. To reduce the printing time, the optimal printing direction can be found by only considering the number of slices, that is set $\lambda = 0$. Figure 5 shows the Chair model and Plane model under the initial postures (first column) and the optimized postures (second column). The number of the slices under the initial postures and the optimized postures are reported at the bottom of each figure. From this figure we can see that the proposed algorithm can effectively reduce the number of slices by automatically choosing an optimal printing direction. More results are shown in Figure 7.

In Table 1, we list the testing results by only considering the number of slices for all models. The second column lists the number of slices of the initial models, the third column shows the number of slices of the optimized models, the last three columns show the number of evolutions, the height of the models, and the total number of triangles of the models. Comparing between the second and third columns we can see that our algorithm is able to largely reduce the number of slices by selecting the optimal printing direction.

For most additive manufacturing technologies, the extra supporting structures are always needed in the overhang regions, such as FDM. Although the number of the slices can be largely reduced by choosing an optimal printing direction, the overhang areas of the model may increase under this printing direction. Therefore, an optimal printing direction should be generated by considering both the overhang areas and the number of slices. In Table 2, the Desk model is tested on five different parameter $\lambda$ which are 0, 0.1, 0.5, 5, and 10, respectively. The number of slices and overhang areas are reported in the second and third rows, respectively. From this table we can see that with the increasing of $\lambda$, the number of slices increasing while the overhang areas decreasing which complies with our expectation. In the following experiments, we set $\lambda = 0.1$ as we hope the number of slices is our major goal. In Figure 6, Hand model is tested by only considering...
the number of slices (Figure 6(middle)) and both the number of slices and the overhang areas (Figure 6(right)). The number of slices for these models from left to right are 221, 83, and 330, respectively. The overhang areas from left to right are 4104.8 mm$^2$, 8228.2 mm$^2$, and 347 mm$^2$, respectively. Although the model in Figure 6(middle) has the smallest number of slices, but it also has very large overhang areas. In Figure 6(right), the number of slices and the overhang areas have some kind of balance.

In Table 3, we show the testing results by considering both the number of slices and the overhang areas for all models with $\lambda = 0.1$. From the second column to the last column, we show the initial overhang areas, the optimized overhang areas, the optimized slice number and the number of evolutions. From this table we can see that the overhang areas in the last column are the smallest. The overhang areas of some models in the third column are bigger than that in the second column. That is because that the overhang areas are not considered in the optimization when $\lambda = 0$.

The models stand at the initial postures, the postures optimized with $\lambda = 0$, and the postures optimized with $\lambda = 0.1$ are shown in Figure 7.

**C. DISCUSSIONS**

As we know that many aspects of 3D printing technologies are affected by printing direction, such as part accuracy, strength and surface finish, etc [10], [11]. However, the proposed method only take into account the printing time and supports. One of our future research topic is to integrate part accuracy, strength or other aspects into our optimization framework.

Different 3D printing technologies may have different support conditions. The proposed method is mainly focused on the FDM 3D printing technology. In future, we will try to propose more general algorithm which is suitable for most of 3D printing technologies.

Sometime, curved surfaces are always recommended to be built vertically such that the curved slice is built in z-axis direction. For example, building a high simple cylinder. In this case, we can increase the weight $\lambda$ to make the optimizer find the vertical printing direction through more printing time is needed.
Since the objects fabricated with 3D printing technologies may have very strong structural anisotropy, especially FDM printing technology, so the final objects are likely to delaminate and fracture when placed in tension in the built direction as compared to the other directions. In the proposed algorithm, we could also integrate the loading conditions in the optimization framework. In Equation (11), we can add an extra term which measure the structural performance with loading conditions. In the next step, we will test the result for this.

Our method is mainly focused on the optimization of one single 3D model. However, 3D printing technology offers a distinct advantage of printing different parts together in a single built. In this situation, our method can be used to...
optimize each of these parts and then rotate each part to make sure its ideal printing direction matches with z-axis.

