Support-Free Volume Printing by Multi-Axis Motion

CHENGKAI DAI, Delft University of Technology, The Netherlands
CHARLIE C. L. WANG*, Delft University of Technology, The Netherlands
CHENMING WU, Tsinghua University, China
SYLVAIN LEFEBVRE, INRIA, France
GUOXIN FANG, Delft University of Technology, The Netherlands
YONG-JIN LIU, Tsinghua University, China

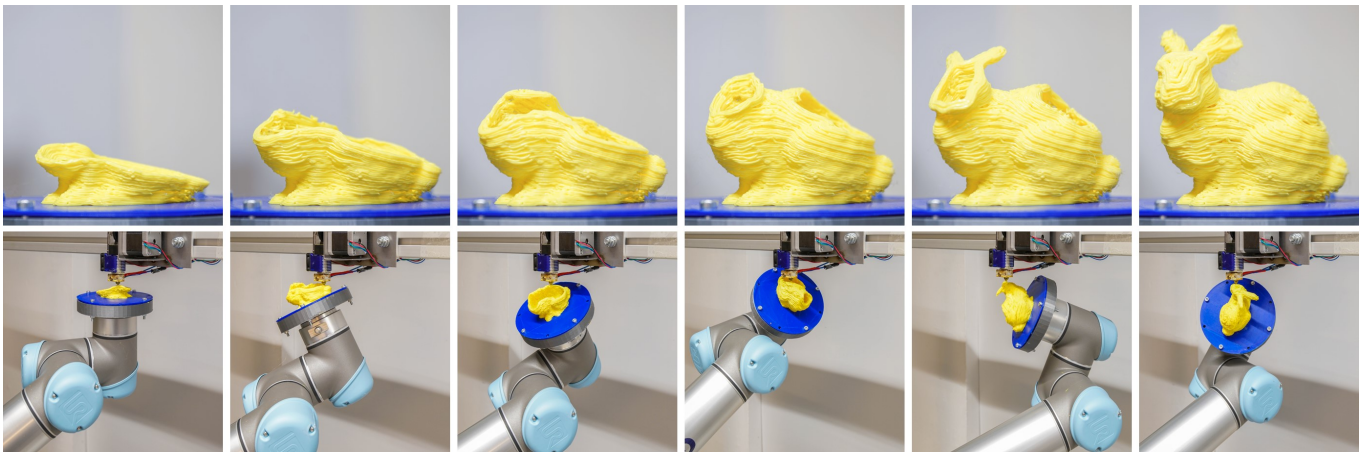


Fig. 1. Our method enables support-free 3D printing of solid models. By exploiting all 6 degrees of freedom (translations, rotations) and depositing material along *curved* layers, we make support structures unnecessary in most cases. This increases further the flexibility offered by 3D printing, such as freeing designers from support constraints on complex parts.

This paper presents a new method to fabricate 3D models on a robotic printing system equipped with multi-axis motion. Materials are accumulated inside the volume along curved tool-paths so that the need of supporting structures can be tremendously reduced – if not completely abandoned – on all models. Our strategy to tackle the challenge of tool-path planning for multi-axis 3D printing is to perform two successive decompositions, first volume-to-surfaces and then surfaces-to-curves. The volume-to-surfaces decomposition is achieved by optimizing a scalar field within the volume that represents the fabrication sequence. The field is constrained such that its iso-values represent curved layers that are supported from below, and present a convex surface affording for collision-free navigation of the printer head. After extracting all curved layers, the surfaces-to-curves decomposition covers them with tool-paths while taking into account constraints from the robotic printing system. Our method successfully generates tool-paths for 3D printing models with large overhangs and high-genus topology. We fabricated several challenging cases on our robotic platform to verify and demonstrate its capabilities.

CCS Concepts: • **Computing methodologies → Shape modeling**;

Additional Key Words and Phrases: 3D printing, multi-axis motion, supporting structures, tool-path generation

ACM Reference Format:
Chengkai Dai, Charlie C. L. Wang, Chenming Wu, Sylvain Lefebvre, Guoxin Fang, and Yong-jin Liu. 2018. Support-Free Volume Printing by Multi-Axis Motion. *ACM Trans. Graph.* 37, 4, Article 1 (August 2018), 14 pages. https://doi.org/10.1145/3197517.3201342

1 INTRODUCTION

The prompt development of *additive manufacturing* (AM) techniques has motivated significant research effort in the area of computer graphics, computer-aided design, biomedical engineering and robotics (e.g., [Liu et al. 2014; Shamir et al. 2016]). Although it is called 3D printing, the fabrication in most commercial systems is still taken in a 2.5D manner – materials are accumulated layer upon layer in planes along a fixed printing direction. This significantly reduces the complexity and development cost of both hardware and software. However, this introduces additional manufacturability constraints,

*Corresponding author: c.c.wang@tudelft.nl (Charlie C. L. Wang)

Authors' addresses: Chengkai Dai, Delft University of Technology, Delft, The Netherlands; Charlie C. L. Wang, Delft University of Technology, Delft, The Netherlands; Chenming Wu, Tsinghua University, Beijing, China; Sylvain Lefebvre, INRIA, France; Guoxin Fang, Delft University of Technology, Delft, The Netherlands; Yong-jin Liu, Tsinghua University, Beijing, China.

