Computer-Aided Design 114 (2019) 51–63

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Curved layer based process planning for multi-axis volume printing of freeform parts☆,☆☆

Ke Xu [a], Yingguang Li [a,*], Lufeng Chen [b], Kai Tang [c]

[a] National Key Laboratory of Science and Technology on Helicopter Transmission, Nanjing University of Aeronautics and Astronautics, China
[b] School of Automation Engineering, University of Electronic Science and Technology of China, China
[c] Department of Mechanical and Aerospace Engineering, Hong Kong University of Science and Technology, China

ARTICLE INFO

Article history:
Received 7 October 2018
Received in revised form 30 April 2019
Accepted 7 May 2019

Keywords:
3D printing
Multi-axis additive manufacturing
Path generation
Support structure

ABSTRACT

The traditional 2.5-axis volume printing process purely relies on planar and parallel slicing layers, which imperatively requires the support structure when dealing with overhanging features on the part. The advent of multi-axis additive manufacturing inaugurates a brand new type of printing process with an adjustable build direction, based on which the support structure can be successfully reduced (if not completely eliminated) upon a proper process planning. Presented in this paper is a curved layer based process planning algorithm for multi-axis printing of an arbitrary freeform solid part. Given a freeform solid model represented as a watertight mesh surface, our algorithm starts with the establishment of a surface embedded field, whose value at any particular point is exactly the geodesic distance to the specified bottom of the model. Any iso-level contour induced from this field is first flattened, filled by a Delaunay triangular mesh, and then mapped back to 3D space through the Harmonic mapping to interpolate the original 3D contour, thus generating a curved layer. After the entire model is decomposed into curved layers by the proposed adaptive slicing strategy, the multi-axis printing paths are then generated on these layers in a contour-parallel fashion. Finally, following the strict increasing order of iso-levels, the contours are printed one by one till the final formation of the part. Preliminary tests in both computer simulation and physical printing of our algorithm have given a positive validation on its effectiveness and feasibility in eliminating the need of support structure.

1. Introduction

Additive manufacturing (AM), also known as 3D printing, is one of the primary choices for fast prototyping and customized production of parts with complex design features [1]. The majority of up-to-date commercial AM systems are still simply based on the 2.5-axis configuration, in which the part is first decomposed into parallel thin layers with uniform thickness, and the AM nozzle deposits materials sequentially from bottom up along a fixed printing direction (+Z). Though easily implemented, such a 3D printing platform with highly restricted printing motions comes along with several issues (see Fig. 1 for illustration). The first and most prominent issue is the excessive usage of support structure constructed prior to those overhanging features [2], which obviously leads to material and time waste. Another pertinent problem of the support structure is that the residue artifacts staying on the contact surface even after a delicate cleanup process leads to a compromised surface quality. The second issue of the 2.5-axis platform is the poor surface quality at those non-vertical faces caused by the staircase effect [3], as demonstrated in Fig. 1(b). Restricted by this 2.5-axis configuration, people have spent great endeavor trying to reduce the support volume as well as the staircase effect, either by finding an optimal build direction [4] or by adaptively adjusting the slicing layer thickness [5–8]. However, these two issues always exist just because of the nature of 2.5-axis deposition.

Inspired by the five-axis numerical controlled machining with additional two rotary axes, the concept of multi-axis 3D printing, as already realized in some prototypes, is expected to offer an ultimate solution to the aforementioned issues. The advent of multi-axis 3D printing can successfully bypass these two issues by dynamically adjusting the printing direction [9]. Considering the gravity effect that the fused material needs to be deposited vertically, two additional rotary DOFs are usually integrated on the platform either formed by a rotary table or a robotic arm. Based on this configuration, it becomes possible that the additive

☆ This paper has been recommended for acceptance by Yong Chen.
☆☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.cad.2019.05.007.

* Corresponding author.
E-mail address: liyingguang@nuaa.edu.cn (Y. Li).

https://doi.org/10.1016/j.cad.2019.05.007
0010-4485/

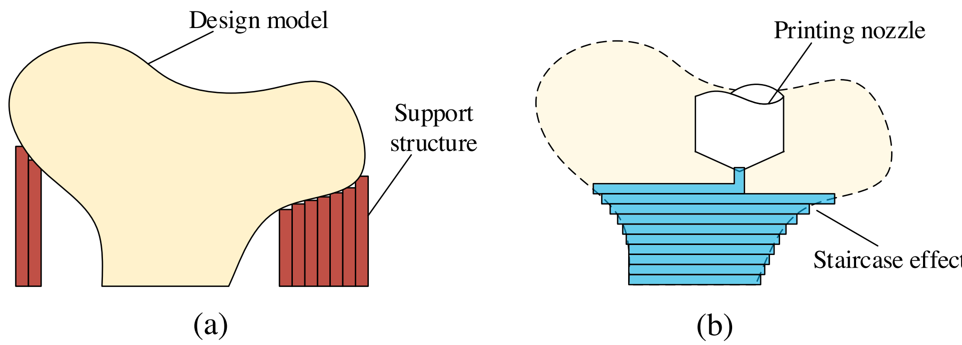

Fig. 1. (a) Support structure and (b) staircase effect in 2.5-axis AM process.

process can be conducted in a theoretically support-free manner (when the potential nozzle collision is not considered), while achieving a much more enhanced surface finish quality. However, situation becomes particularly intricate in multi-axis fabrication. In terms of hardware, as the nozzle orientation can constantly change during the fabricating process, a delicate motion control to synchronize the deposition rate with the nozzle movement (including rotation and translation) is indispensable. On the other hand, the trickiest part is the layer decomposition of the target model, which still lacks a versatile strategy towards the objective of support-free fabrication. This newly emerged area has attracted researchers to develop various hardware and software solutions for additive fabrication of complex parts. Some of the recent developments are summarized below.

1.1. Related work

In terms of hardware system, Pan et al. [10] developed a CNC based accumulation platform to realize five-axis motions, which was successfully utilized for fabricating conformal features on existing surfaces. Another hardware configuration based on Stewart platform [11] was later devised for a low-cost FDM process, in which a laser-camera system was employed to correct the backlash errors in real time. Its major deficiency is still the lack of software system for automatic generation of the printing path. To facilitate the process planning while achieving multidirectional printing, a 3D printer named "RevoMaker" [12] was introduced lately. As the core of this system, a cuboidal base with embedded functional components was first extracted from the 3D model, and the external features were then printed directly on the six facets of the base, in respective order. The drawback of this system is the need of multiple setups for the workpiece during the process. A novel decompose-and-pack idea was presented recently [13], which divided a complex geometry into printable parts and stacked them up to minimize the pile height subject to the overhanging feature constraint. Similarly, Attene [14] inaugurated a disassembling strategy to minimize the packing volume, i.e., the bounding box, to practically facilitate a batch production. Keating and Oxman [15] integrated a 6-DOF robotic arm into the FDM process and successfully realized a proof-of-concept multi-axis printing process. Wu et al. [16] recently presented another robotic system named RoboFDM that incorporated both hardware and software for support-free FDM — a robotic arm providing six degrees of freedom was employed to realize multidirectional motions, while the printing nozzle remained vertically fixed to guarantee a correct material deposition along the gravity.

Notwithstanding recent advancements in hardware of multi-axis printing, the major challenge still resides on the software development. In particular, so far there is no definitive answer to how a general sculptured part with overhanging features can be fabricated on a multi-axis 3D printer, without the need of support structure and hence with a better finish surface quality. Solutions already proposed towards this objective can be roughly classified into planar and non-planar decomposition. In terms of planar decomposition, the sliced primitive is still with a planar base but fabricated along a changing but fixed direction. Zhang and Liou [17] presented an adaptive slicing method to decompose a columnar model into layers of non-uniform thickness while trying to eliminate the support. Lee and Jee [18] proposed an auto-partitioning algorithm for 3D metal printing, which was successfully applied to structural parts consisting of prismatic shapes. Wu et al. [16] developed a support-free decomposition strategy for sculptured parts based on the extracted skeleton. A similar idea using 3+2-axis printing motion was reported in our recent work [19], where a novel downward flooding approach was devised for the extraction of support-free features. Ding et al. [20] introduced a decomposition-regrouping workflow for multi-directional slicing of STL models. However, this scheme is incapable of handling complex geometries or models with non-sharp edges. In terms of shell models, Wei et al. [21] recently developed a skeleton-based partitioning algorithm towards the goal of minimizing the total number of partitions and the length of cuts. The partitioned parts were individually fabricated in a support-free manner and then glued together to form the final model.

All the above multi-directional strategies are based on planar layers when fabricating each segment. To fully exploit the kinematic capacity of a multi-axis system, recently, some state-of-the-art research came up with multi-axis continuous fabrication on curved slicing layers, i.e. using non-planar decomposition. Coupek et al. [22] proposed a multi-axis FDM process for cylindrical base models, which were represented and decomposed in a cylindrical coordinate system. A novel process specifically for slender parts was presented recently [23], which decomposed the model along its medial axis into parallel curved fibers in order to achieve better mechanical properties; the support structure however was still needed. Dai et al. [24] proposed a support-free curved layer decomposition algorithm using convex-fronts. Their approach is versatile towards various freeform models including multi-genus features. However, one noticeable concern of their work is the discretization error due to the voxel representation, which also demands a huge data storage as well as a high computing cost.

