

# Global Continuous Toolpath Planning with Controllable Local Directions



Yingxin Ma<sup>a</sup>, Yuan Yao<sup>a,b,c,\*</sup>, Jinxiu Yang<sup>a</sup>, Hang Zhang<sup>a</sup>, Beishui Liao<sup>c</sup>

<sup>a</sup> School of Mechatronic Engineering and Automation, Shanghai University, 99 Road Shangda, Shanghai, 200444, China

<sup>b</sup> Engineering Training Center, Shanghai University, 99 Road Shangda, Shanghai, 200444, China

<sup>c</sup> School of Philosophy, Zhejiang University, 866 Yuhangtang Rd, Hangzhou, 310028, China

## ARTICLE INFO

### Article history:

Received 15 January 2023

Received in revised form 19 June 2023

Accepted 9 July 2023

### Keywords:

Path planning

Global path continuity

Toolpath optimization

## ABSTRACT

Toolpath planning for Continuous fiber reinforced thermoplastic composites (CFRPCs) 3D printing is a challenging task, which is vital to decrease the process complexity. In this paper, we propose a global continuous path planning method for CFRPCs. First, for the set of 2D connected regions generated by slicing, we establish an optimized print sequence with a minimum idle stroke under interference constraints. Second, a continuous Zigzag path generation algorithm with controlled deposition direction is constructed for each 2D connected region, and path optimization is used to lessen the uneven deposition spacing. Finally, A separation axis theorem is used to prevent interference during the printing process, and we use a genetic algorithm to choose the connection points between 2D paths in a given spatial printing sequence to make the minimum idle stroke, creating a global continuous path with controllable deposition direction. The method proposed in this paper can provide the ability to customize the local deposition path direction and ensure the continuity of the global path to avoid interference and reduce the post-processing workload, so it is suitable for continuous path planning and multi-physics field optimization of complex structures, which can fully utilize the advantages of high-performance fibers to improve the customization ability and the performance of the fabricated parts for 3D printing and provide support for the manufacturing of high-performance structures in different fields.

© 2023 Elsevier Ltd. All rights reserved.

## 1. Introduction

Fused Filament Fabrication (FFF) is a widely used process in additive manufacturing (AM) [1] due to its low cost and compatible with a large of materials [2]. However, printing with pure polymer materials does not meet the strength requirements in industrial applications, which is a major problem for the FFF. Now one of the widely used strategies to improve the mechanical properties of printed parts is using high-performance continuous fiber such as carbon fiber, glass fiber, and Kevlar, as the reinforcement material and using polymer composites as the matrix part. Among them, continuous fiber-reinforced thermoplastic composites (CFRPCs) have the advantages of good manufacturability and become the main form of reinforcement. Usually, the tensile strength of CFRPCs is an order of magnitude higher compared to staple fiber reinforced polymers [3–5].

In CFRPCs printing, path direction, continuity, volume fraction, porosity and printing speed are all key parameters that affect the quality of printed parts. CFRPCs printers need additional equipments to provide fiber cutting [6]. Frequent cut-off

operations can severely reduce printing efficiency and quality. In addition, the difference in the axial and radial properties of the fiber-reinforced material itself will further increase the impact of printing filament arrangement on the performance of 3D printed parts in all directions [7–9]. Therefore, continuity and directional control are crucial for CFRPCs 3D printing. By ensuring global continuity and avoiding frequent cutting operations, the efficiency of continuous fiber-reinforced printing can be increased, hardware costs can be reduced, and it is suitable for CFRPCs printing on general FFF platforms. Using infill patterns with controllable path directions can achieve local customization of the path direction, providing an effective path-planning tool for various physics-based optimizations. It can achieve a more uniform filling and reduce internal voids when filling into structures with complex geometries or sharp features. In addition, the controllable path direction can simplify the design of interlayer staggered filling planning and can also be integrated with other path planning strategies to strengthen the interlayer connection.

This paper proposes a global continuous path planning method with controlled deposition direction to generate continuous paths with a uniform infill spacing in controlled deposition direction by geometric decomposition, partitioning method, and global optimization strategies on 2D slices. To establish a global continuous

\* Corresponding author.

E-mail address: [yaoyuan@shu.edu.cn](mailto:yaoyuan@shu.edu.cn) (Y. Yao).

path for printing complex structures, idle stroke length minimizing in overall printing is required. The main contributions of this paper are described as followed:

1. We propose a new algorithm for the geometric decomposition of 2D contours and continuous filling paths generation with controlled local fiber deposition direction. It offers fiber direction control techniques and path generation tools for various kinds of physical field-driven material forming optimization processes.
2. We provide a method for designing global continuous filling paths for intricate models that can significantly cut down on idle strokes and post-processing work when printing multi-branch structures at CFRTCs.
3. Optimization methods under the constraint of global continuity are provided to avoid path interference when switching between successive processing multilayer structures. For 2D continuous paths, it reduces the unevenness of deposition pitch and improves the stability and quality of printing.

## 2. Related works

In the last decade, CFRTCs technology has evolved rapidly and path planning for CFRTCs printing has become an important problem. In industry, CFRTCs printing equipments provider Markforged has developed the Eiger which provides both polymer and long fiber reinforced paths in the form of auxiliary reinforcement to enable printing of fiber reinforced paths [10]; Anisoprint has developed AURA which generates reinforcement fiber trajectories in each layer of 3D printing and allows to adjust the reinforcement scheme to obtain strong and lightweight parts [11]; AREVO's AQUA system combines CFRTCs technology with robotics that mounts AQUA's deposition head on a six-axis robot that heats the incoming filament by laser and compacts the heated filament by rollers during deposition [12]. Various methods for continuous path planning have also been proposed successively in academia, which are summarized in this paper as global continuous path planning and physical field-based optimization are reviewed separately to discuss their progress and problems.

### 2.1. Global continuous path

Maintaining path continuity is a special requirement in CFRTCs. Dwivedi et al. [13] proposed a torch path planning method for welding-based SFF, which uses turn vertexes to decompose 2D contours into multiple homotopy sub-polygons, and each homotopy sub-polygon generates a path and then joins them into a whole. Ding et al. [14] use a partitioning strategy to decompose concave polygons into convex sub-polygons, the method defines the edge forming an interior angle greater than  $180^\circ$  as a notch and tries to eliminate the notch by three different methods. Gomez et al. [15] made an improvement to Ding's method, which generates fewer convex sub-polygons when decomposing concave polygons. Jin et al. [16] proposed a "go and back" strategy to generate continuous paths using a reference line and formulating conditions to generate dividing lines to decompose the polygons and use adjacency trees to connect the sub-paths.

Zhao et al. [17] improved the classic Fermat spiral by providing a lengthy, low-curvature continuous Fermat spiral path, decomposing a domain into a collection of subregions based on the location of the offset curve, filling a continuous Fermat spiral curve in each subregion by rerouting and linking, and lastly joining the subpaths to generate a global continuous path.

The method is also appropriate for the print path generation of CFRTCs because of the method's lower curvature. The method, however, calls for the starting and exit points to be roughly in the same position and may have a high underfill rate. Zhai et al. [18] further extended the method to porous structures; they first introduce a framework to model a type of porous structure that contain lots of small holes. Then they propose an efficient path planning algorithm for this type of porous structure to improve the filling rate of the connected Fermat path. Bedel et al. [19] proposed the closed space-filling curves with controlled orientation for 3D printing. The path generation is flexible and it has more complete mathematical arguments compared to zigzag-type of paths.

Xia et al. [20] proposed a hybrid global continuous path that combines contour-parallel paths and dual Zigzag paths. Compared with other methods, this method does not require the geometric decomposition of polygons. Bi et al. [21] proposed a continuous hybrid path for large-format additive manufacturing. The method focuses on solving the problems of generating sharp corners, overfilling, and underfilling in the central part of the conventional contour-parallel paths, and proposes an optimization algorithm for the overfilling and underfilling cases.

It is rather typical to use graph theory to address the issue of generating a continuous path. Wang et al. [22] view each slice of a lightweight filled model as a graph, construct Eulerian paths by deleting edges and adding edges, and find Eulerian paths in the graph using the Heilholzer algorithm. Gupta et al. [23] mesh polygons so that each vertex and edge in a two-dimensional complex form constitutes a two-dimensional graph, converting the problem of generating continuous paths into the problem of finding Eulerian loops in directed graphs and generalizing the Eulerian transform.

The creation of continuous paths in 2D has been the main focus of the studies of the aforementioned methods. The optimization of the idle stroke and the requirement to prevent path interference when printing branching structures after optimization are additional issues for 3D global continuous paths that cannot be disregarded in real-world manufacturing. For conventional FFF printing, optimizing the idle stroke can improve printing efficiency and quality. For continuous fiber-reinforced printing without the shearing device, there will be fiber extrusion even during the idle stroke, and eventually, there will be material redundancy, which increases the difficulty of post-processing. Yoo et al. [24] give an algorithm for FFF toolpaths using Monte Carlo trees search (MCTS) to optimize the idle stroke. Zhong et al. [25] propose the as-continuous-as-possible printing toolpaths and a novel geometric criterion "one-path-patch" (OPP) to decompose the input surface into a minimal number of continuous printing patches with flat and curved collision-free paths. Wang et al. [26] proposed a method to determine the printing sequence with the constraint of no-collision for support-free 3+2-axis printing. Li et al. [27] models the nozzle trajectory accurately to predict whether a collision will occur with an already printed part when printing a contour of a layer.

### 2.2. Physical field driven path filling

Physical field-driven optimization is a popular strategy for print path optimization for CFRTCs, using heat transfer, force transfer, and electromagnetic fields as references to optimize the corresponding properties of the printed part, with force fields being the main form. This strategy aims to fully exploit the properties of additive manufacturing and high-performance fiber materials. [28–31]. The two main types of force field-driven path optimization techniques are: (1) filling along the direction of the principal stress field, which has the obvious drawback that

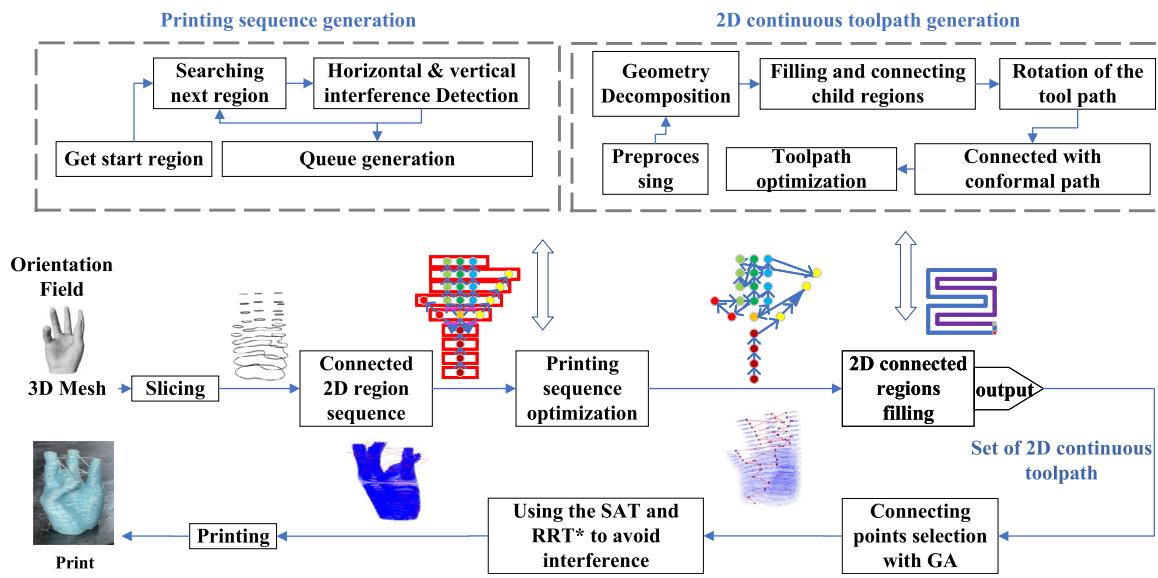


Fig. 1. Framework.

the filling spacing is very uneven and does not ensure global continuity in 2D; and (2) partitioning the 2D connected area by the direction of the principal stress and filling, which has the advantage of more regular filling spacing but cannot guarantee the continuity and uniformity of the path due to the geometric shape limitations.

Xiao et al. [32] proposed an in-plane isotropic filling path pattern to improve the mechanical strength of FFF printed parts. Their in-plane isotropy is achieved by continuous deposition in a random distribution within a layer. The tensile strength of the parts fabricated by this method can be improved by at least 20% compared to the specimens generated by conventional tool paths. Chen et al. [33] proposed a field-based toolpath generation method for continuous fiber-reinforced thermoplastic composites. The tool paths were generated along the direction of tensile stress in the critical region, and the anisotropy of continuous fibers was used to improve the mechanical strength of the printed parts. Compared with the load-independent tool paths, the samples produced by this method showed a 71.4% improvement in mechanical strength in physical tests when the same (or even slightly smaller) number of continuous fibers were used. An innovative method for enhancing the fill path to match the major stress field of the employed part was put forth by Li et al. [34]. The irregular vector field was represented for the first time as a grayscale image, allowing for the stable generation of subregions and matching print pathways. Compared to the usual method, the method yields parts with a tensile strength that is about 20% higher. Zhang et al. [35] proposed a method based on principal stress partitioning to optimize the print path, which reduces the anisotropy of the FFF process by vertically staggering the filling between layers, thus improving the actual performance of the FFF process. The strength of the parts printed by this method was increased by 17% compared to filling methods such as the conventional Zigzag filling strategy. Xia et al. [36] proposed a path-generation strategy based on the direction of the maximum principal stress. The method calculates the stresses of the part by finite elements, discretizes the polygon by hexagonal meshes, and then generates paths parallel to the maximum principal stress direction using Depth First Search(DFS) and connection criterion, plans the connection angle between the connections of each node, reduces the sharp turn of the overall path, and proposes its optimization method for the continuity of the path. The method has a significant improvement in the strength of the printed

parts, with 44.65% and 48.86% improvement in tensile strength compared to Zigzag and contour parallel.

In conclusion, path planning has emerged as a hot area of research right now, but it is still in the experimental stage. First, geometry decomposition is a common method used in continuous path planning, but most of the methods are only applicable to regular polygons. The strategy of completely decomposing concave polygons into convex sub-polygons is easy to generate too small and too many subregions when dealing with polygons with complex boundaries, and these sub-regions will generate holes in the print if they are directly discarded. How to avoid or deal with these too-small subregions is one of the problems to be solved in this paper. Second, both space-filling continuous paths and hybrid continuous paths have high filling accuracy; however, because most methods cannot control the filling direction, they cannot be cross-filled to improve print performance [37], much less in the case of multi-physics field-driven generation of paths with different filling angle requirements.