0730-0301/2018/8-ART1 $15.00
https://doi.org/10.1145/3197517.3201342

in particular requiring the addition of supporting structures below overhangs [Dumas et al. 2014; Hu et al. 2015]).

Various robotic fabrication systems have been introduced in recent years (e.g., [Mueller et al. 2014; Song et al. 2015; Volker et al. 2015]). They provide additional *degree-of-freedom* (DOF) in motion so that the direction of material accumulation can be changed during fabrication. However, planning for a global fabrication sequence without collisions is challenging, and most existing approaches can only deal with models of relatively simple shapes (see Section 2) – although theoretically the method of Huang et al. [2016] can be extended to handle large general models.

The challenge stems from the large size of the motion configuration space: deposition paths can follow arbitrary curves in space, exploiting all 6DOFs. Unlike multi-axis milling, that mainly focuses on forming the surface of a model by cutting materials, multi-axis AM faces the problem of *filling* the volume with evenly spaced, non overlapping trajectories, which are always deposited on top of an already solidified volume (no isolated 'floating' component), and that do not result in collisions during motion.

Compared to conventional 3D printing, this change from planar to arbitrarily curved layers tremendously increases the complexity of computations. While constraining layers to be planar leads to a well defined slicing problem, the additional freedom introduced by curved layers makes it challenging to even define what the geometry of the layers should be. In addition, a feasible solution has to take into account geometric constraints as well as hardware constraints.

In this paper, we present a new methodology to tackle the challenge of multi-axis AM tool-path generation. Our technique is based on the observation that the dimensionality of the problem can be successively reduced by first decomposing the volume into sequences of curved surface layers, and then decomposing each surface into curved tool-paths. Our algorithm searches for an accumulation sequence, which is collision-free, ensures always supported material deposition, and can print all regions as much as possible. Curved surface layers are covered with tool-paths taking into account hardware constraints.

Contributions:

- A novel approach for support-free multi-axis printing, that decouples the problem into first extracting curved surface layers, and second covering each surface with curved tool-paths, successively reducing the dimensions of the problem.
- An algorithm based on the computation of a scalar field representing the accumulation sequence of material within the shape during the AM process. The field is carefully constructed such that layers are convex and collision-free, supported by previous layers, and – as much as possible – do not prevent *future* layers to be accessed.
- The covering of curved layers with smooth tool-paths, optimizing both positions and orientation according to the constraints of the 6DOF robot realizing the motions.

We demonstrate our approach on a variety of models in both computational and physical experiments, fabricating actual objects using the tool-paths generated by our approach to drive a 6DOF filament based 3D printing platform. To the best of our knowledge, this is the first multi-axis 3D printing approach that can fabricate *general* volume models with minimal supporting structures.

2 RELATED WORK

Since its invention in the late 1980s, the core principle of AM remained largely based on the accumulation of planar layers along a single build direction [Gibson et al. 2015]. Active research in the area focuses on the use of multi-materials, faster printing and increased deposition flexibility [Gao et al. 2015b]. In particular, a recent trend is to exploit additional motion DOFs, moving away from the current limitations of planar material accumulation.

2.1 3DOF additive manufacturing

The first attempt of using non-planar layers in AM was made a decade ago in an approach called the *Curved Layer Fused Deposition Modeling* (CLFDM) [Chakraborty et al. 2008]. It departs from standard FDM fabrication by dynamically changing z-values within individual layers. Recently, such motions have been realized on a Delta style FDM printer [Llewellyn-Jones et al. 2016]. A shell model is fabricated by depositing a double-curved layer on top of a sandwich structure printed with planar layers. As a result, the surface of the 3D printed model does not exhibit the staircase effect. However, this approach is limited to models with relative simple shapes – i.e., height-fields facing up along the z-axis. In addition, regions with a steep slope lead to local collision between deposited materials and the printer-head. A most recent effort was paid to generate 3-axis motion tool-paths inside a given volume [Ezair et al. 2018], which is also suffered from gouging.

2.2 5DOF additive manufacturing

Keating and Oxman [2013] introduced a FDM based proof-of-concept printing, showing how exploiting all 6DOFs of a robotic arm can improve the 3D printing process. The demonstration is however limited to simple shapes (e.g., cubes, torus and cylindrical surfaces) and there are no details regarding tool-path generation. Interestingly, the extrusion nozzle is fixed while the robot moves the part below. We use a similar setup (see supplemental material). This increases filament adhesion with the help of gravity, in contrast to moving a printer-head around a fixed part (e.g., [Peng et al. 2016]). Pan et al. [2014] developed a five-axis motion system similar to CNC machining, accumulating materials onto an existing model. The tool-path planning algorithm only handles specific cases and relatively simple components.