1.2. Contributions

All the above endeavors in the area of multi-axis printing provide great inspiration to us. However, most of them are of limited utility for an arbitrarily shaped model. A general multi-axis process for freeform part fabrication towards the goal of reducing the need of support is particularly scarce in both industry and academia. Dai's work [24] is among the most effective ones for its ability to print a wide range of models without the need of

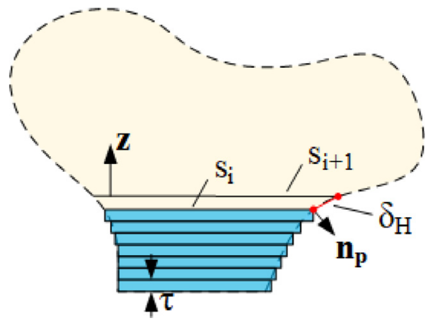

Fig. 2. Definition of Hausdorff distance between each two printing layers.

support structure. The key to the success of their algorithm is the idea of convex growing based on a third-party voxel representation. Aimed at the same target of support-free fabrication, we in this paper propose an alternative scheme without the need of a third-party representation, for the process planning of multi-axis 3D printing of freeform models. The chief contribution of this work is the development of a curved layer decomposition, which is purely based on the original boundary representation of the input model. To handle the non-uniformity issue caused by the decomposed curved layers, we also propose a new pattern of printing path, which is quite similar to Dai's but offers more precise control of the depositing rate. As to be elaborated, the proposed method shows a high versatility for a variety of models, including those with multi-branches or multi-genus features.

This paper is organized as follows. In the Section 2, we introduce the idea of shape constrained geodesic field and its generation, and describe how to construct curved layers from this field. Then, in Section 3, based on the constructed curved layers, the multi-axis printing path generation algorithm is presented. In Section 4, the test results of our method are presented, followed by the conclusion in Section 5.

2. Curved layer decomposition of freeform parts

The task of layered additive process planning can be formulated as a two-phase procedure. Given a solid model Ω which is conventionally represented by a watertight triangular mesh surface $\partial\Omega$, the first step is to decompose Ω into layers of cross-sectional surfaces $\{S_i\}$. Subsequently, a printing path T_i is generated for fabricating each S_i with a controlled depositing rate. Pertaining to the former, in the realm of 2.5-axis lamination, the Hausdorff distance δ_H between each two consecutive layers S_i and S_{i+1} can be approximated according to Fig. 2 as:

$$\delta_H(S_i, S_{i+1}) = \begin{cases} \max\limits_{p \in \partial S_{i+1}} \dfrac{\tau}{\sin\left(\cos^{-1} \boldsymbol{n_p} \cdot \boldsymbol{z}\right)}, \mathit{if}\, \boldsymbol{n_p} \cdot \boldsymbol{z} < \boldsymbol{0} \\ \tau, \mathit{if}\, \boldsymbol{n_p} \cdot \boldsymbol{z} \geq \boldsymbol{0} \end{cases} \quad (1)$$

where τ defines the layer thickness. Conceivably, when δ_H is large, the staircase effect becomes prominent while the need of support structure arises, as shown in Fig. 2. Given this, the Hausdorff distance δ_H is regarded as the direct culprit that affects the additive process, which thus needs to be confined to ensure a firm adherence and better surface quality.

To maintain a relatively stable δ_H throughout the printing process, in a traditional 2.5-axis printing process it is intuitive to make adaptive adjustment of the thickness τ for different layers. However, there is an efficiency issue when τ becomes too small, and then an accuracy issue when τ is smaller than the resolution of nozzle extrusion. In our previous work [19], we took advantage of the multi-axis platform to partition the model into several segmented parts, each of which can be fabricated along a unique direction without the need of support structure. This approach manages to bypass the need of smaller layer thickness and is able to achieve some plausible results for parts with overhanging features; however, the adherence at the segment interfaces is weak. Another recent work of ours [25] verified the feasibility of curved layer deposition with variable thickness by fully exploiting the capacity of a multi-axis FDM printer, which set up a foundation of curved layer fabrication. Inspired by these previous achievements, here we want to decompose a freeform model in a more consistent manner, i.e., into stacks of curved layers, towards the target of support-free and high-quality fabrication. To achieve this goal, the following requirements for the layer decomposition should be satisfied:

Rq1: The Hausdorff distance δ_H of any two consecutive layers should be confined within a threshold.

Rq2: The layer thickness τ should be bounded inside an admissible range $\tau \in [\tau_l, \tau_u]$ to guarantee proper material deposition.

These two requirements can be simultaneously respected by introducing one criterion, which is the ratio between the Hausdorff distance and the layer thickness δ_H/τ, i.e., the *HT ratio*. This ratio evaluates the layer decomposition strategy in terms of the need of support structure. To ensure a support-free fabrication, for every decomposed layer this criterion should be restricted below a threshold value, e.g., 1.5 for most PLA filaments according to [26].

Bearing these conditions in mind, we now propose a curved layer decomposition scheme for freeform shapes which is based on an important and also intrinsic characteristic of shapes, i.e., the geodesic distance. In a nut shell, we first generate a set of iso-field-value contours on the boundary $\partial\Omega$ of the solid model. These contours are then filled with triangular meshes by a rigidity-preserved algorithm. The contour together with its filled internal surface forms one curved layer. By carefully choosing the interval (i.e. the Hausdorff distance) of the field value, a stack of curved layers with no intersection can be generated, based on which the solid part is decomposed. While the exemplified procedure is illustrated in Fig. 3, the algorithmic details are given next.

2.1. Generation of shape constrained iso-geodesic contours

As illustrated in Fig. 4(a), the geodesic distance is a unique metric defined over a surface domain. As opposed to the Euclidean distance which defines the distance between any two points $\boldsymbol{p}$ and $\boldsymbol{q}$ in the Euclidean space, the geodesic distance $\delta(\boldsymbol{q}, \boldsymbol{p})$ is defined as the arc-length of the shortest surface embedded curve between $\boldsymbol{q}$ and $\boldsymbol{p}$.

Assume that the given solid model Ω has a flat bottom layer whose boundary curve is denoted as $l_0 \in \partial\Omega$. We now define a new geodesic metric $\delta^*(\boldsymbol{q})$ which identifies the shortest geodesic distance between point $\boldsymbol{q}$ and l_0, i.e.:

$$\delta^*(\boldsymbol{q}) = \min_{\boldsymbol{p_i} \in l_0} \delta(\boldsymbol{q}, \boldsymbol{p_i}) \quad (2)$$

This geodesic metric effectively induces a scalar field over surface $\partial\Omega$, which is called a *shape constrained geodesic distance field (SCGDF)*. As implied by the name, this scalar field is highly constrained by the shape of the base site (in our case, the bottom contour l_0 whose field value is zero) instead of a conventional point site. Given a watertight boundary surface $\partial\Omega$ and the prescribed bottom contour $l_0 \in \partial\Omega$, the scalar field of Eq. (1) can be computed based on the well-known Mitchell–Mount–Papadimitriou (MMP) algorithm [27] which gives the exact solution to the discrete geodesic distance between any two vertices. The calculation can be expedited by adopting an error-bounded approximated MMP method with a time complexity of

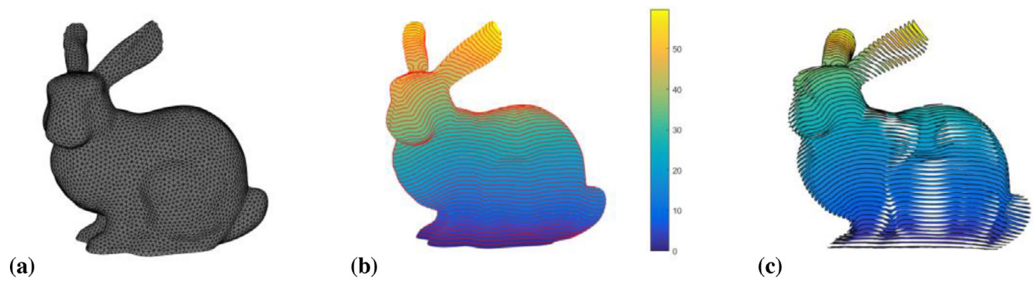

Fig. 3. Illustration of the curved layer decomposition process on a Stanford Bunny: (a) the original mesh model; (b) the generated iso-geodesic contours; (c) the reconstructed surface layers with no intersection.

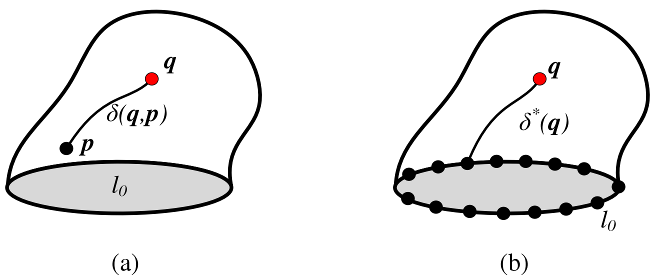

Fig. 4. Geodesic distance from a source point to: (a) destination point; (b) destination contour.

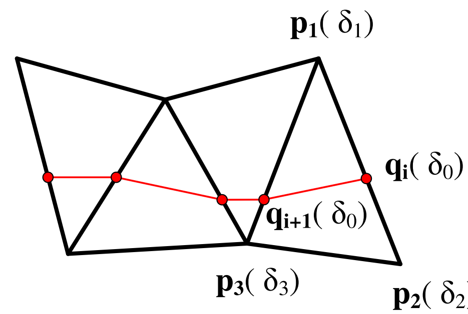

Fig. 6. Computation of an iso-contour.