### 3. Method

The method is divided into three parts: *Main process flow*, *Printing sequence optimization* and *2D continuous toolpath generation*. The *Main process flow* uses the traditional slicing method to create a collection of 2D connected regions based on the input of the triangle mesh model and Orientation Field. The *Printing sequence optimization* is used to reorder all 2D connected regions in order to minimize the idle stroke length. The *2D continuous toolpath generation* fills each 2D connected region using the Orientation Field, which can be generated automatically by pre-physical field simulation or specified manually. A global continuous tool path is created by connecting the 2D continuous tool paths in order to decrease idle stroke. Finally, the global continuous tool path is checked for collisions and interference and corrections are made as necessary. The framework of this method is shown in Fig. 1.

#### 3.1. Two-dimensional continuous toolpath generation

The goal of *2D continuous toolpath generation* is to generate directionally controllable continuous paths in independent connected regions, which is the basis for exploiting the anisotropy

of high-performance fibers. *Printing sequence optimization* is required before performing 2D continuous toolpath generation in the *Main process flow*, but as a relatively independent process, we will first clarify how to implement this approach and then embed it in the *Main process flow*. The input of the algorithm is an arbitrary 2D connected region, fill angle, and fill spacing, and the output is a continuous deposition path. The key to this algorithm is our suggested geometry decomposition method, which can eliminate unnecessary decomposition and sub-regions that are too small. Also crucial is our tool path optimization algorithm, which uses the connected area as a boundary constraint, designs the objective function for the path's fill spacing, and iteratively optimizes it to improve the actual printing effect. The process of implementing this method will be discussed in detail in Section 4. The algorithm process is as follows:

1. The preprocess of 2D connected regions is to get the conformal tool path and the set of fill regions, and convert any angle of fill to horizontal fill by inverting the target region by one fill angle.
2. Geometry decomposition of the filled region, filling sub-regions with a local continuous Zigzag, and connecting these sub-paths to generate a continuous path.
3. The continuous paths generated from all filled regions are rotated by the fill angle and then connected to the conformal tool paths to generate a continuous path with a controlled deposition direction.
4. Finally, we design the objective function for the fill spacing of the path with the boundary constraint of the connected region and iteratively optimize it, and smooth the path after the optimization is completed.

### 3.2. Idle stroke length optimization

The implementation of a continuous fiber-reinforced printing toolpath depends on idle stroke length optimization, which makes it possible to print a single branch continuously and hence significantly reduce shearing and post-processing operations. In 3D printing, the idle stroke is the stroke in which the extrusion head moves but does not extrude material. For printing with continuous fibers, especially without a shearing device, continuous fibers are still present in the idle stroke, so the length of the idle stroke must be reduced. Idle stroke length optimization is divided into three main parts, the flow of the method is as follows, and the specific implementation process will be discussed in detail in Section 5.

1. *Printing sequence optimization*: The input is a set of slices arranged by the z coordinate of the layer, and the print sequence of the slices is generated with the objective of minimizing the sum of the distances between the slice contours while avoiding horizontal interference and height interference. The goal of the method is to have the printer head leap to the branch that is nearest to the current branch whenever interference is projected to occur during the subsequent print operation; otherwise, the printer head will continue printing the current branch.
2. *Connecting point selection with GA*: The input to the algorithm is a set of 2D continuous toolpaths with controllable deposition direction. Our approach uses a genetic algorithm to select connecting points between toolpaths in an optimized print order so that the idle stroke length is minimized. To make the genetic algorithm converge better, we sample the tool paths and use the sampled point sequences to participate in the encoding, while constraining the connecting points to be on the conformal tool paths, and finally output the global continuous paths.

3. *Using the Separating Axis Theorem(SAT) and the Rapid-exploration Random Tree Star(RRT\*) to avoid interference*: the global continuous path is the input. The algorithm adjusts idle strokes to avoid collisions. Starting at the root node of a random search tree and ending at the target point, the RRT\* algorithm generates new nodes in the sampling space, which are checked for collisions using SAT. If a new node satisfies this constraint, it is added to the random search tree as a leaf node. This process is repeated until the distance between the leaf node and the target point is within a set threshold, and the path from the leaf node to the target point is collision-free. The target point is then added to the random search tree, and the algorithm backtracks the target point to the root node to obtain a safe path. This path corresponds to an adjusted idle stroke.

## 4. Two-dimensional continuous toolpath generation

An essential factor in the printing of customized CFRTCs is the direction in which the reinforcing fibers are laid. Zigzag routes can facilitate the development of continuous paths and have a clear directional characteristic as compared to Fermat spiral, parallel contour filling paths. In this work, the basic filling mode is set to Zigzag. This path pattern is unsuitable for fiber-reinforced materials with high stiffness, such as carbon fibers.

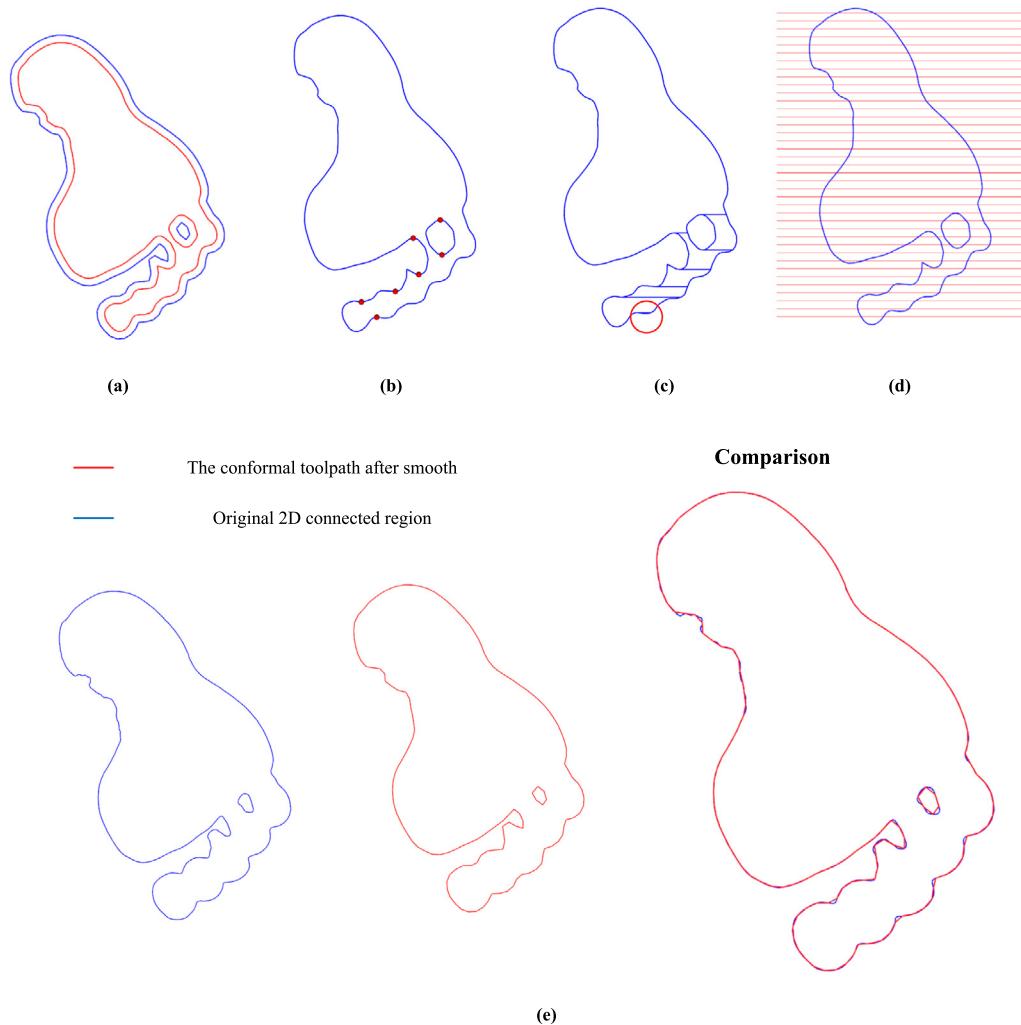
### 4.1. Pre-processing

First, in order to balance the computational performance and accuracy, we need to preprocess the input 2D connected region  $\mathbf{R}$ . Let the fill angle be  $\alpha$  and the fill spacing be  $w$ .

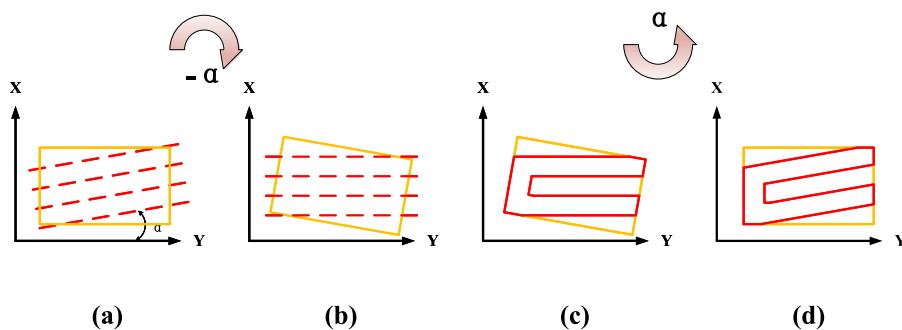
1.  $\mathbf{R}$  is smoothed and filtered the noise by letting its coordinates convolve with the Gaussian function [38] according to Eqs. (1) and (2), where  $x(n)$  denotes the x-coordinate of the  $n$ th point in  $\mathbf{R}$  and  $g(k, \sigma)$  denotes the  $k$ th value of the derivatives of Gaussian with  $\sigma$  as the standard deviation. the derivatives of Gaussian are sampled at  $2L$  points. Take  $2L$  as the even integer closest to  $M\sigma$ , where  $M$  is an integer. The values of  $\sigma$  and  $M$  are affected by the perimeter of the contour. Too large  $\sigma$  and  $M$  will lead to distortion of the contour, too small  $\sigma$  and  $M$  are not good for smoothing. Let  $\sigma = 0.05$ ,  $M = 2$ , and the effect is shown in Fig. 2(e).
2. Offset  $\mathbf{R}$  inwards and let the offset distance as  $w$  to get the set of fill regions  $\mathbf{o} = [o_1, o_2, \dots, o_n]$ . The conformal tool path  $b$  is obtained after Equidistant sampling of  $\mathbf{R}$  at  $w$  interval, as shown in Fig. 2(a). Next, we need to generate a continuous path  $\zeta_i$  of  $o_i$ . Then, all  $\zeta_i$  are connected to  $b$  to generate a 2D continuous toolpath  $\gamma$ .
3. The horizontal global scan lines are generated with an interval of  $w$  from the lowest point to the highest point of  $o_i$ , as shown in Fig. 2(d).
4. Rotating the  $o_i$  by  $-\alpha$  converts any angles of fill to horizontal fill. The principle of fill angle conversion is shown in Fig. 3.
5. Use the *turn vertexes* to decompose polygons [13]. Let the set of *turn vertexes* be  $\mathbf{t} = [t_1, t_2, \dots, t_n]$ . The *turn vertexes* of polygon are shown in Fig. 2(b). The effect of the method proposed by Dwivedi [13] is shown in Fig. 2(c).

$$X(n, \sigma) = \sum_{k=-L}^L x(k-n)g(k, \sigma) \quad (1)$$

$$Y(n, \sigma) = \sum_{k=-L}^L y(k-n)g(k, \sigma) \quad (2)$$



**Fig. 2.** Schematic diagram of each step of pre-processing. (a):  $b$  and  $\mathbf{o}$ , where blue solid line is  $b$ , red solid line is  $\mathbf{o}$ , and the  $\mathbf{o}$  contains one fill region in this example; (b): Turn vertexes of  $\mathbf{o}_i$  is rotated by  $-\alpha$ , where  $\alpha = 10^\circ$ , red points are the turn vertexes; (c): The schematic diagram of geometry decomposition proposed by Dwivedi [13]. The red circle indicates the unreasonable sub-region; (d): Diagram of the global scan lines and  $\mathbf{o}_i$ . Fill interval is  $w$ , Red solid line is the scan line; (e): Smoothing effect picture. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

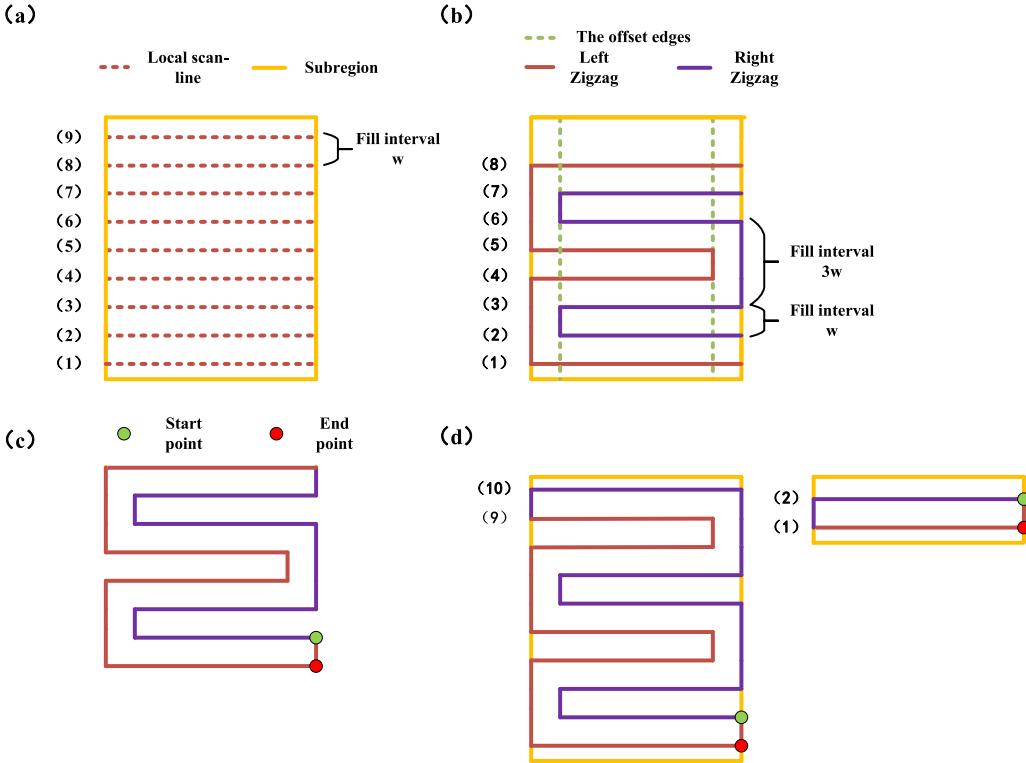


**Fig. 3.** Filling angle conversion diagram. (a): Schematic diagram of fill angle and  $\mathbf{o}_i$ . Red dashed lines denote the global scan lines of  $\mathbf{o}_i$  and yellow solid lines denote  $\mathbf{o}_i$ ; (b): Rotate  $\mathbf{o}_i$  by  $-\alpha$  and generate horizontal global scan lines of  $\mathbf{o}_i$ ; (c):  $\zeta_i$  of  $\mathbf{o}_i$ . Red solid lines denote  $\zeta_i$ ; (d): Rotate  $\zeta_i$  by  $\alpha$ . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

#### 4.2. Geometry decomposition

In this section, we get the  $\mathbf{o}_i$  and  $\mathbf{t}$  from Section 4.1. Next, we try to decompose the complex 2D connected region and generate a local continuous Zigzag toolpath in its subregions.