Recently, researchers have focused on printing wire mesh models using 5DOF. Such models are fabricated edge by edge, using freeform motions. Wu et al. [2016] compute collision-free tool paths for this purpose. A naïve ordering can lead to a configuration where some edges cannot be approached anymore. To tackle this challenge, a global sequence planning is formulated on a directed graph. Huang et al. [2016] further considers stability constraints jointly with the collision-free constraints. Both these approaches detect collisions in the optimization loop, which is time-consuming. As a result, only wire meshes with small number of primitives can be considered (e.g., less than 1k struts in [Huang et al. 2016; Wu et al. 2016]). This drawback prevents applying these algorithms to large scale

problems (e.g., the Bunny model in Fig.1 has 97.5k voxels with 0.8mm width – relevant to the nozzle's diameter).

High-DOF robotic devices have been extensively used in composite fabrication (e.g., aeronautical industry [Marsh 2011]). However, specifying the tool-paths for placing composite tapes on curved surface often requires an intensive manual work. Our ambition is to automate the tool-path generation for high-DOF 3D printing on general models, which is a critical step for direct digital manufacturing. None of the existing approaches investigates a method to produce curvilinear tool-paths within the volume of a part.

2.3 Volume decomposition for fabrication

In another thread of research, volume decomposition has been used to enable the fabrication in different scenarios. Luo et al. [2012] decompose large models into smaller pieces so that they can be fabricated on 3D printers with limited working envelopes. Other methods decompose a given model into height-fields [Herholz et al. 2015] or pyramid-based shapes [Hu et al. 2014] so that they can be fabricated by molding or support-free 3D printing. To fabricate large models, Song et al. [2016] decompose a volume into a set of large-core-supporting height-field pieces that are 3D printed. These approaches require a manual assembly step and the final parts present fragilities along assembly surfaces.

Rotational motions have also been used to avoid manual assembly. In [Gao et al. 2015a], material accumulation is applied around a cubic component, printing a 3D model on top of an existing object. Only cubic shapes are considered as the cores. Wu et al. [2017] propose an algorithm to segment a model into support-free parts, each fabricated by a robotic arm using planar layers.

All these methods still rely on planar layers, which constrains both the decomposition and the complexity of the parts that can be handled. In this paper, we investigate a more general curved layer decomposition method for 5DOF volume printing.

2.4 Accessibility for machining

Determining accessibility remains a challenging aspect of multi-axis tool-path planning, despite its extensive study in the context of CNC milling. Algorithms include the visibility map [Elber 1994] to analyze accessibility, as well as approaches that detect and avoid collisions between tools and workpieces [Ilushin et al. 2005]. Recent work focus on computing gouging-free tool-paths (i.e., no over-cut caused by local interference between tool and workpiece) while also optimizing the dynamic behavior of machines [Kim et al. 2015; Wang and Tang 2007]. The surface accessibility of a given model has also been widely studied in other areas such as for molding [Liu et al. 2009], and for inspection and computer-assisted surgery [Zhang et al. 2015]. In general, the computation of accessibility is very time-consuming. Our problem is even more challenging – we seek to decompose a volume into a *sequence of accessible* surfaces with nearly uniform thickness.

Our attempts at using existing state-of-the-art collision detection techniques (i.e., the *Flexible Collision Library* [Pan et al. 2012]) were discouraging. For example, the candelabra model in Fig.2 has 186,735 voxels using a voxel dimensions of 0.8mm, based on the nozzle diameter. Given a sequence that adds voxels one by one, the collision detection step alone – that incrementally adds voxels and checks for collisions – can take up to *96 hours* in total. However, to find a support-free tool-path a large number of such possible sequences have to be checked. A brute force approach could not be computed in any feasible amount of time. Instead, we propose a new method to *maintain* an accessible working surface while progressively constructing the sequence of material accumulation.

In summary, advanced AM hardware capable of multi-axis motions cannot be fully utilized at present, for lack of effective tool-path planning algorithms. Although the techniques developed for CNC milling are relevant, going from surface machining to volume filling for AM tremendously increases the complexity and the difficulty of the related geometric problems.

3 OVERVIEW OF 5DOF SUPPORT-FREE 3D PRINTING

To tackle the challenge of tool-path generation for 5DOF volume printing, we propose a novel approach based on a *dimension reduction* strategy. As illustrated in Fig.2, a given model will be first decomposed into a valid sequence of curved layers that are manufacturable (3D to 2D), which are then further decomposed into curved tool-paths (2D to 1D).

There are many possible choices for the decomposition in curved layers, as well as for covering each layer with curves. We aim at effectively finding feasible solutions for these two problems, while taking into account manufacturability constraints.

3.1 Decomposition in curved layers

We formulate the problem as follows. Given a solid model $\mathcal{H}$, we seek to decompose it into a sequence of (curved) surface layers $\{\mathcal{S}_i\}_{i=1,\ldots,n}$, such as to represent the material accumulation in AM. This requires satisfying the following conditions:

(1) The solid $\mathcal{H}$ is well approximated by the curved layers as $\mathcal{H} \approx \cup_{i=1,\ldots,n}\Pi(\mathcal{S}_i)$ with $\Pi(\mathcal{S}_i)$ denoting the convolution solid of $\mathcal{S}_i$ by a sphere with diameter r (layer thickness), and there is no overlap between layers – $\Pi(\mathcal{S}_i) \cap \Pi(\mathcal{S}_j) = \emptyset$ $(\forall j \neq i)$;
(2) All surface patches $\{\mathcal{S}_i\}$ are *accessible* – i.e., can be touched by a printer-head while not colliding any $\Pi(\mathcal{S}_k)$ $(\forall k < i)$;
(3) Every curved layers $\mathcal{S}_i$ is enclosed by the dilation of previous curved layers, $\cup_{k=1,\ldots,i-1}\Pi(\mathcal{S}_k)$, with radius r – i.e., the overhang of $\mathcal{S}_i$ is small so that an object under printing is self-supported.