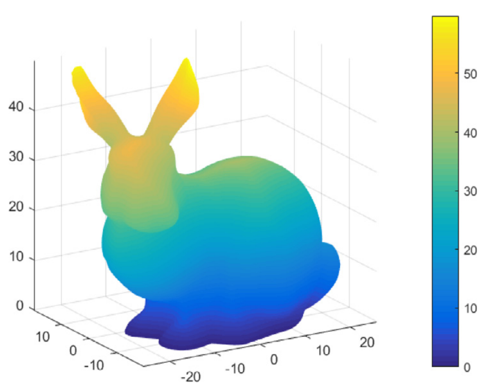

Fig. 5. The scalar field δ^* on the bunny model (a larger value of δ^* is shown in yellow while a smaller one becomes indigo).

$o(n \log n)$ [28]. Fig. 5 depicts one example of the scalar field (the bunny model with a planar base contour) as computed by our implemented computer program.

With SCGDF calculated on the boundary surface $\partial\Omega$, the iso-geodesic contour (or simply iso-contour) C_{δ_0} is essentially the level set with a constant field value δ_0:

$$C_{\delta_0} = \{\boldsymbol{q} | \delta^* (\boldsymbol{q}) = \delta_0\} \tag{3}$$

For a discretized $\partial\Omega$ in the form of a triangular mesh, since the field value is only calculated on those triangle vertices, the iso-contour corresponding to any specified field value δ_0 in general does not exactly pass through these vertices. Fig. 6 illustrates how C_{δ_0} is calculated by us, which is made of a series of $\boldsymbol{q_i}$ lying on some edges. First of all, a candidate edge $\overline{\boldsymbol{p_1 p_2}}$ is sought out where the field values on its two end vertices satisfy the following condition:

$$(\delta_0 - \delta_1) \cdot (\delta_0 - \delta_2) \leq 0 \tag{4}$$

When Eq. (4) holds, there must be a $\boldsymbol{q_i}$ lying on this candidate edge, whose calculation is simply based on the linear interpolation:

$$\boldsymbol{q_i} = \boldsymbol{p_1} + \frac{\delta_0 - \delta_1}{\delta_2 - \delta_1}(\boldsymbol{p_2} - \boldsymbol{p_1}) \tag{5}$$

In this way, the iso-contour C_{δ_0} is approximated as a group of $\{\boldsymbol{q_i}\}$, whose geodesic distances to the bottom contour are exactly δ_0. Consequently, the iso-contours of different field levels can be generated, as shown in Fig. 3(b). Obviously, being a scalar field, these iso-contours never intersect each other. Apart from this nice property, the iso-contours establish a smooth transition from the bottom upwards, giving a natural order of the curved layers to be generated. However, an iso-contour is only a closed 3D curve, not a layer. To print the volume of Ω, we need to devise a "hole filling" operation to generate a layer from an iso-contour, as to be elaborated next.

2.2. Surface layer construction

There are several requirements that need to be respected when a surface layer is constructed from an iso-contour. First and obviously, the layer's boundary must strictly align with the iso-contour. Secondly, the construction should be deterministic, meaning that the solution is unique with no ambiguity, despite countless possibilities that satisfy the first requirement. Thirdly, the surface curvature of layer should be as small as possible.

Our overall procedure is illustrated in Fig. 7, which is composed of three steps: (1) the planarization that flattens the iso-contour into a plane; (2) the filling by triangles of the hole in the plane; and (3) the mapping of the planar triangular mesh back to a 3D surface based on the least square fitting.

The planarization is to project the iso-contour $\{\boldsymbol{q_i}\}$ onto a target plane, while preserving its shape as much as possible. Note that the original iso-contour resulting from the iso-geodesic computation (see Fig. 6) is very likely to have non-uniformly distributed vertices, as shown in Fig. 8(a), which would lead to unevenly sized triangles when filling the planar hole. To overcome this issue, the arc-length based resampling is invoked on

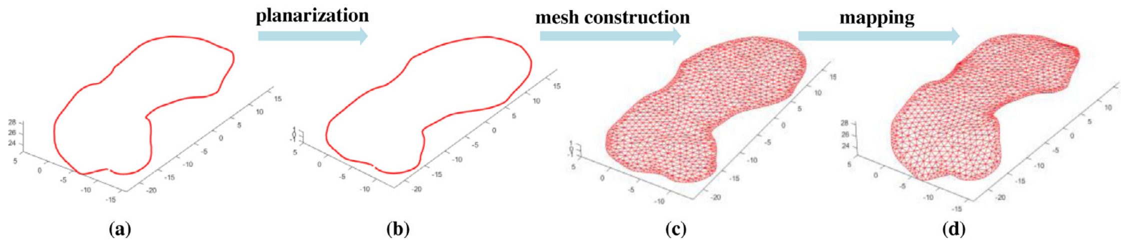

Fig. 7. Surface layer construction from an iso-contour curve.

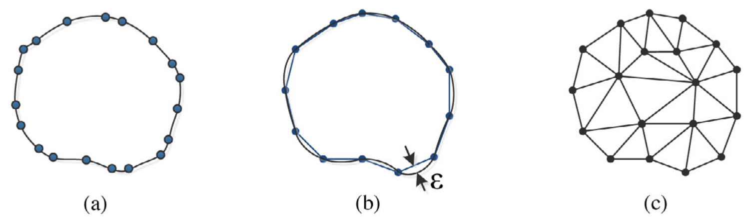

Fig. 8. Filling the planar hole: (a) the original planar iso-contour vertices; (b) the resampled vertices; (c) the constructed triangular mesh.

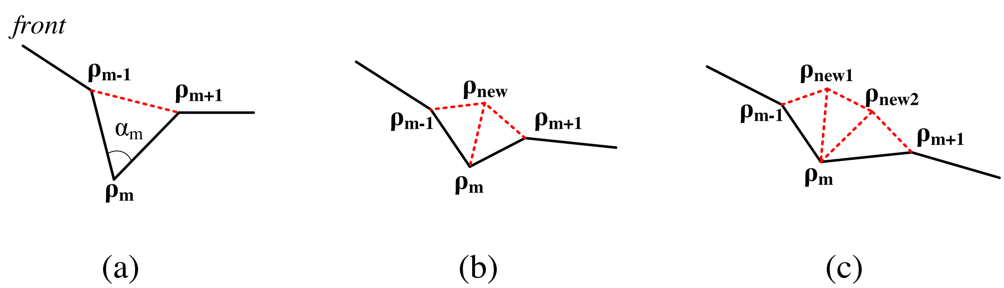

Fig. 9. Rules for creating new triangles: (a) $\alpha_{jmin}^{B} \leq 80°$; (b) $80° < \alpha_{jmin}^{B} \leq 130°$; and (c) $130° < \alpha_{jmin}^{B} < 180°$.

these vertices prior to the planarization as well as the mesh construction process. About this resampling process, the boundary formed by the original vertices is first expressed by a piecewise cubic spline passing through all vertices. We then uniformly distribute the resampled nodes on this spline curve to maintain a constant arc length $\gamma = \widehat{\boldsymbol{q}_l\boldsymbol{q}_{l+1}}$ between each two adjacent nodes. In order to faithfully preserve the shape of boundary, the qualified value of γ is computed such as to prevent the chordal error ε not exceeding the threshold value ε_0 (see Fig. 8(b) for a better understanding of the chordal error).

While resampling is a trivial process, the crux here is the determination of the projecting direction $\boldsymbol{d_p}$, i.e. the normal of the target plane. By applying the well-known principal component analysis (PCA) to the iso-contour vertices, the most irrelevant principal direction is selected as the projecting direction:

$$\left[\left(\boldsymbol{q}_1^-\right)^T, \ldots, \left(\boldsymbol{q}_n^-\right)^T\right] \cdot \begin{bmatrix} \boldsymbol{q}_1^- \\ \vdots \\ \boldsymbol{q}_n^- \end{bmatrix} = \left[(\boldsymbol{d_1})^T, (\boldsymbol{d_2})^T, (\boldsymbol{d_3})^T\right] \cdot \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{d_1} \\ \boldsymbol{d_2} \\ \boldsymbol{d_3} \end{bmatrix} \tag{6}$$

where $\boldsymbol{q}_i^- = \boldsymbol{q_i} - \frac{\sum \boldsymbol{q_i}}{n}$ is the adjusted coordinates of the vertex whose origin is the centroid of the contour. After going through the eigendecomposition, the eigenvector corresponding to the smallest eigenvalue is selected as $\boldsymbol{d_p}$.