To create the local continuous Zigzag toolpath shown in Fig. 4, we first obtain even local scan lines in the subregion, if the number of local scan lines is odd, we need to choose one of the top or bottom scan lines that is shorter in the X-axis direction to round off, as shown in Fig. 4(a). The remaining scan lines generate 3w and w interleaved left and right Zigzag paths. The left Zigzag



**Fig. 4.** Schematic diagram of a local continuous Zigzag. (a): Schematic diagram of sub-regions with local scan lines; (b): Schematic diagram of the left and right Zigzag. The green dashed line indicates the offset edges of the subregion after offsetting a fill spacing, the red solid line indicates the left Zigzag and the purple solid line indicates the right Zigzag; (c): Schematic diagram of the local continuous Zigzag; (d): Schematic of the local continuous Zigzag generated by other numbers of local scan lines. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

starts from the first scan line and the right Zigzag starts from the second scan line. We then offset the left and right edges of the subregion inward by a fill spacing, as shown by the green dashed line in Fig. 4(b).

The left and right Zigzags, shown in Fig. 4(b), generate zigzag along the opposite offset edges when the interval between zig and zig is equal to  $w$ . Connect the left and right zigzag counter-clockwise along the contour, as shown in Fig. 4(c). If there is no local scan line in the sub-region, the outline of the sub-region is made as the fill path. In the end, we get fill paths that are all oriented counterclockwise.

We need to discuss how to decompose polygons to generate subregions suitable for generating continuous paths. The traditional partitioning method in Fig. 2(c) generates many unnecessary subregions, or the generated partitions are too small to be filled. To address this issue, we propose an improved geometry decomposition algorithm.

$t_i$  can have three states:  $Q_i = 0, 1, 2$ .  $Q_i = 0$  means no geometry decomposition is performed at  $t_i$ ,  $Q_i = 1$  means geometry decomposition is performed to the left at  $t_i$  by drawing a horizontal line through it and connecting it to the closest intersection point on the left side of  $t_i$ , and  $Q_i = 2$  means geometry decomposition is performed to the right at  $t_i$  by connecting it to the closest intersection point on the right side of  $t_i$ .

Let  $\mathbf{Q} = [Q_1, Q_2 \dots, Q_n]$  denote the state of each turn vertexes in  $\mathbf{t}$ ,  $\mathbf{S}$  is the set of the final generated subregions, and  $f(x)$  returns the reciprocal of the number of scan lines in the subregion with the minimum number of scan lines among subregions  $x$ . Based on the above description, the objective function of our geometry decomposition can be shown in Eq. (3). However, to traverse all possibilities of  $\mathbf{Q}$  to pick the global optimal solution, the complexity of the algorithm is  $O(3^n)$ , where  $n$  is the length of  $\mathbf{t}$ . Therefore, we use the greedy algorithm to determine the

state of each  $t_i$ . The pseudo-code of the geometry decomposition algorithm in this paper is given below. Diagram of the geometry decomposition algorithm is shown in Fig. 5.

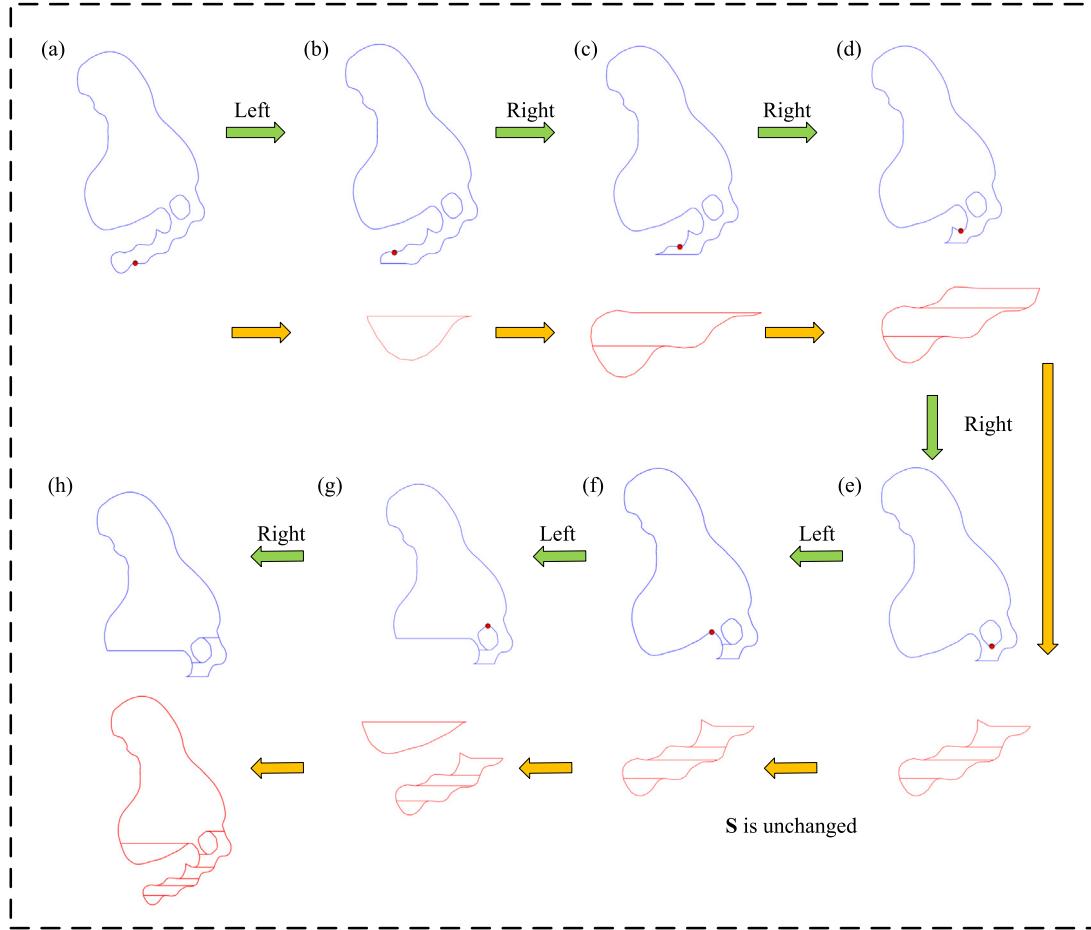
$$\arg \min_{\mathbf{Q}} f(\mathbf{S}) \quad (3)$$

In step 7, `get_subregions()` takes out the subregions from the queue that satisfy the condition of generating continuous paths into the set  $\mathbf{S}$ . The condition is: if the subregion has intersection points with the scan line, there can only be two intersection points.

In step 14–20, we decide the state of  $\mathbf{t}[0]$  by comparing the size of  $f(L)$  and  $f(R)$ , and store the subregion generated by  $q$  in the queue, and then remove  $\mathbf{t}[0]$  from  $\mathbf{t}$ . Note that when  $f(L) = f(R) = 0$ , it is decided that the state of  $\mathbf{t}[0]$  is 0, because in this case, whether or not the geometry decomposition is performed at  $t[0]$ , it does not change the number of intersection points of  $q$  with its scan lines; in other words, the  $\mathbf{t}[0]$  does not affect whether  $\mathbf{R}$  can generate a continuous path. When  $f(L) = f(R) \neq 0$ , either  $L$  or  $R$  is selected and stored in the queue.

#### 4.3. Path connection

After completing the geometry decomposition, we fill all subregions in  $\mathbf{S}$  according to the described fill pattern, defining the fill path of  $S_i$  as  $C_i$ .  $\mathbf{S} = [S_1, S_2 \dots, S_n]$ ,  $\mathbf{C} = [C_1, C_2 \dots, C_n]$ .  $\mathbf{C}$  are shown in the left side of Fig. 7. Let the set of line segments composing  $C_i$  and  $C_j$  be  $seg_i$  and  $seg_j$ , and pick the horizontal line segments among  $seg_i$  to get the set of horizontal line segments  $hseg_i$ , pick the horizontal line segments among  $seg_j$  to get the set of horizontal line segments  $hseg_j$ . Define  $o_x(x, y) = 1$  if line segment  $x$  and line segment  $y$  have overlapping parts on the x-axis. Define  $d_y(x, y)$  to return the distance between line segment  $x$  and line segment  $y$  on the y-axis. Then



**Fig. 5.** Illustration of the algorithm for geometry decomposition. (a): Subregion in *queue* after step 5; (b): After a left geometry decomposition, blue solid lines are the subregions in *queue* and red solid lines indicate subregions in *S* after step 7, red point is  $t[0]$ ; (c): After a right geometry decomposition, subregions in *queue*, subregions in *S* after step 7 and the red point is  $t[0]$ ; (d) (e) (f) (g): Represent the same content as above subfigures; (h): After a right geometry decomposition, subregions in *queue* before step 7, subregions in *S* after step 7. All subregions are taken out from *queue* into *S*, so after step 9, the algorithm will end. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$\beta_i$  and  $\beta_j$  are the lines belonging to  $hseg_i$  and  $hseg_j$  provided  $(\forall x \in hseg_i) (\forall y \in hseg_j) (d_y(x, y) \geq d_y(\beta_i, \beta_j))$ , s.t.  $o_x(x, y) = 1$ ,  $o_x(\beta_i, \beta_j) = 1$ . Generate  $\zeta_i$  by connecting paths within  $C$  starting from  $S_1$  and  $C_1$ . Select  $C_i$  connected to  $C_1$  by Eq. (4). The paths are connected to each other as shown in Fig. 6, assuming that  $C_1$  is connected to  $C_2$ .

$$\min_i d_y(\beta_1, \beta_i), i \neq 1 \quad (4)$$

The whole connection process is shown in Fig. 7, and finally  $\zeta_i$  is obtained.

#### 4.4. Toolpath optimization

Rotating  $\zeta_i$  by  $\alpha$ , it is not difficult to find that there are many uneven spacings in  $\zeta_i$ , as shown in the right side of Fig. 7. In this paper, an objective function is established to optimize the toolpath, firstly,  $\zeta_i$  is interpolated to obtain a series of points  $P_1, P_2, \dots, P_n$  denotes  $\zeta_i$ . Let  $b$  be the boundary constraint  $\partial b$ , note that  $\partial b$  is also a sequence of points, and deposit  $P_1, P_2, \dots, P_n$  and  $\partial b$  into the KD-tree. Our aim is to make the  $P_1, P_2, \dots, P_n$  slightly offset, i.e.,  $P_i^{n+1} = P_i^n + \hat{d}_i$ , so as to obtain a smooth and uniformly spaced toolpath, where  $P_i^n$  is the  $P_i$  after  $n$  iterations. For the point  $P_i$  in  $P_1, P_2, \dots, P_n$ , generate a point set consisting of all points with  $P_i$  as the center and  $3w$  as the search radius and recover the topology of the point set.

Define:

$$\mathbf{d}_i = \frac{\mathbf{P}_{i+1}\mathbf{P}_i}{|\mathbf{P}_{i+1}\mathbf{P}_i|} + \frac{\mathbf{P}_{i-1}\mathbf{P}_i}{|\mathbf{P}_{i-1}\mathbf{P}_i|}, \mathbf{d}_i^* = -\mathbf{d}_i \quad (5)$$

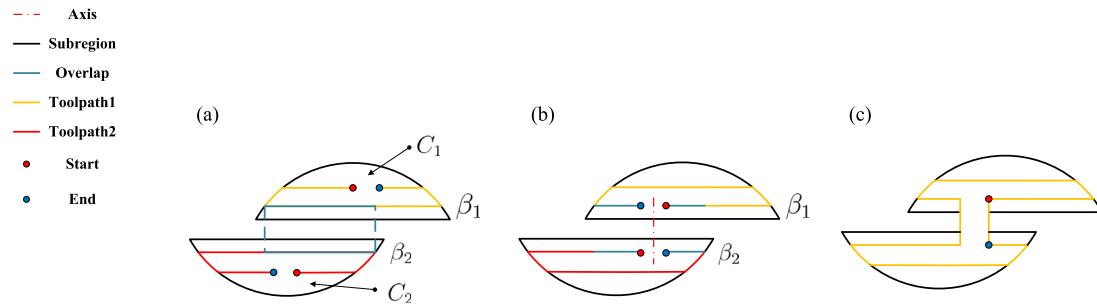
Let  $\overrightarrow{\mathbf{P}_i\mathbf{d}_i}$  be the ray starting from  $\mathbf{P}_i$  and following the direction of  $\mathbf{d}_i$ . Among the intersections of  $\overrightarrow{\mathbf{P}_i\mathbf{d}_i}$  with the topology of the point set, the closest intersection to  $\mathbf{P}_i$  is selected as  $\mathbf{P}_i^o$ . Let  $\overrightarrow{\mathbf{P}_i\mathbf{d}_i^*}$  be the ray starting from  $\mathbf{P}_i$  and following the direction of  $\mathbf{d}_i^*$ . Among the intersections of  $\overrightarrow{\mathbf{P}_i\mathbf{d}_i^*}$  with the topology of the point set, the closest intersection to  $\mathbf{P}_i$  is selected as  $\mathbf{P}_i^{*o}$ . The whole process is shown in Fig. 8, then the objective function is:

$$\text{minimize } f = \sum_{i=1}^n (|\mathbf{P}_i\mathbf{P}_i^o| - w)^2 + (|\mathbf{P}_i\mathbf{P}_i^{*o}| - w)^2 \quad (6)$$

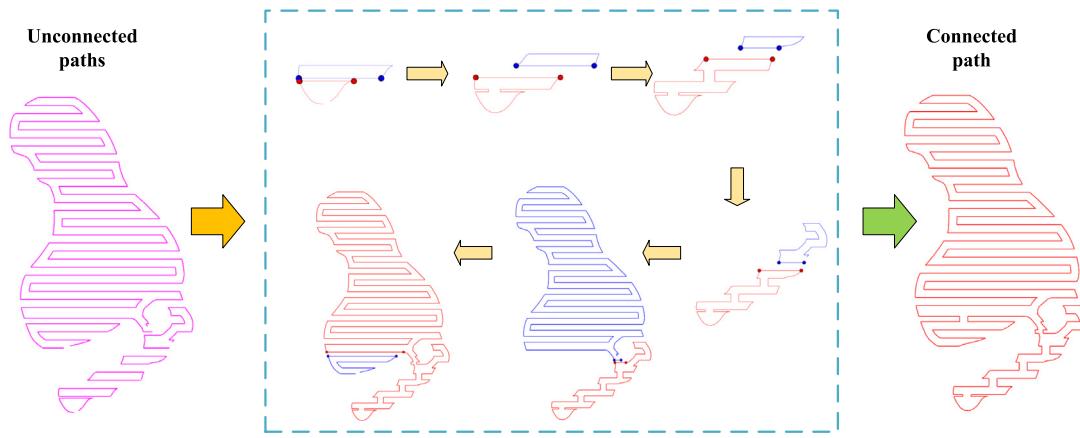
Use the gradient descent method to solve the objective function:

$$\mathbf{P}_i^{n+1} = \mathbf{P}_i^n - \alpha \frac{\partial}{\partial \mathbf{P}_i^n} f(\mathbf{P}_0^n, \mathbf{P}_1^n, \dots, \mathbf{P}_n^n) \quad (7)$$