Each $\Pi(\mathcal{S}_i)$ represents a solid layer with thickness r that can be fabricated by moving a printer-head along the surface $\mathcal{S}_i$. We name each surface $\{\mathcal{S}_i\}$ a curved layer, or *working surface*, as the printer head will keep moving along it to accumulate solidified material with a thickness r during fabrication. The use of the symbol '$\approx$' above implies a decomposition minimizing the shape approximation error.

We treat the surface decomposition over the whole model under manufacturability condition as a global search problem. However, searching over all possible sequences of surface layers in a continuous volume space is impractical. To make this amenable to computation, we first discretize space into a regular voxel grid: such

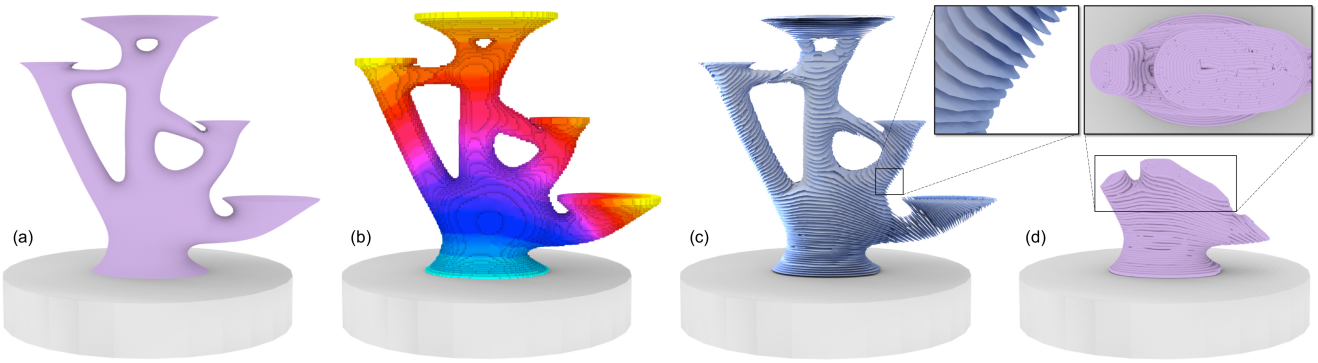


Fig. 2. Illustration of dimensionality reduction for the process planning of multi-axis 3D printing. From left to right: (a) input solid $\mathcal{H}$ for a topology optimized candelabra, (b) accumulation field, (c) curved layers $\{\mathcal{S}_i\}$ extracted from $\mathcal{H}$ and (d) curved tool-paths $\{\mathcal{P}_j\}$ covering each curved layer.

a representation is easy to construct, store and process. We then assume that material accumulation during AM is performed by adding voxels one by one. The criteria for validating the feasibility of a manufacturing sequence are converted into geometric constraints between neighboring voxels. The computed sequence of voxel accumulation indicates the flow of fabrication. The sequence is encoded by storing an integer (rank in sequence) at the center of each voxel. This defines a *growing field* $G(\mathbf{x})$.

The efficient computation of a feasible growing field on the voxel grid of $\mathcal{H}$ is presented in Section 4. We start by introducing a simple greedy scheme using convex-fronts to ensure the accessibility of the working surfaces in Section 4.2. We then introduce the concept of *voxel shadowing* in Section 4.3, which is used to avoid the advancing front from a current layer to a next layer to produce inaccessible regions – i.e., regions that become 'behind' the working front and cannot be accessed any more. In Section 4.4, we introduce a heuristic based on inverse peeling to control the growth. This strongly reduces the chance of generating shadowed regions, resulting in faster computation and less failure cases.

One drawback of the voxel discretization is to cause severe aliasing of the layer geometries. To solve this problem, we compute the working surface for each layer by extracting a corresponding isosurface $\mathcal{S}^*$ from $G(\mathbf{x})$, as a polygonal surface mesh. A working surface $\mathcal{S}$ with accurate boundary is obtained by trimming $\mathcal{S}^*$ with $\mathcal{H}$ so that the boundary $\partial\mathcal{S}$ of $\mathcal{S}$ is exactly located on the boundary $\partial\mathcal{H}$ of $\mathcal{H}$. Details for extracting a working surface with accurate boundary are presented in Section 4.5.

An illustration for this phase of volume-to-surface decomposition in our framework can be found in Fig.2(a)-(c). After obtaining the working surfaces, tool-paths for 5DOF 3D printing are generated on each of them by solving the surface covering problem below. The resultant tool-paths are illustrated in Fig.2(d).