Once the iso-contour is projected onto the plane along $\boldsymbol{d_p}$, the inside of the planar contour is then triangulated. To obtain a high quality mesh, we adopt a version of triangulation algorithm that combines the advancing front mesh generation [29] with the Delaunay triangulation [30]. Specifically, let the 2D boundary contour $C^{2D} = \{\boldsymbol{\rho_i}\}$ be a *front*. The angles between each two adjacent boundary edges formed by $\langle \boldsymbol{\rho_{i-1}}, \boldsymbol{\rho_i}, \boldsymbol{\rho_{i+1}} \rangle$ are then calculated and denoted as $\{\alpha_i\}$. Next, we find out the minimum angle α_m and its corresponding three vertices $\langle \boldsymbol{\rho_{m-1}}, \boldsymbol{\rho_m}, \boldsymbol{\rho_{m+1}} \rangle$ to create new triangles based on the following three rules (see Fig. 9):

- If $\alpha_m \leq 80°$, the new triangle is simply $Tri\langle \boldsymbol{\rho_{m-1}}, \boldsymbol{\rho_m}, \boldsymbol{\rho_{m+1}} \rangle$.
- If $80° < \alpha_m \leq 130°$, the new triangles are $Tri\langle \boldsymbol{\rho_{m-1}}, \boldsymbol{\rho_m}, \boldsymbol{\rho_{new}} \rangle$ and $Tri\langle \boldsymbol{\rho_m}, \boldsymbol{\rho_{m+1}}, \boldsymbol{\rho_{new}} \rangle$, where $\boldsymbol{\rho_{new}}$ lies on the bisector of α_m.
- If $130° < \alpha_m < 180°$ (note that the minimum angle will never be larger than 180° for a closed loop), the new triangles are $Tri\langle \boldsymbol{\rho_{m-1}}, \boldsymbol{\rho_m}, \boldsymbol{\rho_{new1}} \rangle$, $Tri\langle \boldsymbol{\rho_m}, \boldsymbol{\rho_{new1}}, \boldsymbol{\rho_{new2}} \rangle$ and $Tri\langle \boldsymbol{\rho_{m+1}}, \boldsymbol{\rho_m}, \boldsymbol{\rho_{new2}} \rangle$, where $\boldsymbol{\rho_{new1}}$ and $\boldsymbol{\rho_{new2}}$ lie on the trisector of α_m.

For the second and third cases above, the newly generated vertices should be merged with other vertices on the front, if the distance between them is smaller than half of the shortest edge length on the front. By doing so, the length of new edges can be stabilized during the advancing propagation rather than gradually shortened. The front will then be updated to indicate the new boundary vertices. By recursively adding new triangles and evolving the front, the boundary contour will be eventually filled by the last triangle, when the final front contains only three last vertices. This front advancing process is guaranteed to converge as it is carried out in the plane. That we further perform a Delaunay refinement on the triangulation is for the sake of the subsequent mapping process, since the Delaunay triangulation preserves the best the original "imaginary" surface shape.

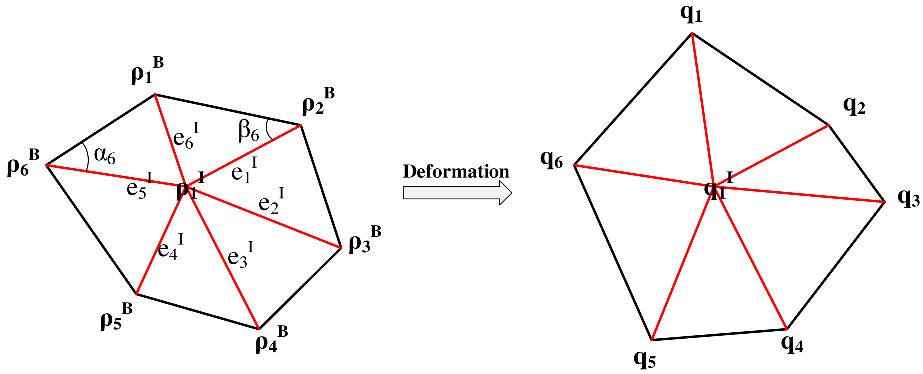

Fig. 10. Topology-preserved deformation of a triangular mesh . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

After the planar contour C^{2D} is triangulated, the final step is to map the planar triangular mesh back to a surface whose boundary is exactly the original 3D iso-contour C. In the spirit of minimum surface, i.e., to be as flat as possible while still meeting the boundary condition, we adopt the Harmonic mapping, which is known to minimize the Dirichlet energy on the final surface. Let the vertices of the triangular surface patch to be found be divided into two groups: $\{\boldsymbol{\rho}_i^B\}$ that represent the boundary vertices corresponding to $\{\boldsymbol{q_i}\}$, and $\{\boldsymbol{\rho}_i^I\}$ that are the internal vertices and correspond to the internal vertices $\{\boldsymbol{q}_i^I\}$ of the 2D mesh. $\{\boldsymbol{\rho}_i^B\}$ obviously is the original iso-contour, hence already known. Thus, we have $3 \times m$ unknowns of $\boldsymbol{q}_i^I = (x_i^I, y_i^I, z_i^I)$, where m is the total number of internal vertices.

With the topology of mesh preserved, the Dirichlet energy $E(\{\boldsymbol{q}_i^I\})$ for a deformed configuration of triangular mesh is simply the summation of the elastic energy on each internal edge e_i^I (shown in red in Fig. 10) which acts as a spring:

$$E(\{\boldsymbol{q}_i^I\}) = \sum_{e_i^I} k_i \left\| \boldsymbol{q_{i,1}} - \boldsymbol{q_{i,2}} \right\|^2 \tag{7}$$

where $\boldsymbol{q_{i,1}}$ and $\boldsymbol{q_{i,2}}$ are the two vertices of e_i^I and k_i is a spring coefficient.

There are a number of ways for the determination of k_i, and a classical one is based on the two opposite angles α_i and β_i shared by e_i^I, e.g., α_6 and β_6 of e_6^I in Fig. 10, or specifically:

$$k_i = \frac{\cot\alpha_i + \cot\beta_i}{2} = \frac{\sin(\alpha_i + \beta_i)}{2\sin\alpha_i \sin\beta_i} \tag{8}$$

Although theoretically k_i could be negative when $\alpha_i + \beta_i > \pi$, this situation is rarely seen in our case since the Delaunay algorithm guarantees a good triangulation quality. By substituting Eq. (8) back into Eq. (7) and calculating the partial differential of each vertex coordinate, a large sparse linear equation system that contains $3 \times m$ unknowns can be constructed, i.e.:

$$\begin{cases} \dfrac{\partial}{\partial x_i^I} E(\{\boldsymbol{q}_i^I\}) = 0 \\ \dfrac{\partial}{\partial y_i^I} E(\{\boldsymbol{q}_i^I\}) = 0 \\ \dfrac{\partial}{\partial z_i^I} E(\{\boldsymbol{q}_i^I\}) = 0 \end{cases} \tag{9}$$

The non-trivial solution of this linear system gives us the final coordinates of the internal vertices with respect to the original 3D iso-contour.

A caution should be made: though iso-contours never intersect each other, the surface layers of two neighboring iso-contours could, especially if the $\Delta\delta^*$ between the two is very small and they are near a local maximum of δ^*, as shown in Fig. 11.

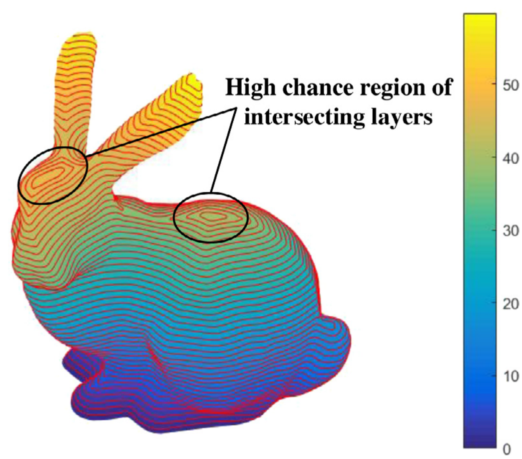

Fig. 11. Possible intersection between neighboring surface layers when they are close to a local maximum of δ^*.

2.3. Adaptive slicing of curved layers

In the simple 2.5-axis layered fabrication, the layer thickness is a constant since all layers are planar and parallel to each other. This obviously is no longer the case in multi-axis printing as iso-contours are 3D curves and their induced surface layers are 3D surfaces. To maintain a relative stable Hausdorff distance δ_H of two consecutive layers, it is best to choose a constant interval $\Delta\delta_0$ between two consecutive iso-contours. This would however cause intersecting layers in some critical regions. Therefore, an interrogation of the minimum thickness of the sandwich between two layers should be conducted once a new layer is formed. To facilitate the planning, we use the average distance (along some common direction) between any two neighboring surface layers to approximate the thickness of the sandwiched slice, i.e., $\overline{\tau} = \sum_{i=1}^{n} \tau_i / n$. In addition, the minimum thickness τ_{min} should be sufficient to enable a controllable deposition of fused filament, which is usually a dependent value τ_l with respect to the nozzle specification. To satisfy these two requirements, an adaptive geodesic interval $\Delta\delta$ should be sought for the generation of surface layers, starting from the bottom iso-contour l_0.

Referring to Fig. 12, suppose the previous surface layer of the iso-contour l_δ is already constructed. We first extract its most irrelevant principal direction $\boldsymbol{d_p}$ using the PCA method as introduced in Section 2.2. Given a geodesic interval $\Delta\delta$, the next iso-geodesic contour $l_{\delta+\Delta\delta} = \{\boldsymbol{q_i}\}$ is readily computed. Therefore, the average thickness between the two surface layers is:

$$\overline{\tau} = \frac{\sum_{i=1}^{n} \Delta\delta \cdot \sin\left(\cos^{-1}(\boldsymbol{d_p} \cdot \boldsymbol{n_i})\right)}{n} \tag{10}$$

where $\boldsymbol{n_i}$ is the unit surface normal at $\boldsymbol{q_i}$. This equation holds when $\Delta\delta$ is small and the geodesic can be approximated as a line segment. Based on this relationship, a tentatively qualified $\Delta\delta$ can be found once its resultant $\bar{\tau}$ equals the specified thickness.

As soon as the δ^* interval $\Delta\delta$ is determined, the iso-contour $l_{\delta+\Delta\delta}$ as well as the surface layer $S_{\delta+\Delta\delta}$ is fully constructed. We then compute the true minimum thickness $\tau_{min} = min(\tau_i)$ from each vertex in $S_{\delta+\Delta\delta}$ to S_δ and rectify $\Delta\delta$ if $\tau_{min} < \tau_l$ by the following equation:

$$\Delta\delta' = \Delta\delta \cdot \tau_l/\tau_{min} \tag{11}$$

With several iterations (usually less than three in our test examples), the final geodesic interval $\Delta\delta$ is determined satisfying both requirements. Below we summarize the entire curved layer decomposition process for a given solid model Ω with a flat bottom contour l_0.