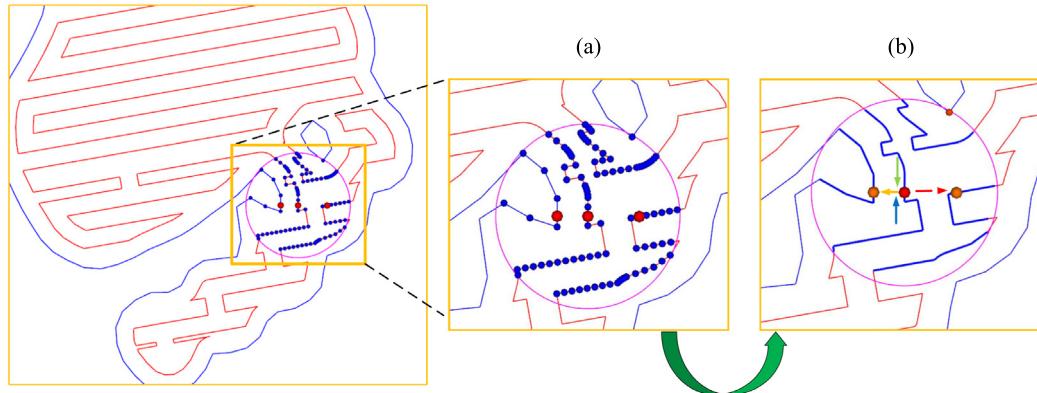
Set the step size  $\alpha$  as 0.01, set the number of iterations  $N = 15$ , and then smooth  $\zeta_i$  using Eqs. (1) and (2). Finally, connect  $\zeta_i$  with  $b$  to generate a 2D continuous path  $\gamma$ . Try to select the points with low curvature for connection and do self-intersection judgment, and if the selected connection point makes



**Fig. 6.** Schematic diagram of local continuous path connection. (a): Schematic diagram of  $C_1$  and  $C_2$ ,  $S_1$  and  $S_2$ ,  $\beta_1$  and  $\beta_2$  and their overlapping parts.  $C_1$  and  $C_2$  are open paths; (b): Find the axes of the overlapping part of  $\beta_1$  and  $\beta_2$ , insert two points with interval  $w$  and symmetry about the axes in the overlapping part of each of  $\beta_1$  and  $\beta_2$ , and move the sequence of points of the path so that these two points become the starting and ending points of the path, as shown in the figure; (c): Connect the endpoint of  $C_1$  with the start point of  $C_2$  to generate the connected path and its start and endpoints as shown in the figure.



**Fig. 7.** Generation of connected toolpath. The magenta path represents the paths in  $\mathbf{C}$ , the red path is  $C_1$ , and each connection replaces the original  $C_1$  with the connected path and repeats Eq. (4) to find the next path to connect. The blue path is the  $C_i$  chosen at the time of connection, and the horizontal lines represented by the red endpoints and the blue endpoints are  $\beta_1$  and  $\beta_i$ . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

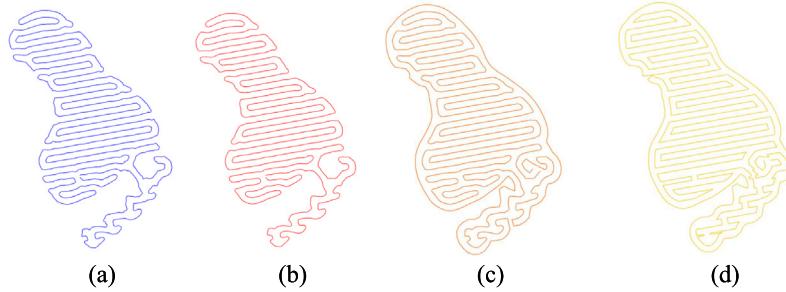


**Fig. 8.** Illustration of the toolpath optimization algorithm. The red solid line is  $\zeta_i$  and the blue solid line is  $\partial b$ ; (a): red points do left to right in order for  $\mathbf{P}_i^{*0}$ ,  $\mathbf{P}_i^o$ ,  $\mathbf{P}_i^o$ . The blue points are the set of points of  $\mathbf{P}_i$ , the magenta circle is the search range; (b): The red points represent  $\mathbf{P}_i$ , the blue solid line, and the orange smaller points are the topological line segments and the topological points recovered from the point set. The blue and green arrows indicate  $\frac{\mathbf{P}_{i-1}\mathbf{P}_i}{|\mathbf{P}_{i-1}\mathbf{P}_i|}$  and  $\frac{\mathbf{P}_{i+1}\mathbf{P}_i}{|\mathbf{P}_{i+1}\mathbf{P}_i|}$ . The red arrow indicates  $\mathbf{d}_i$ , the yellow arrow indicates  $\mathbf{d}_i^*$ . The larger orange points from left to right indicate  $\mathbf{P}_i^{*0}$ ,  $\mathbf{P}_i^o$  respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$\gamma$  appear self-intersection, the connection point is reselected, and the results are shown in Fig. 9.

To speed up the optimization process, we have modified the implementation of the above algorithm to obtain results in parallel with matrix operations. In the following we only explain how to get  $\mathbf{P}_i^o$ . The same applies to  $\mathbf{P}_i^{*0}$ . Write  $\mathbf{P}_i \mathbf{d}_i$  and the segments

$\mathbf{AB}_{i,j}$  near the  $\mathbf{P}_i$  in the form of the following parametric Eq. (8). To be able to find the nearest neighbors of point  $\mathbf{P}_i$  in GPU quickly, we use the ball query proposed in [39]. Set the search radius, the number of nearest neighbors of each point is set to  $m$ , these points all form line segments with their precursors, padding is performed on the unaligned nearest neighbors. Tensors  $\mathbf{A} \in$



**Fig. 9.** Effect of toolpath optimization. (a): The blue solid line is the result of  $\xi_i$  optimization; (b): The red solid line is the result of  $\xi_i$  optimization with smoothing; (c): Orange solid line is the  $\gamma$  generated by connecting the optimized and smoothed  $\xi_i$  to  $b$ ; (d): Yellow solid line is the  $\gamma$  generated by  $\xi_i$  connection directly with  $b$ . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$\mathbb{R}^{n \times m \times 3}$  and  $\mathbf{B} \in \mathbb{R}^{n \times m \times 3}$  denote the start and the end of the segments. In order not to add new symbols, we let the matrix  $\mathbf{P} \in \mathbb{R}^{n \times 3}$  denote all points on the path.  $\hat{\mathbf{D}} \in \mathbb{R}^{n \times 3}$  denotes all  $\mathbf{d}_i$ , which can be get as Eq. (9).

$$\begin{aligned} \overrightarrow{\mathbf{P}_i \mathbf{d}_i}(t) &= \mathbf{P}_i + t\mathbf{d}_i \\ \mathbf{A}\mathbf{B}_{i,j}(s) &= \mathbf{A}_{i,j} + s(\mathbf{B}_{i,j} - \mathbf{A}_{i,j}) \\ \mathbf{D} &= \text{normalized}(\text{roll}(\mathbf{P}, -1) - \mathbf{P}) + \text{normalized}(\text{roll}(\mathbf{P}, +1) - \mathbf{P}) \\ \hat{\mathbf{D}} &= \text{normalized}(\mathbf{D}) \end{aligned} \quad (9)$$

Reshape  $\mathbf{P}, \mathbf{D} \in \mathbb{R}^{n \times 1 \times 3}$ ,  $t$  and  $s$  can be calculated as Eq. (10) [40].  $t \in \mathbb{R}^{n \times m}$  and  $s \in \mathbb{R}^{n \times m}$ , Get the index of the elements in  $t$  less than 0,  $s$  greater than 1 or less than 0, and mask  $t$ , Select the

#### Algorithm 1 Polygon Decomposition

```

Input: the 2D connected region R
Output: the subregions set S
1: queue =  $\emptyset$ 
2: S =  $\emptyset$ 
3: t = get the turn vertexes of R
4: t are arranged in ascending order about the y value
5: queue.append(R)
6: while 1 do
7:   S  $\leftarrow$  get subregions(queue)
8:   if queue.size() == 0 then
9:     break
10:   end if
11:   for subregion in queue do
12:     if t[0] on subregion then
13:       q = queue.pop(subregion)
14:     end if
15:   end for
16:   L = the subregions after left decomposition in t[0]
17:   R = the subregions after right decomposition in t[0]
18:   define the f(X) returns the inverse number of scan lines of
      the subregion with the fewest number of scan line in X, X is a
      set of subregions
19:   if f(L) == f(R) == 0 then
20:     queue  $\leftarrow$  q
21:   else
22:     queue  $\leftarrow$  minX f(X), X = L, R
23:   end if
24:   t.pop(0)
25: end while

```

smallest value in each row of  $t$  to get  $t'$ , and substitute  $t'$  into the parametric equation of Eq. (8) to obtain  $\mathbf{P}_i^o$ .

$$\begin{aligned} t &= \det((\mathbf{A} - \mathbf{P}), (\mathbf{B} - \mathbf{A}), (\mathbf{B} - \mathbf{A}) \times \hat{\mathbf{D}})/|(\mathbf{B} - \mathbf{A}) \times \hat{\mathbf{D}}|^2 \\ s &= \det((\mathbf{A} - \mathbf{P}), \hat{\mathbf{D}}, (\mathbf{B} - \mathbf{A}) \times \hat{\mathbf{D}})/|(\mathbf{B} - \mathbf{A}) \times \hat{\mathbf{D}}|^2 \end{aligned} \quad (10)$$

#### 5. Idle stroke length optimization

##### 5.1. Printing sequence optimization

By slicing the input mesh model by the predetermined layer thickness  $H$ , a series of 2D connected regions with length  $n$  is created. The 2D connected region is identified by the notation  $r_{ij}$ , where  $i$  stands for the number of layers in which the connected region and  $j$  for the fact that it is the  $j$ th connected region of the  $i$ th layer. The regions in the sequence are sorted in queue **D** in ascending order of size  $i$ . The order of the connected regions in the same layer is random.

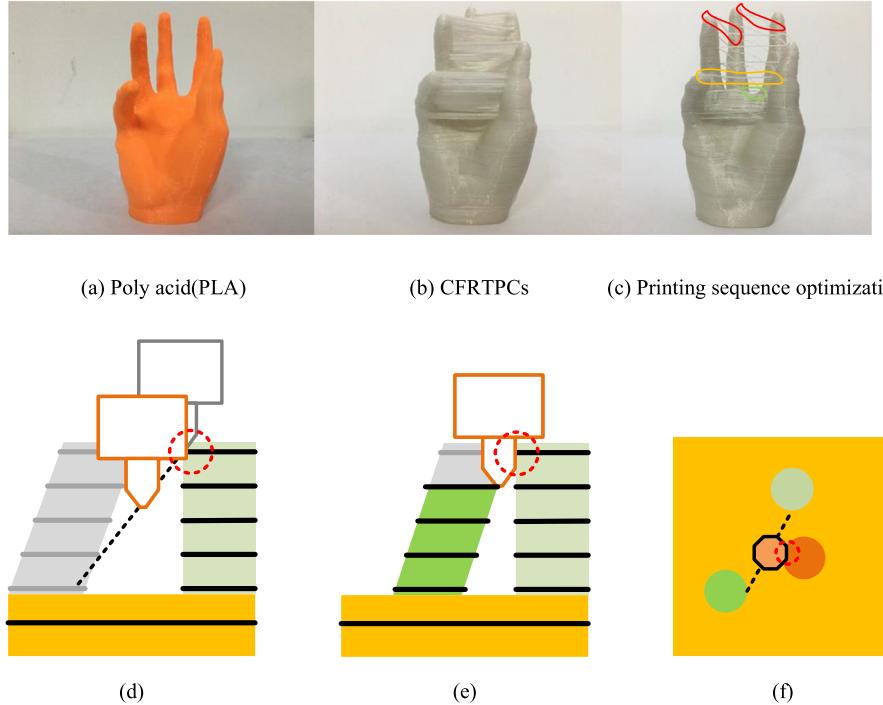
Our work at this stage is to sort the sequence of 2D connected regions considering interference to get a set of printing order  $\mathbf{r} = [r_{1,1}, \dots, r_{k,i}, \dots, r_{m,n}]$  to reduce the idle stroke. Fig. 10 demonstrates the effect of idle strokes on the printing part with pure fused thermoplastic filament and CFRTPCs. Based on the above description, we can describe the problem as Eq. (11):

$$\begin{aligned} \arg \min_{\mathbf{r}} \sum_{i=1}^N d_i(r_k, r_{k+1}) \\ \text{s.t. } \begin{cases} g_x(r_k, r_{k+1}) \leq T_x \\ g_y(r_k, r_{k+h}) \leq T_y \end{cases} \end{aligned} \quad (11)$$

The optimization objective of Eq. (11) [41] is to minimize the sum of the distances  $d_i(r_k, r_{k+1})$  between two connected regions, where  $k$  denotes the sequential index of the connected region  $r_{ij}$  in the print sequence  $\mathbf{r}$ .  $g_x(r_k, r_{k+1})$  denotes the horizontal distance between two adjacent connected regions, and  $g_y(r_k, r_{k+h})$  denotes the height difference between the  $h$  consecutively printed connected regions and other adjacent branches.  $T_x$  and  $T_y$  are the horizontal and height thresholds, respectively, and are related to the hardware construction of the 3D printer itself.

We have changed the printer head from a straightforward bottom-up layer-by-layer movement to a jump in space in the desired order to be able to reduce idle stroke. In order to prevent collisions between the printer head and the printed part, we must plan the printer head's movement. The following three interference instances are described in this paper:

1. Horizontal interference: the printer head collides with the already printed  $r_{ik}$  when printing  $r_{ij}$  ( $k \neq j$ ), as shown in



**Fig. 10.** The effect of the idle stroke and interference on printed parts. (a): Structure printed using pure Poly acid(PLA) materials; (b): Structures printed by conventional printing using CFRTPCs; (c): Structures printed with CFRTPCs after print sequence optimization. Moving the printer head between the branches frequently causes interference; the red solid circle out of the part makes it easy to occur high interference, the yellow solid circle out of the part makes it easy to occur path interference, and the green solid circle out of the part makes it easy to occur horizontal interference; (d): Height interference schematic; (e): Horizontal interference schematic; (f): Path interference schematic. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

- Fig. 10(e). The black dashed line indicates the idle stroke, and the part circled by red circles is where the interference occurs. The black printer head indicates the starting position of the printer head in the idle stroke, and the orange printer head indicates the position of the printer head in motion. A colored portion indicates that the part has been printed, and a transparent portion indicates that the part has not yet been printed. Solid black lines indicate 2D contours.
2. Height interference: after the print of  $r_{ij}$  is complete, the printer head is in the process of jumping to the  $r_{kj}$  that has not yet been printed, and the printer head collides with the branch where  $r_{ij}$  located, as shown in Fig. 10(d).
  3. Path interference: the printer head collides with the already printed part during the idle stroke when it jumps to other branches, as shown in Fig. 10(f). The green circle indicates the branch where the idle stroke begins, the gray-green circle indicates the branch the printer head will jump to, and the red circle is the branch that collides with the printer head.