3.2 Surface Covering

Given a curved layer surface $\mathcal{S}$ that is feasible, we next consider how to efficiently generate a set of (curved) tool-paths $\{\mathcal{P}_j\}_{j=1,\ldots,m}$ such that

(1) We cover the layer: $\Pi(\mathcal{S}) \approx \cup_{j=1,\ldots,m}\Pi(\mathcal{P}_j)$ with $\Pi(\mathcal{P}_j)$ denoting the convolution solid of $\mathcal{P}_j$ by a sphere with radius r, and there is no overlap between paths – i.e., $\Pi(\mathcal{P}_i) \cap \Pi(\mathcal{P}_j) = \emptyset$ $(\forall j \neq i)$;
(2) The number of curves, m, and the distance between the ending points of a tool-path, $\mathcal{P}_j$, and the starting point of the next tool-path, $\mathcal{P}_{j+1}$, are minimized. This reduces the artifacts caused by spurious filaments (so-called *stringing*);
(3) The shape of each curve $\mathcal{P}_j$ should be as smooth as possible and be easily realized on a robotic arm.

To meet these conditions while covering the surface, we rely on a variation of Fermat-spiral curves [Zhao et al. 2016], computed on a mesh surface using geodesic distance-fields. The robotic arm introduces additional difficulties, in particular regarding abrupt changes of orientations. Printing orientations and poses are optimized to realize a 5DOF printing tool-path compatible with the robotic arm. Details are presented in Section 5.

4 DECOMPOSITION IN CURVED LAYERS

This section presents our method for decomposing a given solid model $\mathcal{H}$ into a set of working surfaces $\{\mathcal{S}_i\}_{i=1,\ldots,n}$ for 5DOF tool-path generation. The problem discretization is first introduced in Section 4.1. Then, we present a scheme for generating the growing field $G(\mathbf{x})$ following a greedy strategy (Section 4.2), which is later improved by incorporating a mechanism to reduce the apparition of inaccessible regions (Section 4.3). A peeling-based heuristic is introduced to further reduce failure cases and to improve performance (Section 4.4). Lastly, we describe the extraction of working surfaces from $G(\mathbf{x})$ in Section 4.5.

4.1 Problem discretization and approximation

The input solid model $\mathcal{H}$ is represented by a set of voxels $\{v_{i,j,k}\}$ with a fixed width w (i.e., $\mathcal{H} \approx \bar{\mathcal{H}} = \{v_{i,j,k}\}$), where $\mathbf{c}_{i,j,k}$ denotes the center position of $v_{i,j,k}$ in $\mathbb{R}^3$. $\bar{\mathcal{H}}$ is then converted into a growing field, from which working surfaces are extracted. To allow for the trimming that provides accurate boundaries, the input solid needs to be fully bounded by its voxel representation – that is $\mathcal{H} \subset \bar{\mathcal{H}}$.

We now give the definitions and constraints required for computing feasible sequences of 5DOF material accumulation.

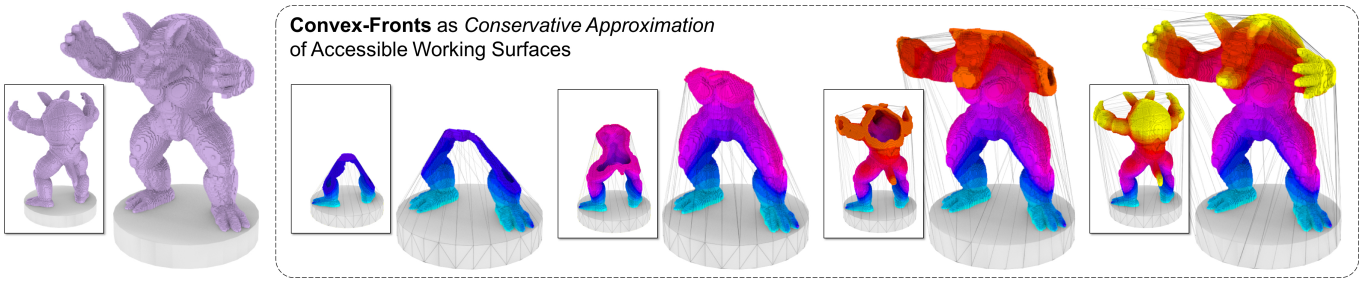


Fig. 3. Advancing convex-front for collision-free 5DOF volume printing. The model is a hollowed Armadillo with 540.6k voxels. Material accumulation is always performed on the *convex-front*: the convex hull of previously deposited voxels and the platform. Back-views are also provided.

Definition 1 Two voxels, $v_{i,j,k}$ and $v_{r,s,t}$, are defined as *AM-stable-neighbors* (ASN) to each other if $\|(i,j,k)-(r,s,t)\|_1 \in \{1,2\}$.

Here $\|\cdot\|_1$ denotes the L^1-norm. Note that only face-neighbors and edge-neighbors are considered as AM-stable neighbors: if two voxels are neighboring by only a vertex, the interface between them is deemed too small for reliable accumulation. The ASN set of a voxel $v_{i,j,k}$ is denoted as $\mathcal{N}(v_{i,j,k})$.