Procedure **Curved_Layer_Decomposition** ($\partial\Omega$, l_0)

Input: The solid model represented by its boundary surface $\partial\Omega$ and the specified bottom iso-contour l_0

Output: A list of ordered surface layers $\{S_0, S_1, \dots, S_n\}$

Begin:

S_0 = the planar bottom base of l_0

Output-list = $\{S_0\}$

$\delta_{max} = \max_{\boldsymbol{q}\in\partial\Omega\setminus l_0} \delta^*(\boldsymbol{q}),\ \delta = 0$

while $\delta < \delta_{max}$

$\Delta\delta = \Delta\delta_0$

$\bar{\tau} = average\_thickness(S_\delta, l_{\delta+\Delta\delta})$

while $\bar{\tau} < \Delta\delta_0$

$\Delta\delta = \Delta\delta \cdot \Delta\delta_0/\bar{\tau}$

$\bar{\tau} = average\_thickness(S_\delta, l_{\delta+\Delta\delta})$

end while

$S_{\delta+\Delta\delta} = surface\_reconstruct(l_{\delta+\Delta\delta})$

$\tau_{min} = min\_thickness(S_\delta, S_{\delta+\Delta\delta})$

while $\tau_{min} < \tau_l$

$\Delta\delta = \Delta\delta \cdot \tau_l/\tau_{min}$

$S_{\delta+\Delta\delta} = surface\_reconstruct(l_{\delta+\Delta\delta})$

$\tau_{min} = min\_thickness(S_\delta, S_{\delta+\Delta\delta})$

end while

$\delta = \delta + \Delta\delta$

Append S_δ to Output-list

end while

Output Output-list.

End

---

3. Multi-axis printing path generation

Once the freeform solid model Ω is decomposed into sequential curved layers, each layer of the part should be eventually filled with filament by the printing nozzle along a specified trajectory. Controlling the width of deposition in its fused state is not an easy task; therefore, the step-over distance of printing path should be a constant, which is the primary regulation for path generation. On the other hand, since the layer thickness is variable due to the way we decompose the model, the material deposition should synchronize with the feed rate f of the nozzle to achieve variable thickness deposition. For example, upon a steady material depositing rate d_0, the feed rate needs to be inversely proportional to the local thickness so as to guarantee a correct deposition. Due to this consideration, sharp turnings on the printing path would lead to constant acceleration and deceleration of the nozzle, making it hard to control the feed rate. Therefore, a smooth path is desired, which is the second requirement for path generation.

According to these requirements, the widely adopted direction-parallel path may not be qualified in this specific task. Alternatively, we employ the well-known pattern of contour-parallel curves to facilitate our path generation, which maintains a constant step-over distance between adjacent iso-contours on the surface layer. Specifically, for each layer consisting of meshed triangles, the geodesic distance field with respect to the boundary contour is constructed based on the MMP method (see Fig. 13(b)), which is similar to the computation of SCGDF in Section 2.1, except that the bottom layer l_0 now becomes the boundary curve. The level-set contours induced from this geodesic field are exactly the desired trajectory for fabricating the layer.

An intuitive choice for nozzle orientation along the path seems to be the one aligned with the surface normal, which works perfectly for the inner part of the layer. However, when the upper layer is overhanging over the one beneath (see Fig. 14), faulty deposition is likely to occur around the layer boundary if the nozzle orientation is still fixed along the surface normal of the layer. To ensure a stable deposition, for each point $\boldsymbol{q}$ on the path in layer l_{i+1}, we search for the closest point $\boldsymbol{q}'$ in the lower layer l_i such that the vector formed by these two points $\overrightarrow{\boldsymbol{q}'\boldsymbol{q}}$ identifies the nozzle orientation at $\boldsymbol{q}$. This strategy automatically adapts the orientation to the surface normal when there is no overhanging feature, while inclining the nozzle when there is. Once the nozzle orientation is determined for every point on the path, a moving average filter is applied to smooth the orientation vectors. Specifically, for each nozzle orientation vector, we take out its previous and the following two vectors and project them onto a Gaussian sphere, as shown in Fig. 15. The centroid of the pentagon formed by these five points is calculated and taken to be the final orientation. Fig. 16 demonstrates this smoothing operation, which is essential and highly effective in improving the dynamic performance of multi-axis printing.

Once the trajectory as well as the nozzle orientations is all settled, the volume of the deposited material per unit time needs to be cautiously synchronized along with the nozzle's movement, accounting for the fact that the layer thickness is no longer uniform but variable. This volumetric rate is defined as the *Material Depositing Rate* (MDR), akin to the material removal rate (MRR) in machining process. In traditional 2.5-axis fabrication, the MDR is normally a fixed constant. However, in multi-axis printing, a properly assigned MDR along the path can effectively reduce the level of porosity and enhance the mechanical properties of the part. In FDM, the MDR is fully controlled by the material feed rate f^m assigned to the feeder, as shown in Fig. 17. Suppose the desired layer thickness at a particular point $\boldsymbol{q_i}$ along the nozzle trajectory is τ_i, which is the shortest distance to the previous

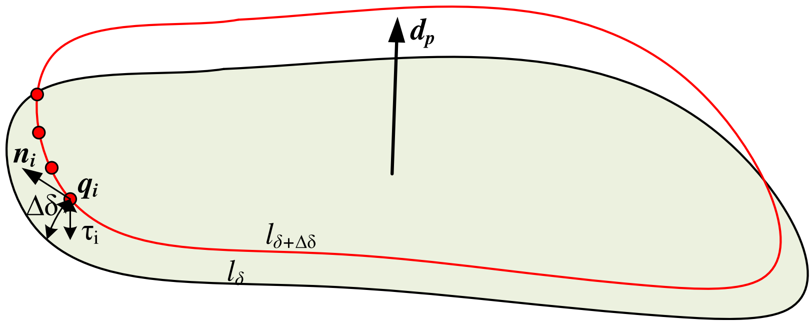

Fig. 12. Determination of the initial geodesic interval.

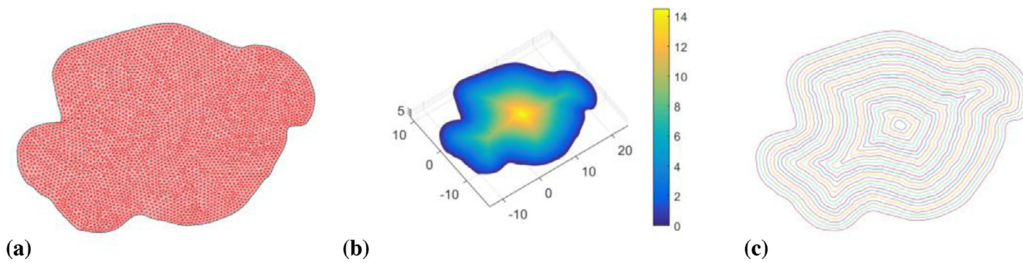

Fig. 13. Contour-parallel path generation based on the geodesic distance field of the curved layer.

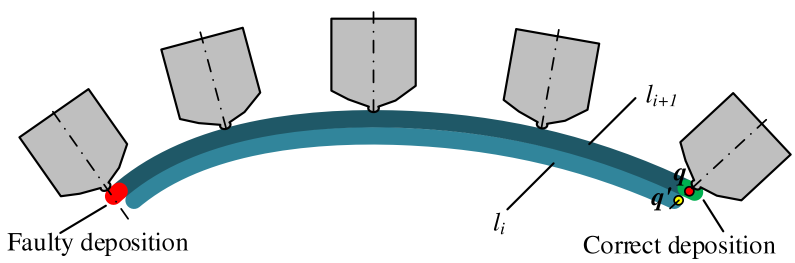

Fig. 14. Faulty deposition on the overhanging layer.

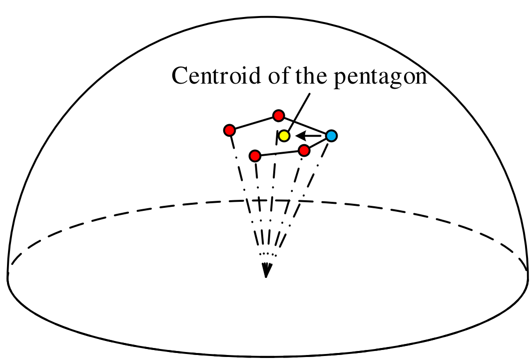

Fig. 15. Orientation smoothing via a moving average filter.

layer, and the step-over of the path is a constant s_d; the following volume conservation should be satisfied in order to make a proper deposition:

$$f_i^m \pi r_m^2 = s_d \tau_i f_i \tag{12}$$

where r_m is the radius of the filament, f_i is the assigned feed rate of the nozzle at $\boldsymbol{q_i}$. The computation of the layer thickness τ_i at $\boldsymbol{q_i}$ is as simple as to calculate the minimum distance between $\boldsymbol{q_i}$ and the previous surface layer. Accordingly, the variable material feed rate f^m can be numerically computed for each point of the path.