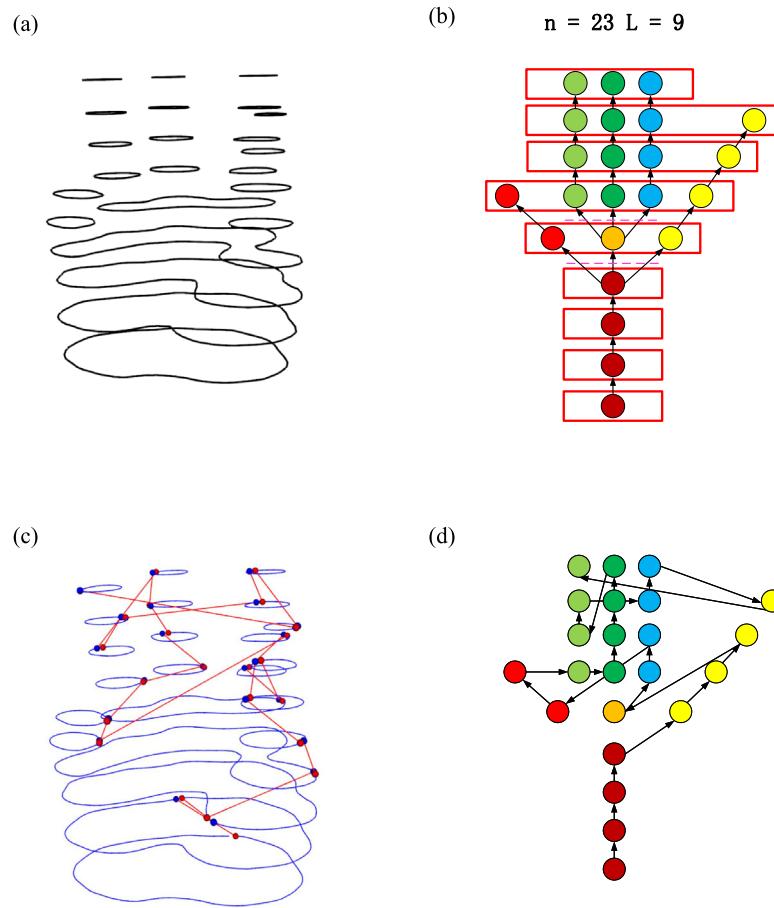
In this section, the print sequence optimization algorithm focuses on solving the horizontal and height interference problems, and the section on path interference will be explained in Section 5.3. We give an example to facilitate our illustration of the algorithm. Fig. 11(a) shows the sequence of 2D connected regions generated after slicing with layer thickness  $H = 2$  mm, and the sequence length is 23. Treating each 2D connected region as a node, the edges between nodes indicate the support relationship between regions. If the overlap area between region  $r_{ij}$  and region  $r_{i+1j}$  is 40 percent larger, we consider that  $r_{ij}$  supports  $r_{i+1j}$  and generate an edge between nodes, as shown in Fig. 11(b). By deleting two or more edges pointing to the same node in the

graph, the resulting connected components represent the individual branches of the print structure, with nodes of different colors representing different branches, and nodes in the same red box indicating that they are in the same layer.

We deposit these 2D connected regions into the queue **D** according to the rules described previously, and pick all the outgoing  $r_{ij}$  into **r**. Start with the element at the head of the queue and before it is taken out of the queue, predetermine whether horizontal or height interference will occur while the printer head is printing it. i.e., determine whether the horizontal distance between  $D_0$  and other elements in the queue that are on the same layer as  $D_0$  is greater than  $T_x$  or whether the difference between the layer in which  $D_0$  is located and the layer  $L$  with the smallest layer height in the queue is greater than  $T_y$ . If one of the above two conditions is satisfied, the element in the L-layer in **D** nearest to  $r_{-1}$  is moved to the head of the queue. If the above conditions are not satisfied,  $D_0$  is taken out of the queue, and the elements supported by  $D_0$  are searched in **D**. These elements supported by  $D_0$  are moved to the head of the queue and the original order between them is maintained if there is no element supported by  $D_0$ , the element nearest to  $r_{-1}$  among the elements of the L-layer in **D** is moved to the head of the queue. Let  $T_x = 2$  and  $T_y = 2$ . The results are shown in Fig. 11(c).

## 5.2. Selecting connecting points

We first generate the print sequence of the connected regions after slicing as described in Section 5.1 and generate the continuous paths of the connected region as described in Section 4 to obtain a counterclockwise sequence of 2D continuous paths  $\gamma = [\gamma_1, \gamma_2, \dots, \gamma_n]$ . Our next job is to pick the appropriate connecting points (start and endpoints) between the paths so



**Fig. 11.** Diagram of the print sequence optimization algorithm. (a): Set of slice contours, black solid lines indicate slice contours; (b): The graph representing the support relationship; (c):The result of the print sequence optimization, the blue solid lines indicate the slice contours, the red solid lines indicate the idle strokes, the red points indicate the starts of the idle strokes and the blue points indicate the ends of the idle strokes; (d): A graph indicating the order of printing, with arrows pointing to indicate the order of printing. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

that the entire printing process has the lowest idle stroke. To make calculations easier,  $\gamma_i$  is open path. This allows us to change the starting point of the path by translating the sequence of points without destroying the shape of the path itself. Once the starting point of the path is known, the ending point of the path also follows, as shown in Fig. 12. To improve the computational efficiency and make the genetic algorithm converge faster,  $b_i$  of length  $n$  is sampled to get  $m$  sampled points. Let the sampled point sequence be  $b_i^s$ , and choose the point in  $\gamma_i$  that lies on  $b_i^s$  as the connection point. Let  $P_{ij}$  be the  $j$ th point of  $\gamma_i$  and the starting point and the length of  $\gamma_i$  is  $l$ .

$$\arg \min \sum_{j=1,2,\dots,l}^N \|P_{i,j-1}, P_{i+1,j}\|_2 \quad (12)$$

s.t.  $P_{ij} \in b_i^s$

We solve this optimization problem using a genetic algorithm, which simulates the natural selection process in the biological world, to generate a high-quality solution to the optimization problem. In genetic algorithms, the population of candidate solutions (individuals) to the optimization problem evolves toward a better solution by three operations on individual chromosomes, namely selection  $O_s$ , crossover  $O_c$ , and mutation  $O_m$ , with crossover rate  $Q_c$  and mutation rate  $Q_m$ . First, we randomly select a  $P_{ij}$  for  $\gamma_i$  subject to the constraints, and encode  $j$  of these starting

points into a chromosome, as shown in Fig. 13(a), and generate an initial population  $X^0$  with population number  $M$ . The population at generation  $t$  is denoted as  $X^t = [x_1^t, x_2^t, \dots, x_M^t]$ , such that  $x_{ij}^t$  is the  $j$ th gene on chromosome  $x_i^t$  at generation  $t$ . Let  $F(x)$  be the inverse of the  $f(x)$  in Eq. (12), and the individual fitness function  $F'(x)$  is shown in Eq. (13), and  $\text{eps}$  is a very small value.

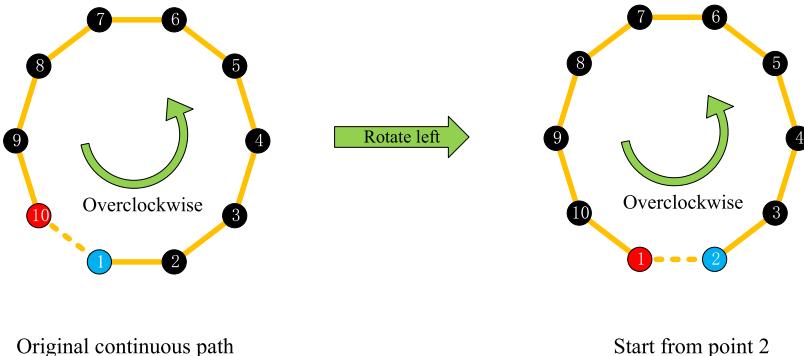
$$F'(x^t) = F(x^t) - \min(F(x_i^t)) + \text{eps}, i = 1, 2, \dots, M \quad (13)$$

Chromosomes in  $X^t$  with the top one percent fitness are preferentially inherited by  $X^{t+1}$ , and the remaining chromosomes are inherited to the next generation with a certain probability according to Eq. (14).

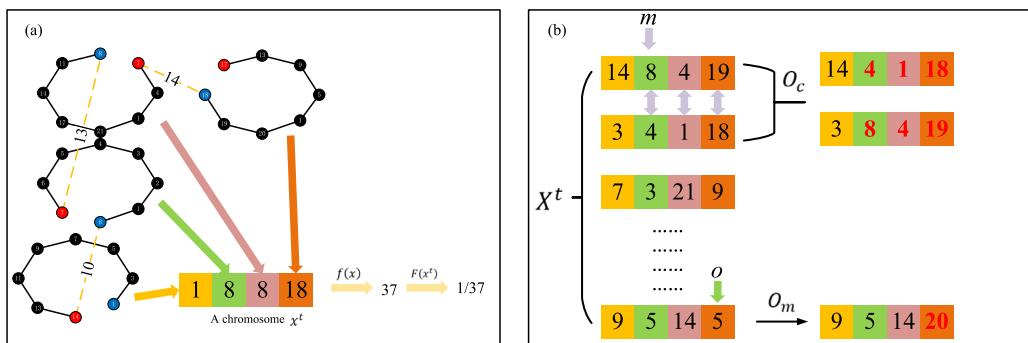
$$P(x_z^t) = \frac{F'(x_z^t)}{\sum_{i=1}^n F'(x_i^t)} \quad (14)$$

The crossover operation is performed as follows. In the population, two adjacent chromosomes are selected sequentially as two parental individuals and a random number is generated between 0 and 1 for each selection. If the number is less than  $Q_c$ , a random position  $m$  is selected between the two parental individuals, and the genes after this position are exchanged, as shown in Fig. 13(b).

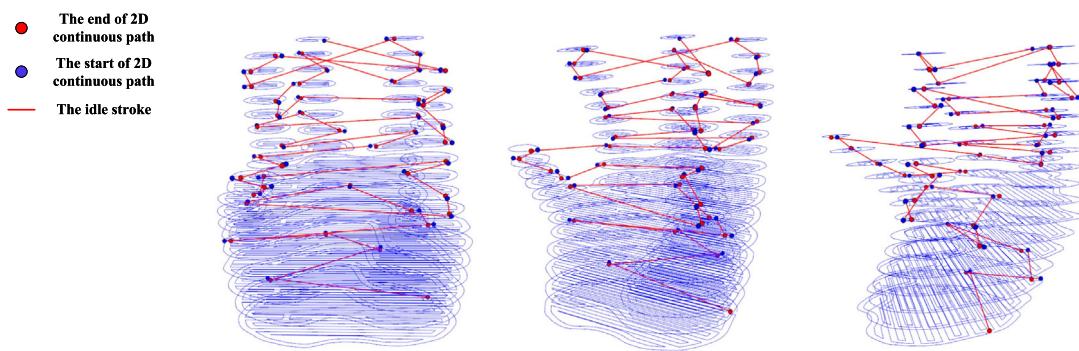
This mutation operation acts on an individual chromosome. A random number in the interval  $[0, 1]$  is generated with uniform



**Fig. 12.** Schematic diagram for selecting the starting and ending points of a 2D continuous path. The points in the figure represent the sequence of points that form a continuous path, blue points represent the starting point of the sequence, red points represent the endpoint of the sequence, and the index of the points is represented by a number. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



**Fig. 13.** Schematic diagram of the encoding and operation of the genetic algorithm. (a): The figure depicts the process of generating a chromosome, where the yellow dashed line indicates the idle stroke between paths and the number indicates the length of the idle stroke, the black solid line indicates the 2D continuous path  $\gamma_i$ , the points on the path in the figure are in  $b_i^s$ , the number in the point represents its index in the  $\gamma_i$ , the blue point indicates the starting point of  $\gamma_i$ ; (b): Schematic diagram of crossover and mutation operations. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

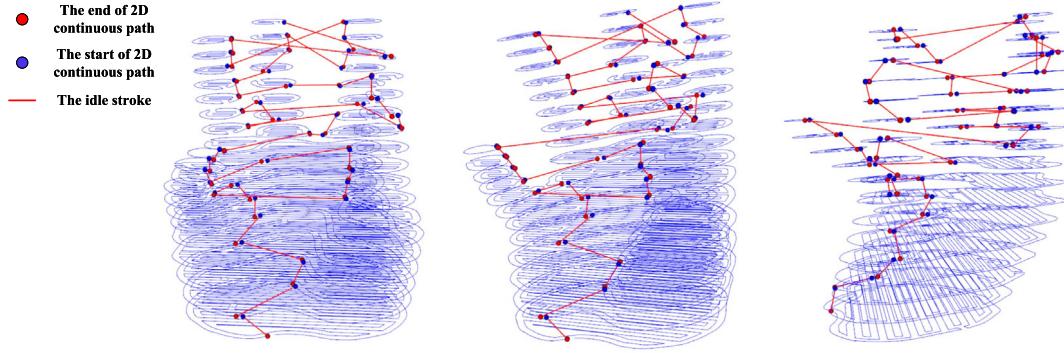


**Fig. 14.** Global continuous path without optimization.

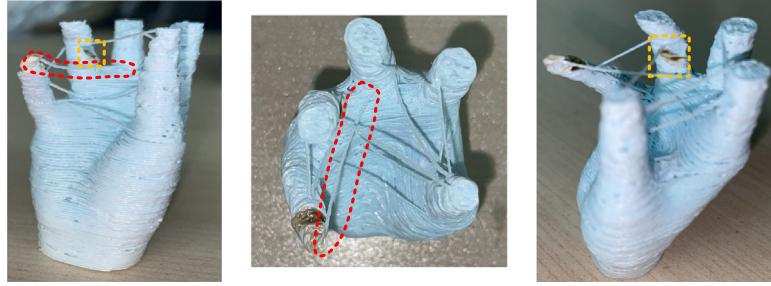
probability. If the number  $> Q_m$ , no mutation is applied. Otherwise, we randomly select a position  $o$  in the chromosome. First, we get the  $x_{io}^t$ th point in  $\gamma_i$ , then we get the index  $i_b$  of this point in  $b_i^s$ , and finally  $x_{io}^t$  varies as the index of the penultimate  $i_b$ th point of  $b_i^s$  in  $\gamma_i$ , as shown in Fig. 13(b).