Definition 2 A voxel $v_{i,j,k}$ is defined as ϵ-located on a polyhedron $\mathcal{P}$ if the distance, $d(\mathbf{c}_{i,j,k}, \partial\mathcal{P})$, between $\mathbf{c}_{i,j,k}$ and $\partial\mathcal{P}$ is less than ϵ, where $\partial\mathcal{P}$ denotes the boundary of $\mathcal{P}$.

Definition 3 A voxel $v_{i,j,k}$ is defined as *outside* a polyhedron $\mathcal{P}$ if $\mathbf{c}_{i,j,k}$ is outside $\mathcal{P}$ and $d(\mathbf{c}_{i,j,k}, \partial\mathcal{P}) \geq \epsilon$; similarly, $v_{i,j,k}$ is defined as *inside* when $\mathbf{c}_{i,j,k}$ is inside $\mathcal{P}$ and $d(\mathbf{c}_{i,j,k}, \partial\mathcal{P}) \geq \epsilon$.

Material accumulation can be simulated by adding the voxels of $\bar{\mathcal{H}}$ one by one, first onto the printing platform and then onto previously added voxels. While generating the sequence of voxel-additions, the constraints of 5DOF 3D printing can be applied directly on the growing set of voxels. In particular, two major constraints for manufacturability are considered – *support-free* and *accessibility*.

Constraint 1 (*Support-free*) A voxel can only be accumulated if one of its ASNs has already been solidified (added).

Note that the support-free constraint using ASN allows to accumulate materials along *all* possible directions, reflecting the rotational capabilities of the 6DOF robotic arm. While this constraint results in a stable accumulation, it however provides no guarantee regarding collisions. This is dealt with through the following constraint.

Constraint 2 (*Accessibility*) When adding a new voxel to a set of already fabricated voxels $\mathcal{V}$, the motion of the printer-head should not collide with $\mathcal{V}$.

This constraint is the most challenging to achieve. It depends on multiple factors, including 1) the size and shape of a printer-head, 2) the sequence of material accumulation and 3) the local geometry of the working surface. The first factor depends on the hardware. The second and the third factors are coupled with each other, as different sequences result in different working surfaces during fabrication.

When incrementally accumulating materials voxel by voxel, both the conditions of support-free and accessibility have to be verified at all times. The computation of ASN is made very efficient by the voxel-representation. However, collision-detection for accessibility is extremely time-consuming if it is taken explicitly on all voxels. To obtain an efficient planning algorithm, we propose to always ensure that the accumulation is performed on an accessible surface, which can be navigated by the printer-head without collisions.

The visible surfaces of a model $\mathcal{H}$ are in fact its accessible surface if the tool is infinitely thin, e.g. is a line. In the other extreme case of using a tool with an infinitely large flat head, the accessible surface of $\mathcal{H}$ becomes its convex hull $C(\mathcal{H})$. Considering that commercial extrusion nozzles have large, nearly flat shapes, the convex hull provides a sensible, conservative approximation of the accessible working surface. Specifically, as the materials are usually accumulated on top of a working platform $\mathcal{T}$, we use the convex hull $C(\mathcal{V} \cup \mathcal{T})$ as the *conservative* accessible surface for the set $\mathcal{V}$ of voxels that have been fabricated. In the remainder of the paper, this convex hull serves as a progressively enlarged volume-bound to supervise the collision-free motion planning. We call it the *convex-front* (see Fig.3 for an illustration of the advancing convex front).

4.2 Greedy scheme for convex-front advancing

As mentioned in Section 3, the growing field $G(\cdot)$ is generated by determining an order of voxel accumulation. A voxel v belongs to the l-th layer if $G(\mathbf{c}(v)) = l$ where $G(\mathbf{c}(v))$ is the value of the grid node enclosing $\mathbf{c}(v)$, the center of v. We seek to compute a sequence of feasible layers, $\{\mathcal{L}_l\}_{l=1,\ldots,m}$, where each $\mathcal{L}_l$ consists a set of voxels that meets the support-free and collision-free constraints. Every current layer $\mathcal{L}_c$ is ϵ-located on a convex hull which encloses all prior layers, $\mathcal{L}_l$, with $l < c$. All voxels in $\mathcal{L}_c$ should also be ASN of voxels on the prior layers – i.e., the support-free condition is satisfied for all. Starting from the layer of voxels connected to the platform model $\mathcal{T}$, the greedy scheme generates a sequence of feasible voxel layers. The algorithm opportunistically adds as many voxels as possible into the next layer, following five steps:

(1) For a model $\mathcal{H}$ represented by a set of voxels $\bar{\mathcal{H}} = \{v_{i,j,k}\}$, first assign all the voxels attached to the platform $\mathcal{T}$ to the first layer $\mathcal{L}_1$. Set it as the current working layer $\mathcal{L}_c$.
(2) All voxels of $\mathcal{L}_c$ are added into the set of *processed* voxels $\mathcal{V}$.
(3) The convex hull of $\mathcal{T}$ and all processed voxels in $\mathcal{V}$ is computed as $C_c = C(\mathcal{V} \cup \mathcal{T})$. It is the current convex-front.
(4) For each voxel $v_{i,j,k} \in \mathcal{L}_c$, any of its unprocessed ASN, $v_{r,s,t}$, (by collision-free condition) will be a candidate voxel to be