Although most of the commercially available software auto orient the parts based upon their technologies and limitations, they only consider the directions along with axis or simply consider the overhang areas. As far as I know, no commercial 3D printing software take into account the printing time, that is the number of slices.

VI. CONCLUSION

In this work, we present a novel printing direction optimization algorithm by considering the number of slices and overhanging areas. Our major goal is to find an optimal printing direction for the purpose of reducing the printing time, since the number slices directly affects the printing time. However, the amount of overhang areas also affects the printing time. Therefore, the weighted slice number and overhang areas is used as our objective function, in which the slice number is our major consideration as it is first considered in this work. A large number of testing results demonstrate the effectiveness of the proposed algorithm.

However, our algorithm still has several drawbacks. First of all, restricted by Genetic Algorithm, our algorithm cannot guarantee a global solution. Secondly, the printing time is indirectly considered in this work which may differ from the actual printing time. In our future work, we will try to directly calculate the manufacturing time under a given slicing direction. Then, we can find an optimal slicing direction for the purpose of reducing the manufacturing time.

REFERENCES

[1] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” IEEE Trans. Evol. Comput., vol. 6, no. 2, pp. 182–197, Apr. 2002.
[2] L. Liu, W. Xu, W. Wang, Z. Yang, and X. Liu, “Survey on geometric computing in 3D printing,” Chin. J. Comput., vol. 38, no. 6, pp. 1243–1267, 2015.
[3] R. Z. Hu and H. Huang, “Recent progress in 3D printing inspired fabrication optimization,” J. Comput.-Aided Des. Comput. Graph., vol. 27, no. 6, pp. 961–967, 2015.
[4] K. V. Wong and A. Hernandez, “A review of additive manufacturing,” ISRN Mech. Eng., vol. 2012, pp. 1–10, Aug. 2012.
[5] M. Livesu, M. Attene, M. Spagnuolo, and B. Falcidieno, “A study of the state of the art of process planning for additive manufacturing,” Istituto di Matematica Applicata e Tecnologie Informatiche, pp. 1–13, May 2016.
[6] J. Gardan, “Additive manufacturing technologies: State of the art and trends,” Int. J. Prod. Res., vol. 54, no. 10, pp. 3118–3132, May 2016.

[7] N. Umetani and R. Schmidt, “Cross-sectional structural analysis for 3D printing optimization,” in Proc. SIGGRAPH Asia Tech. Briefs (SA), 2013, pp. 5.1–5.4.

[8] W. M. Wang, C. Zanni, and L. Kobbelt, “Improved surface quality in 3D printing by optimizing the printing direction,” Comput. Graph. Forum, vol. 35, no. 2, pp. 59–70, May 2016.

[9] X. Zhang, X. Le, A. Panotopoulou, E. Whiting, and C. C. L. Wang, “Perceptual models of preference in 3D printing direction,” ACM Trans. Graph., vol. 34, no. 6, pp. 1–12, Nov. 2015.

[10] S. H. Masood, W. Rattanawong, and F. Ioveniti, “Part build orientations based on volumetric error in fused deposition modelling,” Int. J. Adv. Manuf. Technol., vol. 16, no. 3, pp. 162–168, Feb. 2000.

[11] H. S. Byun and K. H. Lee, “Determination of optimal build direction in rapid prototyping with variable slicing,” Int. J. Adv. Manuf. Technol., vol. 28, nos. 3–4, pp. 307–313, Mar. 2006.

[12] H.-S. Byun and K. H. Lee, “Determination of the optimal build direction for different rapid prototyping processes using multi-criterion decision making,” Robot. Comput.-Integr. Manuf., vol. 22, no. 1, pp. 69–80, Feb. 2006.

[13] S. Danjou and P. Köhler, “Determination of optimal build direction for different rapid prototyping applications,” in Proc. 14th Eur. Forum Rapid Prototyping, Ecole Centrale, Paris, 2009, pp. 1–10.

[14] D. Frank and G. Fadel, “Expert system-based selection of the preferred direction of build for rapid prototyping processes,” J. Intell. Manuf., vol. 6, no. 5, pp. 339–345, Oct. 1995.