To facilitate the observation of the global continuous paths before and after the optimization of the idle stroke, we sliced

the mesh model by 1 mm layer thickness. We set  $M = 500$ , the number of iterations  $N = 500$ ,  $Q_c = 0.25$ ,  $Q_m = 0.3$ ,  $\text{eps} = 1\text{e}-8$ , the number of sampling points  $m = 32$ , and the filling direction of all paths in  $\gamma$  is  $0^\circ$ . Fig. 14 shows the result without optimization, the total length of the path is 5026.95 mm and the length of the idle stroke is 422.21 mm. Fig. 15 shows the result of optimizing the idle stroke using genetic algorithm, the length of the path is



**Fig. 15.** Optimized global continuous path.



**Fig. 16.** Illustration of the flaw raised by path interference. The part circled by the red dotted line indicates the occurrence of path interference of the idle stroke. The part circled by the yellow dotted line indicates the defect caused by the collision between the printed part and the printer head.  $H = 0.4$  mm,  $T_y = 9$  (layer),  $T_x = 0.4$  mm,  $\alpha = 0^\circ$ . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4892.49 mm and the idle stroke is 287.75 mm. The idle stroke is reduced by 30%.

### 5.3. Using the SAT and RRT\* to avoid interference

After the above steps, we have generated a global continuous toolpath with controllable local direction. In this section, we solve the path interference problem that occurred during the printing process mentioned in Section 5.1. In the previous optimization, we neglected to take into account the interference collision between the printer head and the printed part during the movement of the idle stroke, as evidenced by the flaws in the sample part used for illustration in Fig. 16. Before this section, the idle stroke is just a line segment generated from two endpoints.

In this paper, we use the SAT to quickly detect collisions between the printer head and printed part and RRT\* [42] algorithm to adjust the idle stroke globally to avoid the collision. Path interference only occurs when the distance between the start and end points of the idle stroke on the z-axis is greater than one layer thickness. We define the start point as *root* and the end point as *target*. The point sequence  $\mathbf{E} = [E_1, E_2, \dots, E_n]$  denotes the adjusted idle stroke. The printed contours list is defined as  $\mathbf{M}$ . The sample space is defined as  $\mathbf{S}$ . We define the sampling space as a rectangular prism with a range on the z-axis of  $(\min(\text{target.z}, \text{root.z}), (\text{target.z}, \text{root.z}) + h^*)$ , and on the plane as a rectangle that can enclose all elements in  $\mathbf{M}$  (projected onto a plane). Here,  $h^*$  is a value less than  $H$  to prevent collisions between the printing

head and the plane of the starting position. In this paper,  $h^* = 0.05$  mm. The pseudocode is as follows.

---

#### Algorithm 2 Avoid Collision

---

**Input:**  $\mathbf{E}$ , *root*, *target*,  $\mathbf{M}$ ,  $r$ ,  $\mathbf{S}$   
**Output:**  $\mathbf{E}$

```

1:  $t.init()$ 
2:  $\mathbf{E} = \emptyset$ 
3: while 1 do
4:    $x_{rand} = Sample(\mathbf{S})$ 
5:    $x_{near} = Near(t, x_{rand})$ 
6:   if  $dis(x_{rand}, target) <= StepSize$  then
7:      $x_{new} = target$ 
8:   else
9:      $x_{new} = Steer(x_{rand}, x_{near}, StepSize)$ 
10:  end if
11:   $TrajectoryPolygon = OutwardOffset(x_{new}, x_{near}, r)$ 
12:  if  $SATDetectCollision(M, x_{new}, x_{near}, TrajectoryPolygon) \neq 1$ 
    then
13:     $X_{near} = NearC(t, x_{new})$ 
14:     $x_{min} = ChooseParent(x_{near}, x_{new}, X_{near})$ 
15:     $t.AddNodeEdge(x_{min}, x_{new})$ 
16:     $t.Rewire(X_{near})$ 
17:    if  $x_{new} == target$  then
18:      break
19:    end if
20:  end if
21: end while

```

---

In step 1, we initialize a random search tree  $t$  with *root* as the root node of  $t$ . Each node has three member variables: parent,



**Fig. 17.** Effect of the global path in each direction. We can find that the idle stroke has been changed from a straightforward straight line to a polyline to prevent path interference. If you have doubts about the effect of the picture display, you can further watch our attached video. (a) is the multi-view of the hand model; (b) is the multi-view of the Bunny; (c) is the multi-view of the Female; (d) is the multi-view of the Hulk.

child, and cost. Where cost indicates the total length of the path from the root node to the current node.

In step 4–10, Randomly sample a point  $x_{rand}$  from the sampling space, compute the nearest node  $x_{near}$  from  $x_{rand}$  in  $t$ , and move one  $StepSize$  along the direction from  $x_{near}$  to  $x_{rand}$  to get  $x_{new}$ . If the distance between  $x_{rand}$  and the target is less than  $StepSize$ , then the target is taken as  $x_{new}$ .

In steps 11–12, Get the line segment from  $x_{new}$  to  $x_{near}$ , project the line segment to the xy plane and offset it outward by a distance of  $r$  to obtain  $TrajectoryPolygon$ . Take all contours from  $M$  with z-axis greater than  $\min(x_{new}.z, x_{near}.z)$ , project all contours to the xy plane, perform the Boolean sum operation, and then perform convex wrapping. The SAT algorithm is used to quickly determine whether there is a collision between  $TrajectoryPolygon$  and the convex packet, and if there is a collision,  $x_{new}$  is rounded off. Where we modeled the nozzle as a cylinder and  $r$  as its radius.

In steps 12–16, These steps are used to optimize the speed of convergence of the RRT\* algorithm and to reduce the length of the paths. Using  $x_{new}$  as a round point, search for a set of nodes  $X_{near}$  within a set radius in  $t$ . We select the parent node with a lower cost for  $x_{new}$ . Then, we determine which node in  $X_{near}$  has a

smaller cost than the one with  $x_{new}$  as its parent, and if the latter is smaller,  $x_{new}$  is the parent of the current node and updates its cost.

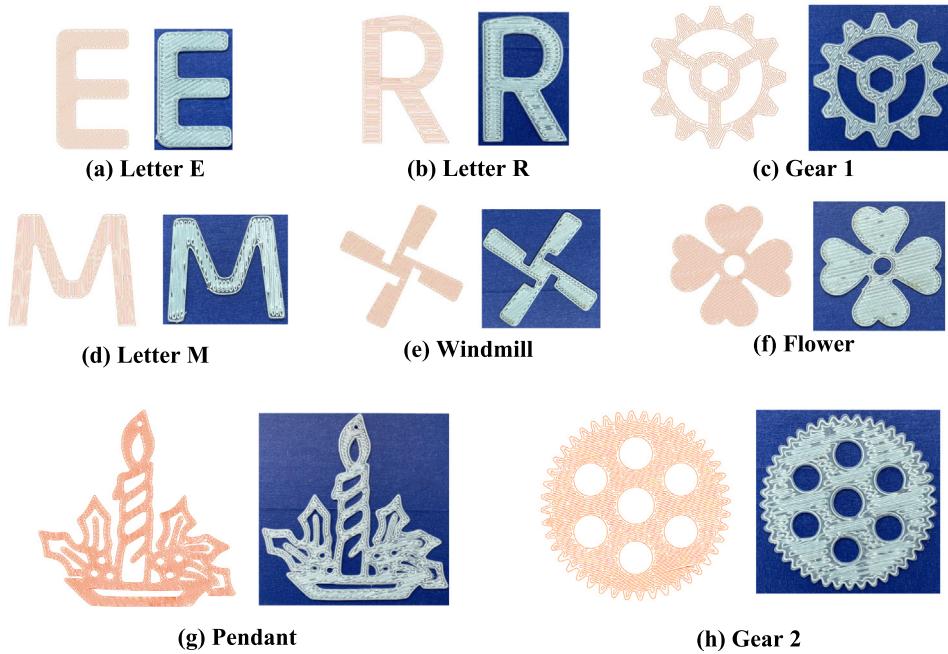
The final results are shown in Fig. 17.  $H = 0.2$  mm,  $T_y = 18$ (layer),  $T_x = 3.6$  mm,  $M = 300$ ,  $N = 300$ ,  $Q_c = 0.25$ ,  $Q_m = 0.3$ ,  $eps = 1e-8$ , the number of sampling points  $m = 32$ , and  $\alpha = 15^\circ$ .

## 6. Evaluation

In this section, we provide three examples to evaluate the method and its applicability in different application scenes. The implementation source code for the entire process is available on [github](#).

First, we selected a set of samples with complex profiles, filled by using a continuous toolpath and printed with continuous fiber reinforcement to test the effect of generating a toolpath, as well as the controllability of local infill directions.

In order to demonstrate that our method can be combined with the method of stress field partitioning to generate a global continuous toolpath with controllable local direction, we first



**Fig. 18.** Continuous fiber printed parts and their toolpaths.

select a sample from the Zhang et al. [35] paper and improve their method by using continuous fibers by the method described in this paper. By doing this, we are able to further improve the strength of printed parts by utilizing continuous fibers.

Finally, we show global continuous fiber printed parts with different height thresholds. It is illustrated that the method in this paper plays an important role in optimizing the idle stroke of the printed parts and can be tuned according to the actual printing platform.

The experimental platform is a continuous fiber-reinforced 3D printer with the pre-impregnation printing method, which was built based on the BM10-Z-25 forming  $250 \times 250 \times 350$  mm model 3D printer. The reinforced fiber used is 200D polyester and the matrix is transparent PLA with 1.75 mm diameter. The diameter of the nozzle is 0.6 mm.

#### 6.1. Continuous fiber printed part with arbitrary deposition direction

Compared to other advanced methods, our method ensures uniform filling spacing with controllable deposition direction and is suitable for continuous fibers. However, during the experiment, we found that due to the path having many corners and the co-extrusion printing mode we used, the position of the fibers at these small corners would shift during the deposition, and when the printing speed was too fast, the fibers would not even be bonded to the hotbed, thus leading to failure of print, so the maximal printing speed we used at 45 mm/min. The result is shown in Fig. 18 and the process parameters of the sample parts are shown in Table 1. We implemented the algorithm using Python 3.9. For models not marked with \*, the performance test was run on a Core AMD Ryzen 9 5900HX laptop(3.3 GHz) with 16 GB of RAM and an RTX3070 laptop GPU with 8 GB memory. Other models were measured using an Intel(R) Xeon(R) W-2245 CPU(3.9 GHZ) with 128 GB RAM and an NVIDIA RTX A6000GPU with 40 GB memory. When generating paths, the computational consumption is very sensitive to the turn vertex number T and dimensions(mm), and when optimizing paths, the computational consumption is more sensitive to the number of sampling points N of the path. The number of iterations I was chosen as 15.

**Table 1**  
Process parameters of the sample parts.

Name	Fill interval	Fill angle
Letter E	0.4 mm	30°
Letter R	0.4 mm	0°
Gear 1	0.6 mm	25°
Letter M	0.4 mm	90°
Windmill	0.4 mm	45°
Flower	0.4 mm	10°
Pendant	0.4 mm	15°
Gear 2	0.6 mm	20°

The learning rate is 0.017. In general, the greater the number of sampling points and the number of iterations, the better the visual effect of the path. The learning rate, sampling accuracy, and the number of iterations are all time-consuming but will make the path more effective. The compute cost is shown in Table 2

#### 6.2. Effect of the idle stroke optimization

The length of the nozzle we used was 6.8 mm from the bottom to the heating block after tightening. The genetic algorithm parameters of the above samples are consistent with those described in Section 5.2. Compared with the conventional printing method, our method reduces the idle strokes, decreases the post-processing difficulty of the continuous fiber printed parts, and is also applicable to pure fused filaments.

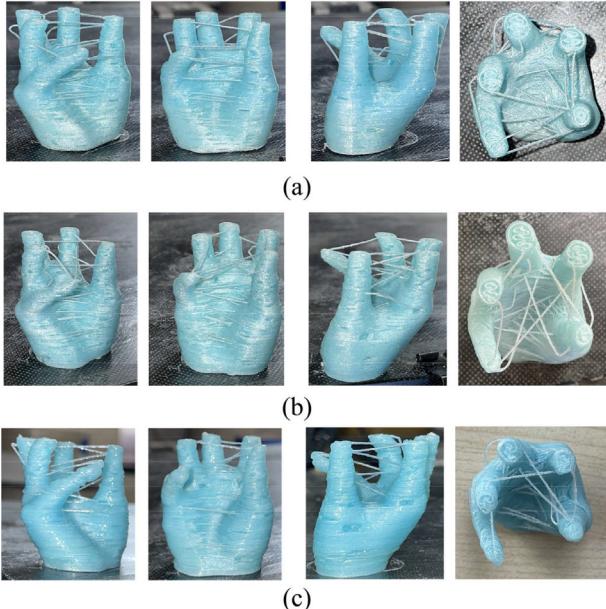
However, during the printing process, the rise of the printer head will cause the fiber at the connecting point to shift, which will create a small gap. In the experiment, we found that because the connecting point is the entrance and exit of the path and has a relatively sharp corner, which further increases the gap at the connecting point. When printing the sample with  $T_y = 3.6$  mm, we reduced the printing speed to 60% of the original, the above problems have been significantly improved, but the selection of the connecting point is also a problem that we need to further improve. The result is shown in Fig. 19 and Table 3.

**Table 2**  
Timings of different models.

Name	T	N	Dimensions	Generation cost(s)	optimized cost(s)
Letter E	3	11461	30 × 50	0.178	13
Letter R	7	9969	32 × 48	0.27	12.41
Gear 1	32	12202	52 × 52	1.50	12.38
Letter M	17	9845	40 × 45	0.61	16
Windmill	7	10910	57.3 × 57.3	0.33	12.56
Flower	18	21703	55.8 × 55.8	0.731	20.13
Gear 2*	50	15010	46 × 46	3.03	12.62
Pendant*	85	31221	90 × 90	17.81	28.02

**Table 3**  
Comparison of the idle stroke for different height thresholds.

Number	$T_y$	$T_x$	H	Total path length	The idle stroke length
(a)	18	3.6 mm	0.2 mm	97179 mm	2401 mm
(b)	24	3.6 mm	0.2 mm	97173 mm	2395 mm
(c)	30	3.6 mm	0.2 mm	97088 mm	2310 mm



**Fig. 19.** Global continuous fiber printed samples. (a): sample with multiple directions,  $T_y = 3.6$  mm, (b): sample with multiple directions,  $T_y = 4.8$  mm, (c): sample with multiple directions,  $T_y = 6.0$  mm.

### 6.3. Principal stress field-driven CFRTPCs printing test

In order to demonstrate the toolpath planning of controllable local direction, we use this method to improve the inter-layer interleaved composite enhancement path planning (IICP) method presented by Zhang et al. [35]. The local direction of the toolpath in IICP is decided by the corresponding principal stress partitions. As shown in Fig. 20, the global continuous paths as well as the actual printed samples are given.