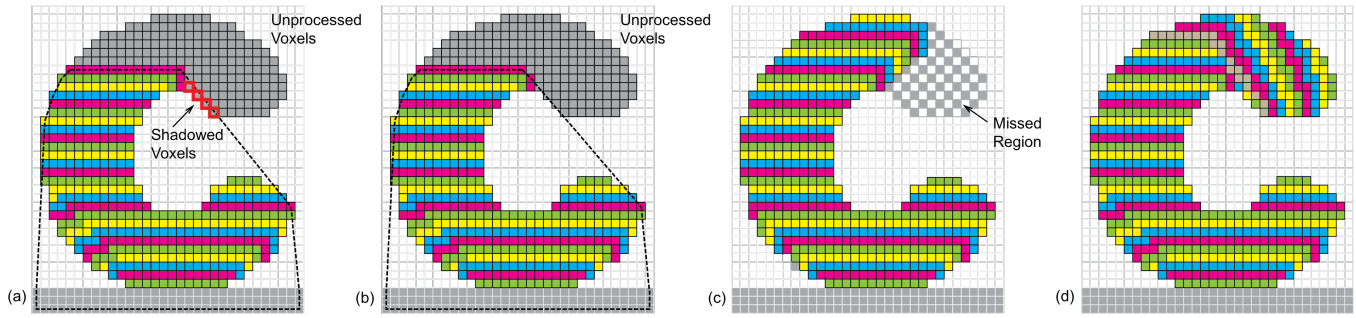


Fig. 4. An example illustrating shadowed voxel avoidance. Colors are used to indicate voxels belonging to different layers. (a) a critical step where shadowed voxels appear, (b) the shadowed voxels could be avoided by not adding a few voxels to the current layer, (c) the final result without shadow prevention – note the large missing regions – and (d) the result when enabling shadowed voxel avoidance.

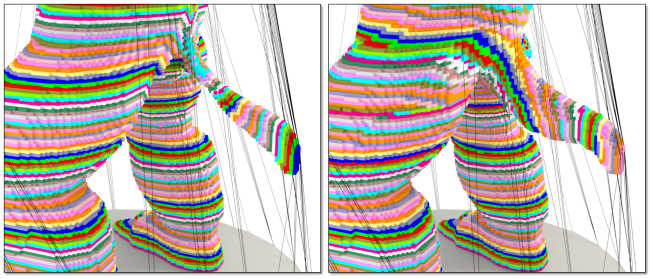


Fig. 5. The base of the Armadillo model's tail cannot be reached by the greedy scheme of convex-front advancing (left – as a result, 10,752 voxels are missed out of 540.6k voxels). This is improved by considering shadowed regions during growth (right – only 3 voxels are missed). Voxels in the same layer are displayed using the same color.

inserted into the next layer, $\mathcal{L}_{next}$, if $v_{r,s,t}$ is not inside C_c (Definition 3).

(5) If $\mathcal{L}_{next} \neq \emptyset$, set $\mathcal{L}_c = \mathcal{L}_{next}$ and go back to Step (2).

It is easy to find that the most time-consuming step is the computation of C_c in step (3). To remove the redundancy of computation, the evaluation including all processed voxels can be replaced by only using the previous convex hull C_{prev} and the newly added voxels in $\mathcal{L}_c$, that is $C_c = C(C_{prev} \cup \mathcal{L}_c \cup \mathcal{T})$. As only the local search and the detection of *in* / *out* convex are included in the computation of this algorithm, it is very efficient – around 32.8 seconds for a model with 540.6k voxels.

This greedy strategy results in curved layers with large areas of connected voxels, that can be later covered with tool-paths. Unfortunately, it also often produces a situation where the convex-front cannot reach all regions of an input model: see the left of Figure 5 where the base of the Armadillo tail (with large overhang) cannot be reached. In the following Section, we present an improved strategy that strongly minimizes such failure cases.

4.3 Preserving accessibility

We modify the greedy scheme to significantly reduce the apparition of unreachable regions. We call *shadowed voxels* these voxels which can no longer be accessed because they are occluded from the fabrication device by previously printed regions.

Definition 4 A voxel, $v_{i,j,k}$ is *shadowed* if it is unprocessed but lies inside the convex hull of the current advancing front, C_c. A set of shadowed voxels form a shadow region, as illustrated in Fig. 4.

The fact that a shadowed voxel can no longer be reached is irreversible – the convex hull is monotonically increasing with respect to the inclusion property during accumulation. Thus our algorithm strives to avoid such cases. This motivates our inclusion of the following constraint to prevent shadowed regions.

Constraint 3 (*Shadow-prevention*) When adding new voxels onto a set of already fabricated voxels $\mathcal{V}$, the number of shadowed voxels should increase as little as possible.

We add a shadow prevention sub-routine before using the next layer of voxels, $\mathcal{L}_{next}$, to update the field values of $G(\cdot)$. The sub-routine selects a reduced set of voxels $\tilde{\mathcal{L}} \subset \mathcal{L}_{next}$ that avoids producing shadowed voxels, as described next.