It needs to be reiterated that the minimum thickness of each decomposed layer is satisfied after calling Procedure **Curved_Layer_Decomposition**. However, though rare, there is chance that the layer thickness could exceed the upper bound τ_u (i.e. in our case $\tau_u = s_d$), which violates the *Rq2* in Section 2. Even though the material deposition volume is controllable by adjusting the material feed rate f^m, the final shape of the fused material is hard to preserve once the layer thickness is too large. To cope with this issue, we further rectified the fabricating process by dividing the unqualified layer into multiple sub-layers, as depicted in Fig. 18. For the example demonstrated in Fig. 18, the layer is eventually divided into two sub-layers whose thickness satisfies *Rq2*. In practice, instead of making additional layer decomposition, we rectify the material feed rate f^m of the path to achieve controllable material deposition. For the case in Fig. 18, it essentially needs two rounds of fabrication to sequentially construct sub-layer1 and sub-layer2, with rectified material feed rate accordingly to each printing path. Fig. 18(c) shows the stacking two-layer paths for fabricating one curved layer with varying but qualified f^m. Note that this case is extremely rare in the testing models to be demonstrated later, as most decomposed layers by our algorithm still exhibit controllable thickness.

By now, we have described, for each curved layer, how to generate a complete printing path consisting of the nozzle trajectory, the nozzle orientation, and the proper MDR along the path. Once the printing path is generated, a post-processing module developed by us converts the path to an improved G*-code dedicated specifically for the fabrication of non-uniform curved layers.

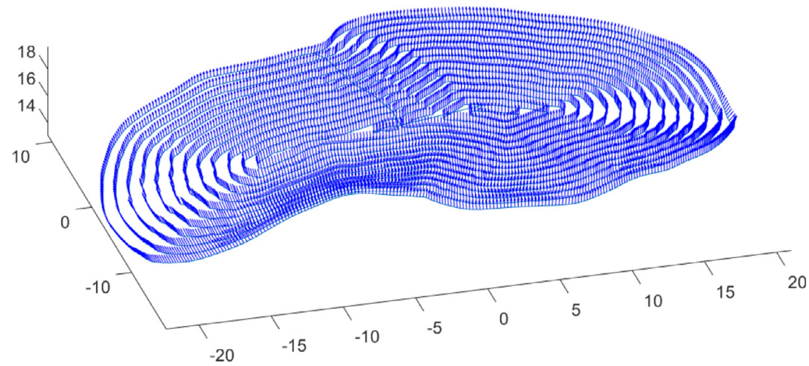

Fig. 16. Example of a printing path and the smoothed nozzle orientation vectors.

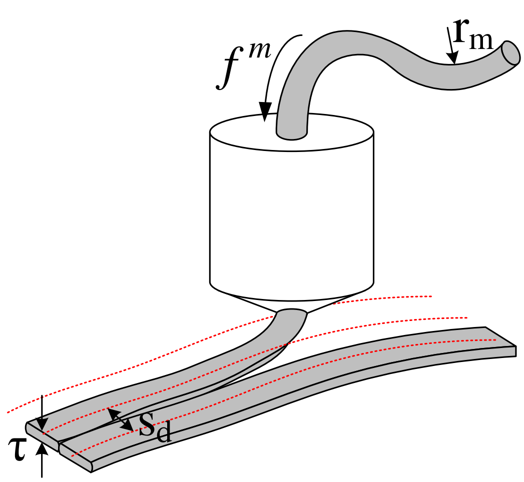

Fig. 17. The material depositing rate.

In this new G*-code format, each main command line consists of three parts: the first three entries indicate the position of nozzle; the next three entries represent the orientation of nozzle represented in the rotation vector form, and the last one indicates the accumulated feed value $E_j = \sum_{i=1}^{j} f_i^m$, which is determined by the variable material feed rate f^m according to Eq. (12). Fig. 19 shows a snippet of our G-code*.

Together with the procedure **Curved_Layer_Decomposition** to generate a suitable set of curved layers, we are now ready to present our final process planning algorithm for multi-axis printing of an arbitrary solid model Ω with a prescribed base contour l_0, as follows.

Step 1. Call procedure **Curved_Layer_Decomposition** to generate a set of curved layers $\{S_0, S_1, \ldots, S_n\}$ that are ordered on the δ^* values of their iso-contours.

Step 2. For each S_i, generate a complete multi-axis printing path made of the nozzle trajectory, the nozzle orientation, and the proper MDR along the path.

Step 3. Starting from S_0, executing the printing path, one by one, until the last one S_n. In case there are two (or more) consecutive layers S_i and S_{i+1} that have the same δ^* value, their execution order is arbitrary.

4. Results and discussion

The proposed curved layer decomposition as well as the multi-axis printing path generation algorithms has been implemented by us in MATLAB. In addition, for the purpose of physical validation of the proposed algorithms, a prototype multi-axis FDM hardware system is also developed, which integrates a FDM extruder located on an X–Y table with a 6-DOF UR5 robotic arm on which the printing bed is mounted. While the total DOF of the system is redundant, the five-axis printing motion defined in the workpiece frame is decoupled into X–Y–Z motions and rotary motions which are synchronized to realize the physical fabrication. The X–Y translational motions are controlled by the X–Y table, while the robotic arm is responsible for the Z motions and the rotary motions. A robotic operation system (ROS) is employed to control and realize the synchronization of all the axes to enable continuous printing for layers with variable thickness. The G*-code is pre-processed and transformed to the corresponding machine commands and distributed to the Marlin firmware and UR5 controller during each time cycle (see Fig. 20 for detailed architecture of the prototype system).

According to the aperture of our nozzle whose diameter is 1 mm, the step-over distance of the printing path s_d is set to 1 mm as default, while the initial layer thickness τ_0 is set to 0.5 mm. The Stanford Bunny, the Kitten, a multi-branch model and a genus-one model are selected as representative models for testing. Shown in Figs. 21 to 24 are respectively the corresponding curved layer decomposition and the printing path of each testing model. For the multi-branch and the genus-one model, we have further performed physical printing on our prototype multi-axis printer using the generated printing path and the fabricated parts are shown in Figs. 25 and 26. It is plausible to note that both models would require a large volume of support structure on a 3-axis printer, or even a 3+2-axis printer. But on a five-axis

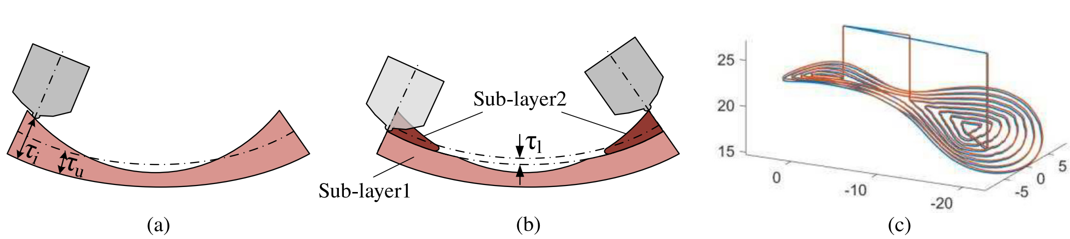

Fig. 18. Rectification of one layer process with excessive thickness: (a) original layer; (b) rectified sub-layers with controllable thickness; (c) an example of one layer fabrication consisting of two layers of printing paths.

```
;FLAVOR:Marlin & UR5
;LAYER:1
G1 F200 X-41.712354 Y10.634942 Z16.429993 RX0.150959 RY-1.193465 RZ0.102897 E0.009329
G1 X-41.691143 Y10.510083 Z16.425194 RX0.150460 RY-1.192255 RZ0.102420 E0.018658
G1 X-41.670305 Y10.385151 Z16.420634 RX0.149185 RY-1.191211 RZ0.101433 E0.027987
G1 X-41.649235 Y10.260257 Z16.416075 RX0.147701 RY-1.190135 RZ0.100300 E0.037316
G1 X-41.627761 Y10.135434 Z16.411460 RX0.146564 RY-1.188947 RZ0.099396 E0.046646
G1 X-41.606517 Y10.010565 Z16.407063 RX0.146320 RY-1.187804 RZ0.099106 E0.055976
```

Fig. 19. A snippet of our G*-code for fabricating non-uniform curved layers.

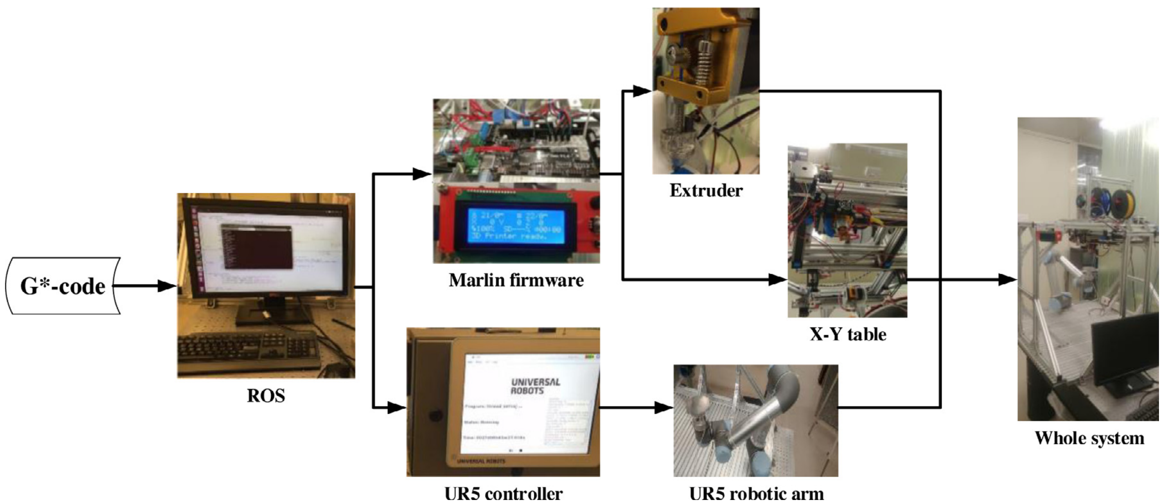

Fig. 20. A prototype multi-axis FDM system.