The geometric parameters of the model used, the filling angle of each partition, the filling spacing, and the interlacing angle between layers are consistent with those described in Zhang's paper. To make the filling path more uniform we choose an offset distance of  $w/2$ .

First, we smooth the boundaries between partitions using Bezier curves to reduce the stress concentration due to sharp boundaries, as shown in Fig. 21(a). Next, we make the partitions as nodes and determine whether the nodes are reachable from node to node by determining whether there are overlapping edges between the partitions to build an undirected graph. Then

**Table 4**

Features of samples fabricated by different tool-paths-part.1.

Physical quantities	CZ(0°/90°)	CFS/CIICP	CZ(0°)/CIICP/CZ(90°)
Ultimate force (N)	322.52	296.72	323.23

**Table 5**

Features of samples fabricated by different tool-paths-part.2.

Physical quantities	CFS	CFS/CZ(0°)	PLA(Cura)
Ultimate force (N)	267.68	348.5	182.48

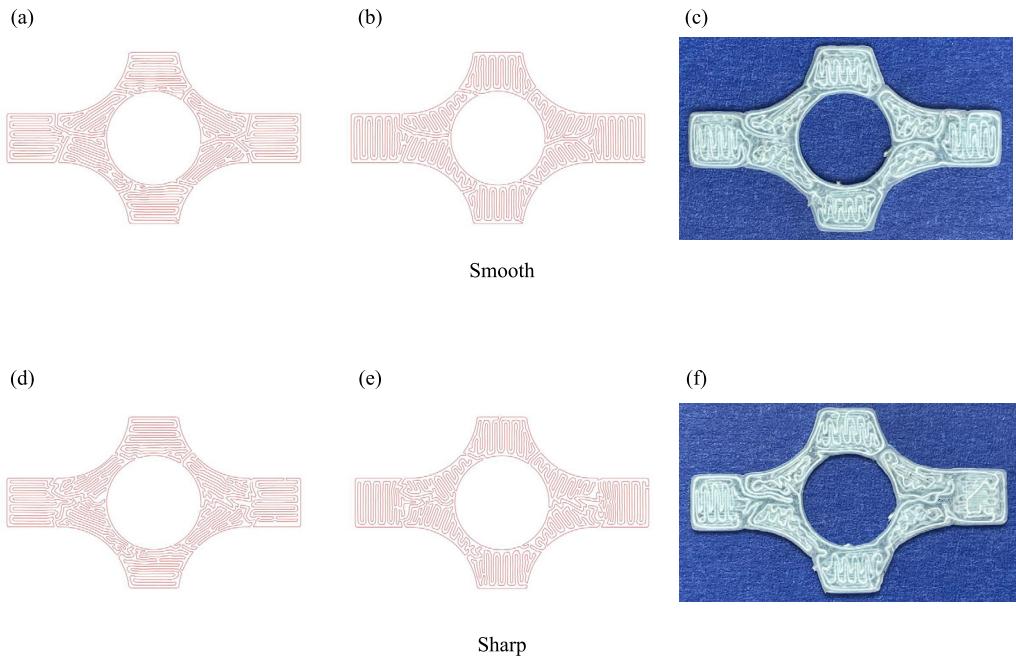
we obtain the minimum spanning tree of this undirected graph, as shown in Fig. 21(b). Finally, for each node, a 2D continuous path is generated as in Section 4, and then a depth-first search is performed from the root node to complete the path-to-path connection and generate a continuous IICP path.

### 6.4. Tensile experiment

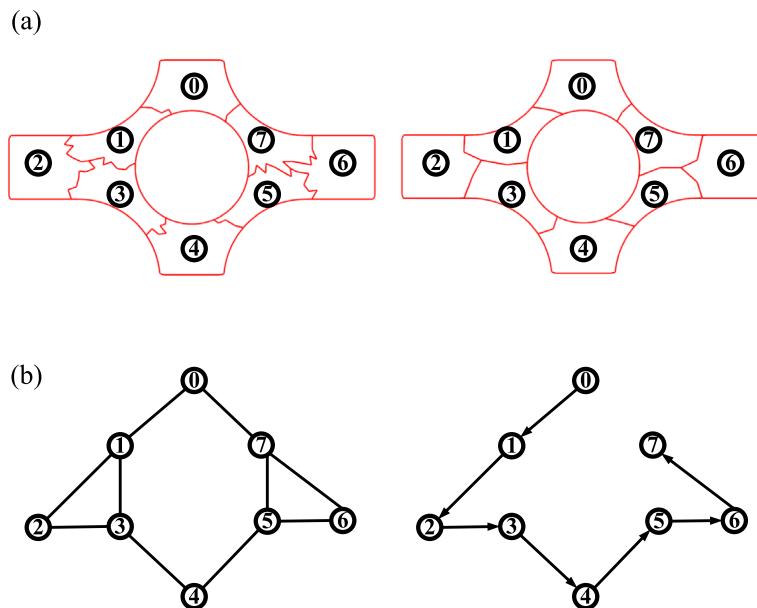
The thickness of the printed samples is 1.7 mm. The initial layer height is 0.3 mm and the layer thickness is 0.2 mm. The printer used is shown in Fig. 22. The experiment was performed with a microcomputer-controlled material testing machine for the tensile test (2 kN capacity, Shanghai songdun Manufacturing Co., Ltd). The loading speed is 10 mm/min. All tested samples were printed using the four continuous paths depicted in Fig. 23. Some samples were printed using only a single path, while others were created by combining multiple paths.

All results are the average of three sets of repeated tests. Fig. 24 shows a sample of the tensile test and the tensile test machine. The testing results are shown in Fig. 25 and Tables 4 and 5. We chose the yield limit as the criterion to measure the strength of the samples. We take the strength of the white PLA sample printed according to the paths generated by Ultimaker Cura 5.1.0 as a benchmark for measurement. We arrange them in ascending order based on the degree of improvement: CFS(46.7%), CFS/CIICP(62.6%), CZ(76.7%), CZ/CIICP/CZ(77.1%), CFS/CZ(90.9%). Due to the weak boundaries of CIICP, the mechanical properties of pure CIICP samples are not ideal. We combined CIICP with CFS and found that CIICP does improve the mechanical properties of CFS. However, for the interlaced CZ samples, the improvement effect is not significant.

In all specimens, the interlacing of CFS and CZ (0°) achieved the highest strength. We speculate that, first, CZ plays a similar strengthening role to CIICP for CFS, although it does not result in the filling path being completely arranged according to the principal stress direction. However, the printing direction of CZ



**Fig. 20.** Global continuous toolpath for IICP with the smooth and unsophisticated feature. (a) is the continuous fiber-reinforced IICP path generated by smoothing the partition boundary using a Bezier curve; (b) is the path intersected with (a) at an angle of 90°; (c) is the actual effect of interleaved printing of (a) and (b); (d) is the continuous fiber-reinforced IICP path generated by the original partition; (e) is the path intersected with (d) at an angle of 90°; (f) is the actual effect of interleaved printing of (d) and (e).

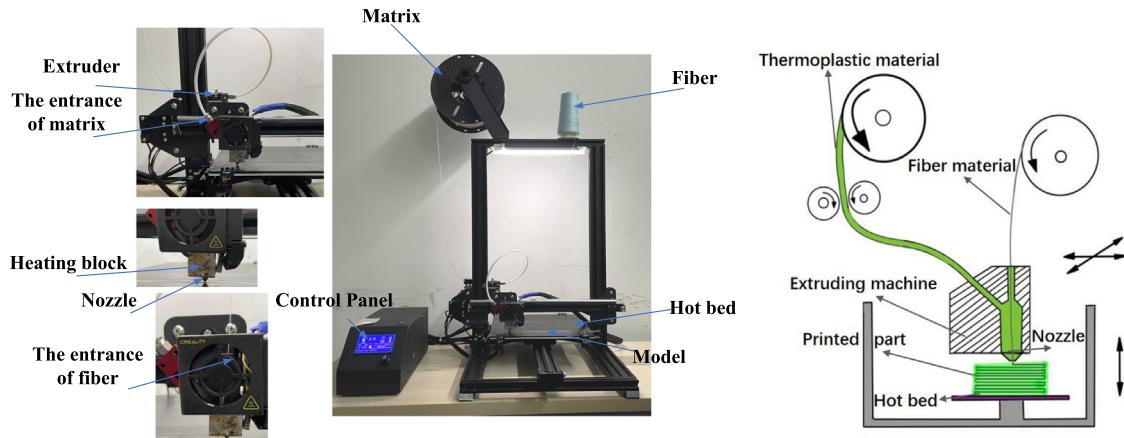


**Fig. 21.** Schematic diagram of the generation process of continuous IICP. (a): The left side shows the original partitions, the right side shows the partitions after smoothing their boundaries using Bezier curves; (b) the left side shows the undirected graph composed of partitions, and the right side shows the minimum spanning tree of the undirected graph.

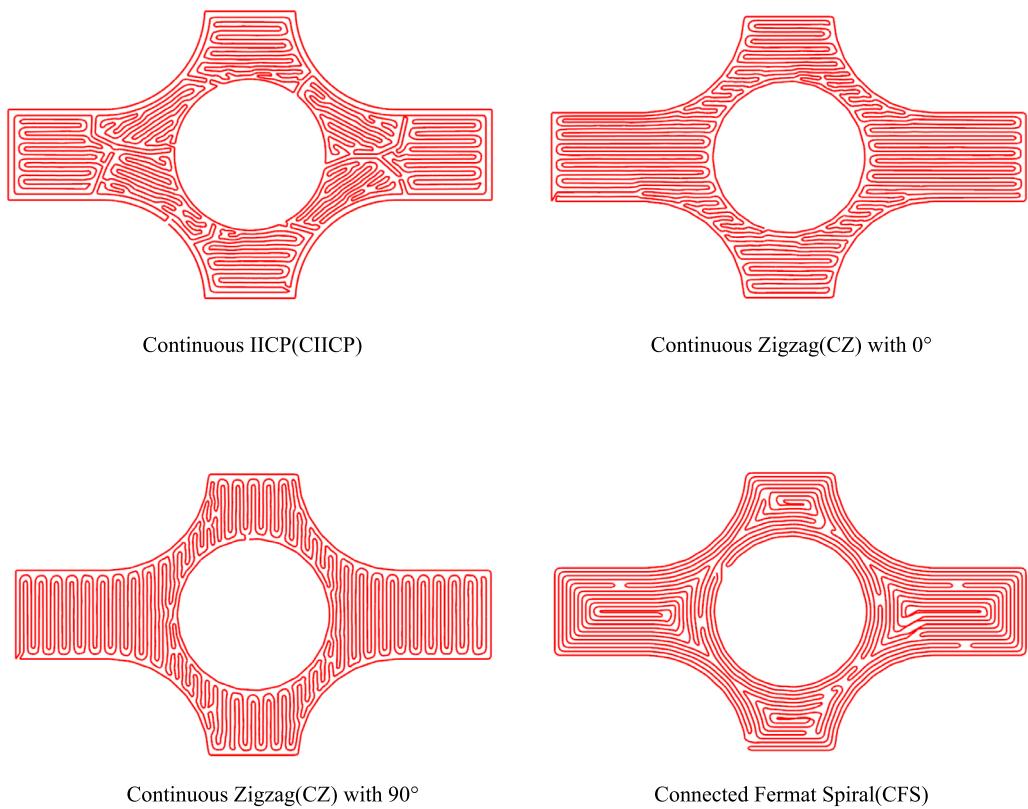
is generally close to the loading direction of various parts of the specimen. Second, due to its inherent Zigzag-like path properties, CZ may drag and produce holes at the edges due to the existence of many sharp corners.

Nevertheless, it has a good filling density and a relatively uniform fiber distribution in other parts, while CFS has a good filling density at the edges but has some holes in the center of the path. The advantages and disadvantages of the two complement

each other, resulting in further improvement in the strength of the specimen. The reason why the staggered filling of CZ(0°/90°) performs well in terms of mechanical strength is believed to be due to the fact that CZ(90°) fills the holes in CZ(0°). However, it is not as good as CFS/CZ(0°) because the paths of CZ(90°) are generally perpendicular to the direction in which the sample is loaded. Although CIICP has a certain degree of improvement effect, due to its weak boundaries. For IICP, we assume that



**Fig. 22.** Left: The printer and the components. Right: The Working schematic of the Co-extrusion printing method. Before printing, the fibers are passed through the heating block and nozzle, feeding a portion of the matrix into the chamber. The fibers and matrix are mixed and heated in the chamber, and printing starts when the set temperature is reached. At this point, the fibers extruded from the nozzle are impregnated with the matrix, and during the printing process, the extruder continuously feeds the matrix into the cavity, and the fibers are continuously pulled toward the cavity by the adhesion force during printing, and finally pass through the nozzle together to finish printing.



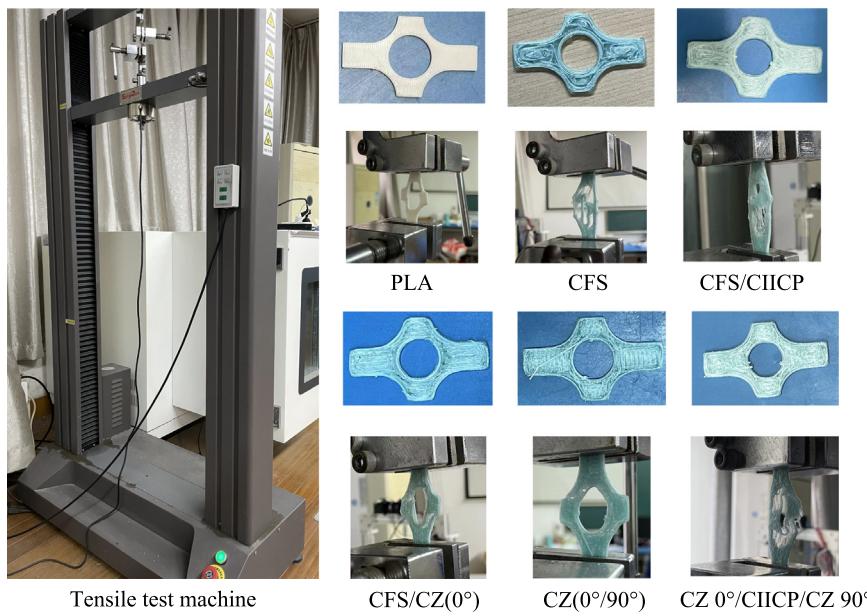
**Fig. 23.** Four continuous paths and their names and abbreviations.

the fiber volume fraction of the printed sample is lower at the boundary, the direction of fiber axial deposition is inconsistent with the direction of force transmission, and the distribution of matrix material is not uniform, making it more prone to failure. It may be more suitable for applications involving thermal or electrical conductivity. Further verification is needed.

During this experiment, as long as we initially adjusted the height between the nozzle and the hotbed and printing parameters, the specific experimental results fluctuated slightly but

basically did not change the order of intensity, except that the second and third places sometimes switched.