[15] F. Xu, Y. S. Wong, H. T. Loh, J. Y. H. Fuh, and T. Miyazawa, “Optimal orientation with variable slicing in stereolithography,” Rapid Prototyping J., vol. 3, no. 3, pp. 76–88, Sep. 1997.

[16] H. S. Byun and K. H. Lee, “Determination of the optimal build direction for different rapid prototyping processes using multi-criterion decision making,” Robot. Comput.-Integr. Manuf., vol. 22, no. 1, pp. 69–80, Feb. 2006.

[17] H. Zhao, F. Gu, Q.-X. Huang, J. Garcia, Y. Chen, C. Tu, B. Benes, H. Zhang, D. Cohen-Or, and B. Chen, “Connected fermat spirals for layered fabrication,” ACM Trans. Graph., vol. 35, no. 4, pp. 1–10, Jul. 2016.

[18] G. Dreifus, K. Goodrick, S. Giles, M. Patel, R. M. Foster, C. Williams, J. Lindahl, B. Post, A. Roschli, L. Love, and V. Kune, “Path optimization along lattices in additive manufacturing using the chinese postman problem,” 3D Printing Additive Manuf., vol. 4, no. 2, pp. 98–104, Jun. 2017.

[19] W. Wang, H. Chao, J. Tong, Z. Yang, X. Tong, H. Li, X. Liu, and L. Liu, “Saliency-preserving slicing optimization for effective 3D printing,” Comput. Graph. Forum, vol. 34, no. 6, pp. 148–160, Sep. 2015.

[20] M. Alexa, K. Hildebrand, and S. Lefebvre, “Optimal discrete slicing,” ACM Trans. Graph., vol. 36, no. 1, pp. 12:1–12:16, 2017.

[21] M. T. Hayasi and B. Asiabanpour, “A new adaptive slicing approach for the fully dense freeform fabrication (FDFF) process,” J. Intell. Manuf., vol. 28, nos. 3–4, pp. 307–313, Mar. 2006.

[22] Y. Wang, Z. Gu, L. Song, T. Li, H. Cui, and F. C. M. Lau, “Speeding up 3D printing using multi-head slicing algorithms,” in Proc. 5th Int. Conf. Enterprise Syst. (ES), Sep. 2017, pp. 99–106.

[23] P. M. Pandey, N. V. Reddy, and S. G. Dhande, “Slicing procedures layered manufacturing: A Review,” Rapid Prototyping J., vol. 9, no. 5, pp. 224–298, 2003.

[24] K. Mani, P. Kulkarni, and D. Dutta, “Region-based adaptive slicing,” Comput.-Aided Des., vol. 31, no. 5, pp. 317–333, Apr. 1999.

[25] P. M. Pandey, N. V. Reddy, and S. G. Dhande, “Real time adaptive slicing for fused deposition modelling,” Int. J. Mach. Tools Manuf., vol. 43, no. 1, pp. 61–71, Jan. 2003.

[26] M. T. Hayasi and B. Asiabanpour, “A new adaptive slicing approach for the fully dense freeform fabrication (FDFF) process,” J. Intell. Manuf., vol. 24, no. 4, pp. 683–694, Aug. 2013.

[27] K. Hildebrand, B. Bickel, and M. Alexa, “Orthogonal slicing for additive manufacturing,” Comput. Graph., vol. 37, no. 6, pp. 669–675, Oct. 2013.

[28] W. Wang, H. Chao, J. Tong, Z. Yang, X. Tong, H. Li, X. Liu, and L. Liu, “Saliency-preserving slicing optimization for effective 3D printing,” Comput. Graph. Forum, vol. 34, no. 6, pp. 148–160, Sep. 2015.

[29] M. Alexa, K. Hildebrand, and S. Lefebvre, “Optimal discrete slicing,” ACM Trans. Graph., vol. 36, no. 1, pp. 12:1–12:16, 2017.