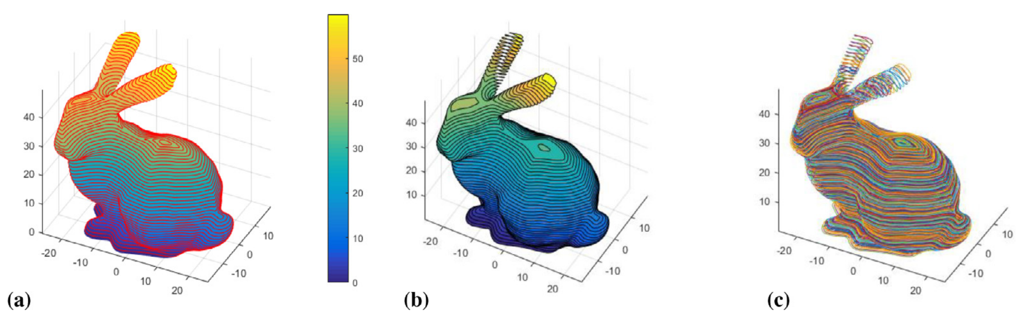

Fig. 21. Simulation results of the Stanford Bunny model: (a) the iso-contours, (b) the curved layers and (c) the accumulated printing path (trajectory only).

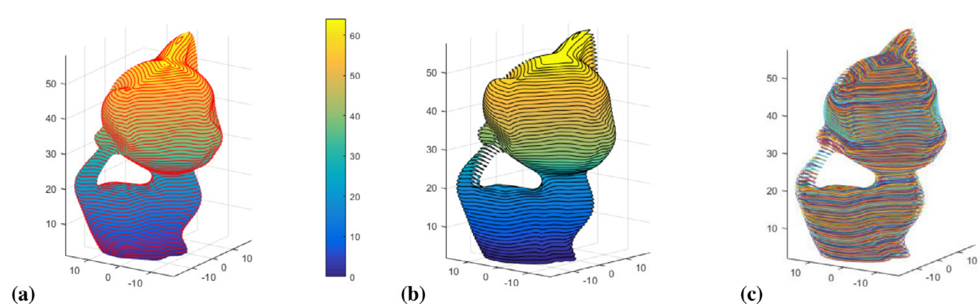

Fig. 22. Simulation results of the Kitten model.

printer using our process planning algorithm, they can be successfully fabricated without any support structure.

In Table 1, we give the model complexity as well as the computing time of our tests on a desktop PC (i7 6770k, 8 GB ram). It is noted that nearly two thirds of the total computing cost is spent on the layer decomposition process, which takes up to 693 s for handling a mesh model with over 10^4 facets. It should be clarified that the most time consuming process in our layer decomposition is actually the calculation of thickness between two consecutive layers. Since no special data structure is utilized in this proof-of-concept implementation, the time complexity for computing the minimum distance between two layers is theoretically $o(n^2)$ using brutal force, which is also confirmed by the data in Table 1.

Finally, to prove that the proposed curved layer decomposition scheme can substantially reduce the need of support structure (if not completely eradicated), we analyze the HT ratio for each

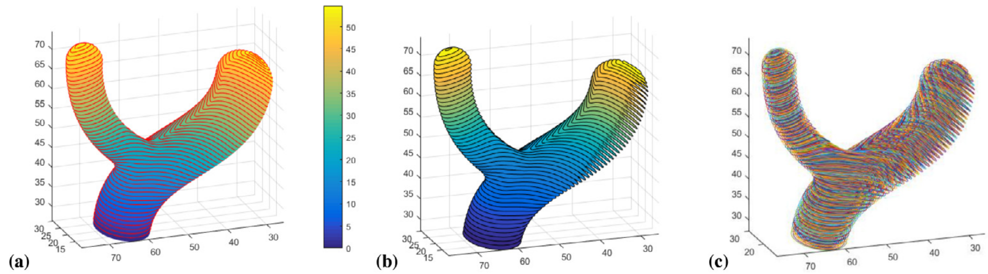

Fig. 23. Simulation results of the multi-branch model.

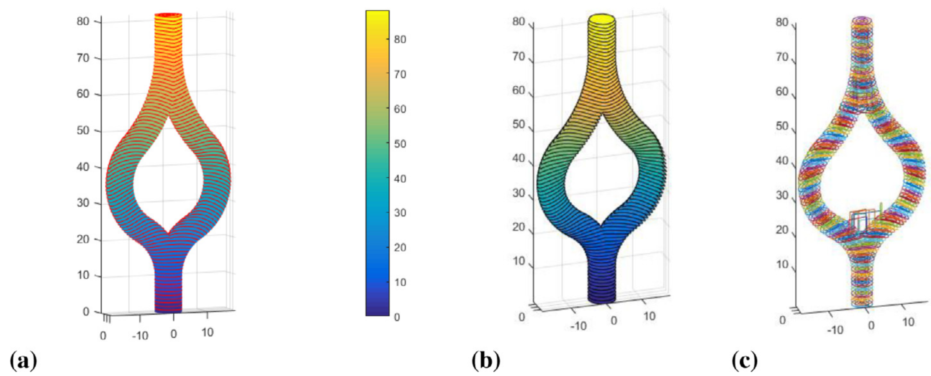

Fig. 24. Simulation results of the genus-one model.

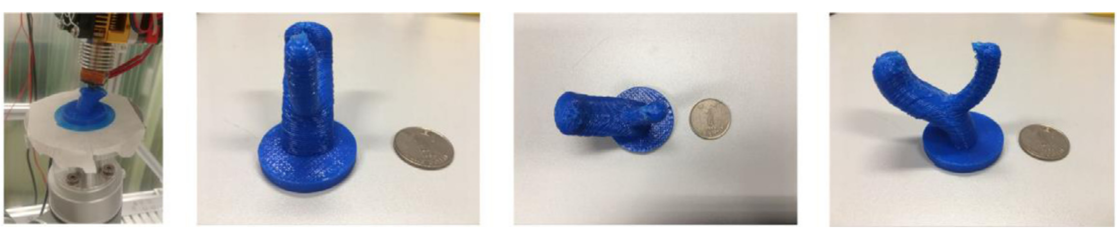

Fig. 25. Physical fabrication of the multi-branch model.

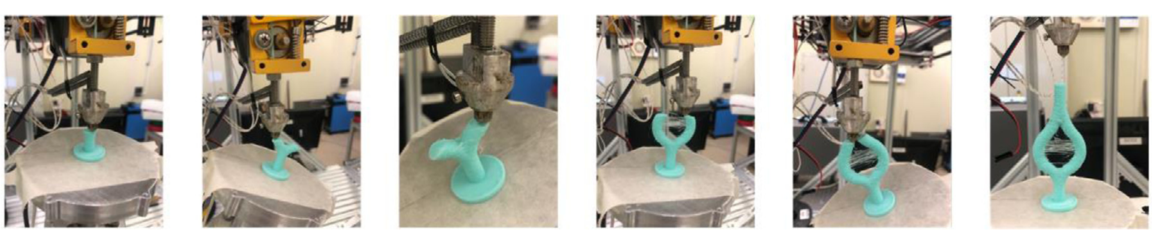

Fig. 26. Physical fabrication of the genus-one model.

Table 1
Computing statistics of the proposed algorithms on the testing models.

Model	Number of facets	Time for curved layer decomposition (s)	Time for printing path generation (s)
Bunny	13 000	693	208
Kitten	8 260	310	106
Multi-branch	6 496	157	61
Genus-one	4 170	85	64

decomposed layer and compare the result with the conventional planar layer decomposition with the same total number of layers, as given in Fig. 27. Note that the HT ratio as derived in Section 2 directly reflects the need of support structure during the fabricating process. It can be observed in Fig. 27 that the HT ratio distribution of curved layers is successfully allocated inside an acceptable range ($\delta_H/\tau < 1.5$), except for the Kitten model which will be explained later, while the HT ratio for planar layers can be as large as 4 (see the Bunny case), which obviously needs support structure prior to the fabrication. The outcome of HT ratio distribution indicates the effectiveness of our proposed curved layer decomposition scheme, which largely avoids the need of support structure when fabricating complex models. There is one exception for the Kitten case that the HT ratio seems to exceed the threshold. A preliminary explanation is that the Kitten model is of non-zero genus and with drastic change of curvature. We will further investigate this issue and see if a variant of the proposed scheme can handle such case.