Next, we carried out a quantitative analysis of the path in Fig. 23 from the maximum/average angle, the number of occurrences of angles greater than 90°, and the stress orientation following. We choose the CZ(0°), CFS, and CIICP and resample these paths so that the number of points on paths is 4000. Results are displayed in Fig. 26. Note that the frequency indicates how many of the 4000 points have a turning angle greater than 90°.



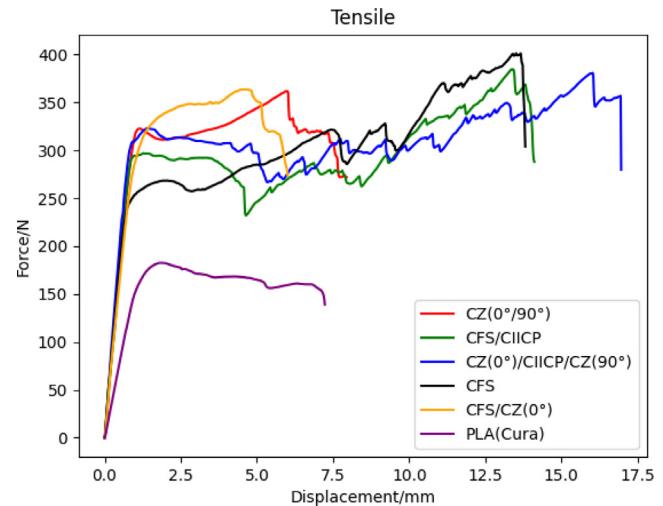
**Fig. 24.** The samples printed according to the path generated in Fig. 23 and fracturing sample after tensile test and the tensile test machine.

On the aspect of stress direction following, we have designed a set of evaluation criteria. We first obtain the principal stress distribution diagram of the sample, and then we use the Farthest Point Sampling (FPS) to obtain the key points in the principal stress distribution. Next, take the sampled stress distribution point as the center, set the search radius to 0.8 mm, and find the point set on the path. Assuming that  $S_i$  is the point of stress distribution and  $M_i$  is the point set on the toolpath near  $S_i$ . Note that the length of  $M_i$  is not fixed. Next, the cosine similarity of the principal stress direction of  $S_i$  to all direction vectors of  $M_i$  is calculated. We take the maximum or we calculate the mean to represent the stress orientation following between  $S_i$  and the path it covers. Finally, the stress orientation following of each sampled point is averaged to represent the whole stress orientation following of the path.

Overall, CFS is better than CZ and CIICP at corners. If we had optimized it according to the original paper, it would have produced better results, particularly at the maximum turning angle. In terms of principal stress direction following, CIICP and CZ are better than CFS in terms of mean or maximum value.

## 7. Conclusion

We demonstrate a global continuous path planning method for CFRTPCs printing that can provide the ability to customize the local deposition path direction and ensure global path continuity to avoid interference and reduce the post-processing workload. Therefore, it is suitable for continuous path planning for the multiphysics field optimization of complex structures. This will take full advantage of high-performance fibers to improve the customization capability and partial performance of 3D printing in different fields. The proposed method can further cooperate with the volume change, surface mapping, and other methods to achieve multi-degrees-of-freedom printing. There are still some areas for improvement in this method. Our algorithm uses a greedy algorithm in both geometry decomposition and print sequence optimization; therefore, it is not guaranteed to be a globally optimal solution. In addition, in the selection of connecting points, we sample the paths and use the sampled sequence of points to proxy the original paths to participate in the encoding of the genetic algorithm, which improves the efficiency but also leads to the obtained solutions not being optimal.



**Fig. 25.** Schematic diagram of tensile testing results.

More importantly, Zigzag-like paths exhibit sharp features at the transitions, which is an inherent problem with this method. This makes our path pattern unsuitable for fiber-reinforced materials with high stiffness, such as carbon fibers. The turning angle [43] of the path, stiffness of fiber materials, and characteristics of different fiber materials affect whether the fiber can be printed in accordance with the planned road stiffness in actual printing. At present, we have optimized the speed and extrusion volume at positions where the path turn to suit actual printing.

In future work, with respect to the algorithm, we will add physical models and data-driven methods to predict the deformation at the path turns, and make predictions and add corresponding post-treatment processes for different materials separately, using a compaction roller [44] and other methods to enhance the bonding between the layers of the printed parts. We will also develop new tests to study the mechanical properties, thermal insulation, electrical conductivity, and other properties of CFRTPCs printed parts in relation to the filling direction. In terms of continuous paths in 2D, we hope to combine the continuous

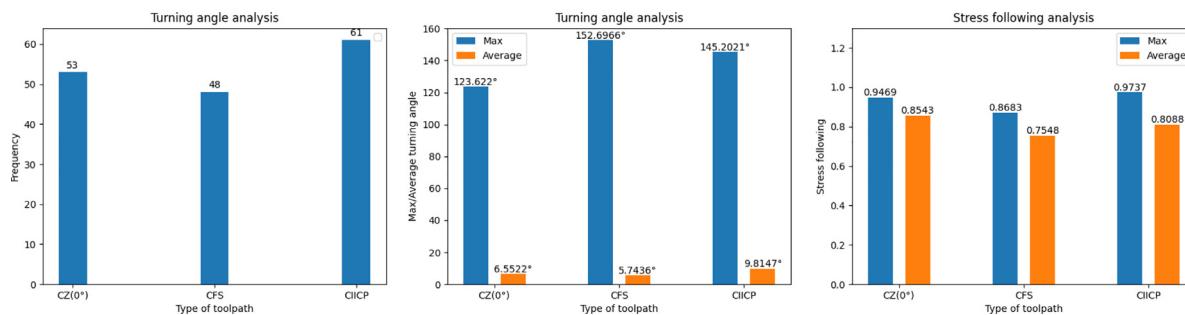


Fig. 26. Quantitative Analysis diagram.

path generation method with machine learning algorithms and transform the path generation problem into a problem of finding optimal solutions in a finite set, which would be an interesting work.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

Data will be made available on request

## Appendix A. Supplementary data

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.cad.2023.103593>.

## References

- [1] Jiang J, Yu C, Xu X, Ma Y, Liu J. Achieving better connections between deposited lines in additive manufacturing via machine learning. *Math Biosci Eng* 2020;17(4):3382–94.
- [2] Haleem A, Javaid M. 3D printed medical parts with different materials using additive manufacturing. *Clin Epidemiol Glob* 2020;8(1):215–23.
- [3] Blok LG, Longana ML, Yu H, Woods BK. An investigation into 3D printing of fibre reinforced thermoplastic composites. *Addit Manuf* 2018;22:176–86.
- [4] Isobe T, Tanaka T, Nomura T, Yuasa R. Comparison of strength of 3D printing objects using short fiber and continuous long fiber. *IOP Conf Ser Mater Sci Eng* 2018;406:012042, IOP Publishing.
- [5] Naranjo-Lozada J, Ahuett-Garza H, Orta-Castañón P, Verbeeten WM, Sáiz-González D. Tensile properties and failure behavior of chopped and continuous carbon fiber composites produced by additive manufacturing. *Addit Manuf* 2019;26:227–41.
- [6] Tu Y, Tan Y, Zhang F, Zhang J, Ma G. Shearing algorithm and device for the continuous carbon fiber 3D printing. *J Adv Mech Des Syst Manuf* 2019;13(1):JAMDSM0016.
- [7] Riddick JC, Haile MA, Wahlde RV, Cole DP, Bamiduro O, Johnson TE. Fractographic analysis of tensile failure of acrylonitrile-butadiene-styrene fabricated by fused deposition modeling. *Addit Manuf* 2016;11:49–59.
- [8] Michell AGM. LVIII. The limits of economy of material in frame-structures. *Lond Edinb Dubl Philos Mag J Sci* 1904;8(47):589–97.
- [9] Fang G, Zhang T, Zhong S, Chen X, Zhong Z, Wang CC. Reinforced FDM: Multi-axis filament alignment with controlled anisotropic strength. *ACM Trans Graph* 2020;39(6):1–15.
- [10] Kabir SMF, Mathur K, Seyam A-FM. The road to improved fiber-reinforced 3D printing technology. *Technologies* 2020;8(4):51.
- [11] Zhang G, Wang Y, He J, Xiong Y. A graph-based path planning method for additive manufacturing of continuous fiber-reinforced planar thin-walled cellular structures. *Rapid Prototyp J* 2023;29(2):344–53.
- [12] Moretz L. How robots help additive manufacturers add precision: Robotics and other technology advances increase accuracy and allow robots to improve precision and make larger parts for additive manufacturing applications. *Control Eng* 2021;68(1):16–9.
- [13] Dwivedi R, Kovacevic R. Automated torch path planning using polygon subdivision for solid freeform fabrication based on welding. *J Manuf Syst* 2004;23(4):278–91.
- [14] Ding D, Pan ZS, Cuiuri D, Li H. A tool-path generation strategy for wire and arc additive manufacturing. *Int J Adv Manuf Technol* 2014;73(1):173–83.
- [15] Gomez G, Cortés C, Creus C, Amilibia MZ, Moreno A. Generation of continuous hybrid zig-zag and contour paths for 3D printing. *Int J Adv Manuf Technol* 2022;119(11):7025–40.
- [16] Jin Y, He Y, Fu G, Zhang A, Du J. A non-retraction path planning approach for extrusion-based additive manufacturing. *Robot Comput-Integr Manuf* 2017;48:132–44.
- [17] Zhao H, Gu F, Huang Q-X, Garcia J, Chen Y, Tu C, et al. Connected fermat spirals for layered fabrication. *ACM Trans Graph* 2016;35(4):1–10.
- [18] Zhai X, Chen F. Path planning of a type of porous structures for additive manufacturing. *Comput Aided Des* 2019;115:218–30.
- [19] Bedel A, Couderc-Osmont Y, Martínez J, Nishat RI, Whitesides S, Lefebvre S. Closed space-filling curves with controlled orientation for 3D printing. *Computer Graphics Forum* 2022;41(2):473–92.
- [20] Xia L, Ma G, Wang F, Bai G, Xie YM, Xu W, et al. Globally continuous hybrid path for extrusion-based additive manufacturing. *Autom Constr* 2022;137:104175.
- [21] Bi M, Xia L, Tran P, Li Z, Wan Q, Wang L, et al. Continuous contour-zigzag hybrid toolpath for large format additive manufacturing. *Addit Manuf* 2022;55:102822.
- [22] Wang D, Wang H, Wang Y. Continuity path planning for 3D printed lightweight infill structures. In: 2021 IEEE conference on telecommunications, optics and computer science. IEEE; 2021, p. 959–62.
- [23] Gupta P, Krishnamoorthy B, Dreifus G. Continuous toolpath planning in a graphical framework for sparse infill additive manufacturing. *Comput Aided Des* 2020;127:102880.
- [24] Yoo C, Lensgraf S, Fitch R, Clemon LM, Mettu R. Toward optimal FDM toolpath planning with monte carlo tree search. In: 2020 IEEE international conference on robotics and automation. IEEE; 2020, p. 4037–43.
- [25] Zhong F, Xu Y, Zhao H, Lu L. As-continuous-as-possible extrusion-based fabrication of surface models. *ACM Trans Graph* 2022;42(3):1–16.
- [26] Wang X, Chen L, Lau T-Y, Tang K. A skeleton-based process planning framework for support-free 3+ 2-axis printing of multi-branch freeform parts. *Int J Adv Manuf Technol* 2020;110:327–50.
- [27] Li Y, Tang K, He D, Wang X. Multi-axis support-free printing of freeform parts with lattice infill structures. *Comput Aided Des* 2021;133:102986.
- [28] Prajapati H, Ravoori D, Woods RL, Jain A. Measurement of anisotropic thermal conductivity and inter-layer thermal contact resistance in polymer fused deposition modeling (FDM). *Addit Manuf* 2018;21:84–90.
- [29] Li Z, Li H, Rao J, Yan C, Li L, Lin F, et al. Anisotropy on electrical insulation performance of 3D printed nylon 12. In: 2017 IEEE conference on electrical insulation and dielectric phenomenon. IEEE; 2017, p. 42–5.
- [30] Nguyen N, Park JG, Zhang S, Liang R. Recent advances on 3D printing technique for thermal-related applications. *Adv Energy Mater* 2018;20(5):1700876.
- [31] Olcun S, Ibrahim Y, Isaacs C, Karam M, Elkholy A, Kempers R. Thermal conductivity of 3D-printed continuous pitch carbon fiber composites. *Addit Manuf Lett* 2023;4:100106.
- [32] Xiao X, Roh B-M, Zhu F. Strength enhancement in fused filament fabrication via the isotropy toolpath. *Appl Sci* 2021;11(13):6100.
- [33] Chen X, Fang G, Liao W-H, Wang CC. Field-based toolpath generation for 3D printing continuous fibre reinforced thermoplastic composites. *Addit Manuf* 2022;49:102470.
- [34] Li Y, Xu K, Liu X, Yang M, Gao J, Maropoulos P. Stress-oriented 3D printing path optimization based on image processing algorithms for reinforced load-bearing parts. *CIRP Ann* 2021;70(1):195–8.
- [35] Zhang H, Yao Y, Ma Y, Lackner M, Jiang Y. A 3D printing tool-path generation strategy based on the partition of principal stress field for fused filament fabrication. *Int J Adv Manuf Technol* 2022;122:1719–35.
- [36] Xia L, Lin S, Ma G. Stress-based tool-path planning methodology for fused filament fabrication. *Addit Manuf* 2020;32:101020.

- [37] Livesu M, Ellero S, Martínez J, Lefebvre S, Attene M. From 3D models to 3D prints: an overview of the processing pipeline. In: Computer graphics forum. Vol. 36. Wiley Online Library; 2017, p. 537–64.
- [38] Mokhtarian F, Abbasi S. Shape similarity retrieval under affine transforms. Pattern Recognit 2002;35(1):31–41.
- [39] Qi CR, Yi L, Su H, Guibas LJ. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In: Advances in Neural Information Processing Systems. 30, Curran Associates, Inc.; 2017, p. 1–10.
- [40] Fechteler P, Eisert P, Rurinsky J. Fast and high resolution 3D face scanning. In: 2007 IEEE International Conference on Image Processing. 3, IEEE; 2007, p. III - 81-III - 84.
- [41] Yao Y, Li M, Lackner M, Herfried L. A continuous fiber-reinforced additive manufacturing processing based on PET fiber and PLA. Materials 2020;13(14):3044.
- [42] Jiang X, Huang B, et al. Global path planning of fixed-wing UAV based on improved RRT\* algorithm. J Appl Sci Eng 2023;26(10):1441–50.
- [43] Matsuzaki R, Nakamura T, Sugiyama K, Ueda M, Todoroki A, Hirano Y, et al. Effects of set curvature and fiber bundle size on the printed radius of curvature by a continuous carbon fiber composite 3D printer. Addit Manuf 2018;24:93–102.
- [44] Andreu A, Kim S, Dittus J, Friedmann M, Fleischer J, Yoon Y-J. Hybrid material extrusion 3D printing to strengthen interlayer adhesion through hot rolling. Addit Manuf 2022;55:102773.