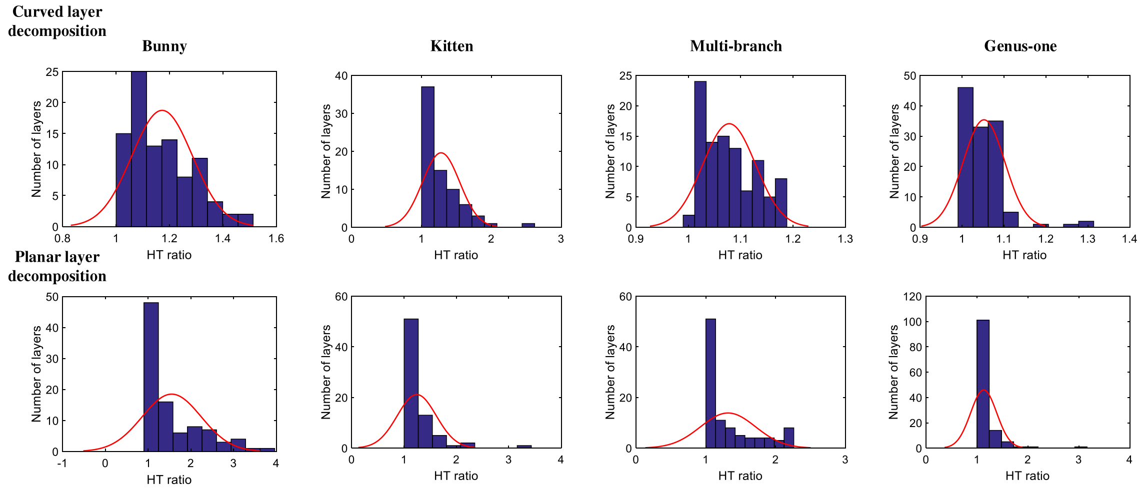

Fig. 27. The histograms showing the HT ratio of the four examples using the curved layer and planar layer decomposition.

5. Conclusion and future work

By taking advantage of the multi-axis printing configuration, we have proposed a general process planning scheme for five-axis fabrication of freeform solids, based on the principle of curved layer decomposition. The core of the proposed scheme is the establishment of the geodesics on surface, from which well-defined and non-intersecting iso-geodesic contours can be extracted. These contours are then filled with triangles with the minimized Dirichlet energy to form curved surface layers of a natural shape. With a contour-parallel printing path generated on each layer and following the strict increasing order of iso-geodesics to print the layers, an entire multi-axis fabricating process is fully determined.

There are two potential limitations of the current scheme. First, and most critically, is the potential local interference when fabricating the curved layers (see Fig. 28(a)). The convexity of surface layers is not guaranteed since they are purely constructed based on the Harmonic mapping principle. While the iso-contours as well as its internal surface layer are deterministically constructed, local interference may happen due to the concavity feature of the later constructed surface layer. The chance of this glitch increases at those areas where the curvature of the model profile changes drastically, leading to twisted iso-contours and hence curvy surface layers. This issue can be alleviated by adopting a slender nozzle head, while a better resolution by devising a convexity-preserved surface layer construction algorithm will be our next target.

The second limitation of the proposed strategy is the low productivity encountered during the physical printing experiments. As the decomposed layers are non-planar, nozzle orientation keeps changing along its working trajectory when fabricating each layer. This constant change of orientation tremendously increases the overall processing time. There are two proposals to handle this issue. The first one is to incorporate more advanced path smoothing scheme into the printing path generation stage, such as the one from our early work [31]. A smoothed multi-axis path leads to a better execution of the machine axes, and hence a shorter time. The other idea is to roughly decompose the model into curved layers, find an adequate build direction for the intermediate part between two consecutive layers, and further decompose this part into planar layers along this fixed build direction. We will conduct more exploration into this in the future.

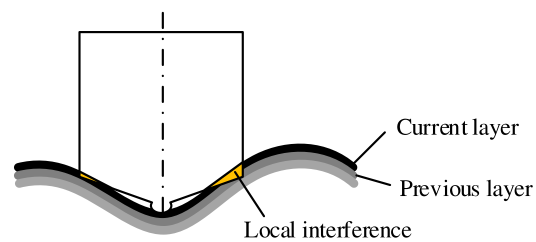

Fig. 28. Illustration of local interference between the nozzle and the part.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Grant No. 51805260).

References

[1] Gao W, Zhang Y, Ramanujan D, Ramani K, Chen Y, Williams CB, Wang CC, Shin YC, Zhang S, Zavattieri PD, status The, challenges. The status challenges and future of additive manufacturing in engineering. Comput Aided Des 2015;69:65–89.
[2] Jiang J, Xu X, Stringer J. Support structures for additive manufacturing: A review. J Manuf Mater Process 2018;2:64.
[3] Mohan Pandey P, Venkata Reddy N, Dhande SG. Slicing procedures in layered manufacturing: a review. Rapid Prototyp J 2003;9:274–88.
[4] Frank D, Fadel G. Expert system-based selection of the preferred direction of build for rapid prototyping processes. J Intell Manuf 1995;6:339–45.
[5] Dolenc A, Mäkelä I. Slicing procedures for layered manufacturing techniques. Comput Aided Des 1994;26:119–26.
[6] Cao W, Miyamoto Y. Direct slicing from AutoCAD solid models for rapid prototyping. Int J Adv Manuf Technol 2003;21:739–42.
[7] Siraskar N, Paul R, Anand S. Adaptive slicing in additive manufacturing process using a modified boundary octree data structure. J Manuf Sci Eng 2015;137. 011007.
[8] Wang W, Chao H, Tong J, Yang Z, Tong X, Li H, Liu X, Liu L. Saliency-preserving slicing optimization for effective 3D printing. In: Computer graphics forum. Wiley Online Library; 2015, p. 148–60.
[9] Wulle F, Coupek D, Schäffner F, Verl A, Oberhofer F, Maier T. Workpiece and machine design in additive manufacturing for multi-axis fused deposition modeling. Procedia CIRP 2017;60:229–34.
[10] Pan Y, Zhou C, Chen Y, Partanen J. Multitool and multi-axis computer numerically controlled accumulation for fabricating conformal features on curved surfaces. J Manuf Sci Eng 2014;136. 031007.
[11] Fichter EF. A stewart platform-based manipulator: general theory and practical construction. Int J Robot Res 1986;5:157–82.

[12] Gao W, Zhang Y, Nazzetta DC, Ramani K, Cipra RJ. Revomaker: Enabling multi-directional and functionally-embedded 3D printing using a rotational cuboidal platform. In: Proceedings of the 28th annual ACM symposium on user interface software & technology. ACM; 2015, p. 437–46.

[13] Chen X, Zhang H, Lin J, Hu R, Lu L, Huang Q-X, Benes B, Cohen-Or D, Chen B. Dapper: decompose-and-pack for 3D printing. ACM Trans Graph 2015;34:213:211–2.

[14] Attene M. Shapes in a box: Disassembling 3D objects for efficient packing and fabrication. In: Computer graphics forum. Wiley Online Library; 2015, p. 64–76.

[15] Keating S, Oxman N. Compound fabrication: A multi-functional robotic platform for digital design and fabrication. Robot Comput-Integr Manuf 2013;29:439–48.

[16] Wu C, Dai C, Fang G, Liu Y-J, Wang CC. Robofdm: A robotic system for support-free fabrication using FDM. In: Robotics and automation (ICRA), 2017 IEEE international conference on. IEEE; 2017, p. 1175–80.

[17] Zhang J, Liou F. Adaptive slicing for a multi-axis laser aided manufacturing process. Trans-Amer Soc Mech Eng J Mech Des 2004;254–61.

[18] Lee K, Jee H. Slicing algorithms for multi-axis 3-D metal printing of overhangs. J Mech Sci Technol 2015;29:5139–44.

[19] Xu K, Chen L, Tang K. Support-free layered process planning toward 3+ 2-axis additive manufacturing. IEEE Trans Autom Sci Eng 2018;1–13.

[20] Ding D, Pan Z, Cuiuri D, Li H, Larkin N, Van Duin S. Automatic multi-direction slicing algorithms for wire based additive manufacturing. Robot Comput-Integr Manuf 2016;37:139–50.

[21] Wei X, Qiu S, Zhu L, Feng R, Tian Y, Xi J, Zheng Y. Toward support-free 3D printing: A skeletal approach for partitioning models. IEEE Trans Visualizat Comput Graph 2018;24:2799–812.

[22] Coupek D, Friedrich J, Battran D, Riedel O. Reduction of support structures and building time by optimized path planning algorithms in multi-axis additive manufacturing. Procedia CIRP 2018;67:221–6.

[23] Ezair B, Fuhrmann S, Elber G. Volumetric covering print-paths for additive manufacturing of 3D models. Comput Aided Des 2018;100:1–13.

[24] Dai C, Wang CC, Wu C, Lefebvre S, Fang G, Liu Y-J. Support-free volume printing by multi-axis motion. ACM Trans Graph 2018;37:134.

[25] Chen L, Chung M-F, Tian Y, Joneja A, Tang K. Variable-depth curved layer fused deposition modeling of thin-shells. Robot Comput-Integr Manuf 2019;57:422–34.

[26] Leary M, Babaee M, Brandt M, Subic A. Feasible build orientations for self-supporting fused deposition manufacture: a novel approach to space-filling tesselated geometries. In: Advanced materials research. Trans Tech Publ; 2013, p. 148–68.

[27] Mitchell JS, Mount DM, Papadimitriou CH. The discrete geodesic problem. SIAM J Comput 1987;16:647–68.

[28] Surazhsky V, Surazhsky T, Kirsanov D, Gortler SJ, Hoppe H. Fast exact and approximate geodesics on meshes. In: ACM Trans Graph (TOG). Acm; 2005, p. 553–60.

[29] George PL, Seveno E. The advancing-front mesh generation method revisited. Internat J Numer Methods Engrg 1994;37:3605–19.

[30] Rebay S. Efficient unstructured mesh generation by means of Delaunay triangulation and Bowyer-Watson algorithm. J Comput Phys 1993;106:125–38.

[31] Xu K, Li Y. Region based five-axis tool path generation for freeform surface machining via image representation. Robot Comput-Integr Manuf 2019;57:230–40.