4.3.1 Incremental Scheme. Our strategy is to incrementally add voxels from $\mathcal{L}_{next}$ into $\tilde{\mathcal{L}}$, checking them one by one for the apparition of shadowed voxels. Input of the algorithm includes the set of processed voxels $\mathcal{V}$, the next layer from the greedy approach $\mathcal{L}_{next}$, and the current set of shadowed voxels $\mathcal{S}_c$. The algorithm consists of six steps starting from an empty $\tilde{\mathcal{L}}$.

(1) Check if any unprocessed voxels are inside
$$C_p = C(C_{prev} \cup \mathcal{T} \cup \mathcal{L}_{next}),$$
and add them into a set $\mathcal{S}_p$ of *potentially* shadowed voxels.

(2) If $\mathcal{S}_p \neq \mathcal{S}_c$, generate the reduced set $\tilde{\mathcal{L}}$ by the following steps; otherwise, exit the sub-routine returning $\mathcal{L}_{next}$ (no additional shadowed voxel will be produced).

(3) Determine a heuristic sequence $\mathcal{Q}$ to add the voxels from $\mathcal{L}_{next}$ in $\tilde{\mathcal{L}}$.

(4) Remove a voxel v from the head of $\mathcal{Q}$, and add v into $\tilde{\mathcal{L}}$ *if* its addition does not increase the set of shadowed voxels when testing with:
$$C_t = C(C_{prev} \cup \mathcal{T} \cup \tilde{\mathcal{L}} \cup v).$$

(5) Repeat the above step until $\mathcal{Q} = \emptyset$.

(6) If $\tilde{\mathcal{L}} \neq \emptyset$, assign $\mathcal{L}_{next} = \tilde{\mathcal{L}}$. Otherwise, keep the original $\mathcal{L}_{next}$ and update $\mathcal{S}_c$ by $\mathcal{S}_p$ to continue advancing the front, sacrificing the new shadowed voxels in $\mathcal{S}_p$ (i.e., they will not be reached in the future).

Different sequences Q result in different 'safe' subsets $\tilde{\mathcal{L}}$. It is desirable to obtain connected large regions – regions that can be easily covered by toolpaths. Therefore, starting from a randomly selected source voxel in $\mathcal{L}_{next}$ we use a Dijkstra's algorithm to generate a sequence Q according to the voxels' distances to the source.

Testing for shadowed voxels in Step (4) potentially requires visiting all remaining unprocessed voxels, which could be extremely slow. However, only the unprocessed voxels within $\mathcal{S}_p$ (determined in Step (1)) can possibly be inside C_t, since $C_t \subset C_p$. Thus, the algorithm only has to test a small portion of the unprocessed voxels.

4.3.2 Recursive shadow-free sets. The algorithm can be further accelerated. The key idea is to recursively split $\mathcal{L}_{next}$ into subsets until finding those that are safe to add (shadow-free), or until reaching individual voxels producing shadowed voxels.

Specifically, given a subset of voxels $\mathcal{L}_{sub}$ producing shadowed voxels, we divide it into subsets $\mathcal{L}^L_{sub}$ and $\mathcal{L}^R_{sub}$ by splitting along the longest principal axis obtained from a *Principal Component Analysis* (PCA) of the voxel centers in $\mathcal{L}_{sub}$. The operation is recursed until reaching a shadow-free subset or a subset with only one voxel. The shadow-free subsets are added into $\tilde{\mathcal{L}}$. Pseudo-code can be found in Appendix B. This algorithm is faster as we test entire sets of voxels in a single 'shadow' check (the convex hull C_t is obtained by adding all voxels from the set). Whenever the check is false, all voxels can be added to $\tilde{\mathcal{L}}$ without further testing.

When running these two algorithms on the Armadillo model shown in Fig.3 with $540k$ voxels, 5,455 minutes is needed for the incremental algorithm while the adaptive refinement algorithm needs only 304 minutes to generate the similar result – i.e., a 17.9× speedup. Similar orders of speedup are observed on other models with smaller number of voxels (e.g., 6.5× on the bunny model in Fig. 1 with $97.5k$ voxels).

4.4 Inverse peeling for guiding the growth

The algorithm presented so far produces good results, but is slowed down by the many shadow prevention checks. In this section we introduce a heuristic that strongly reduces the need for shadow checks, by guiding the growth towards a good solution.

Our heuristic is motivated by considering the process of material accumulation as an inverse process of material removal in subtractive machining. The basic idea is to construct a material removal order by peeling away voxels from a convex-front of remaining voxels. The peeling process starts from the convex hull of the full object. It then peels aways one (curved) sheet of voxels of constant thickness, and it iterates on what remains. This resembles 5-axis CNC as material is removed along all orientations (i.e., the normal of convex-hull). An illustration of the peeling process can be found in Fig.6(a). The order obtained from peeling is stored as an integer rank in each voxel, defining a field $F(\cdot)$. Then, the inverse field of $F(\cdot)$ is defined as:

$$\bar{F}(\mathbf{p}) = 1 + \left(\max_{\forall \mathbf{q} \in \tilde{\mathcal{H}}} (F(\mathbf{q})) - F(\mathbf{p}) \right) \quad (1)$$

ALGORITHM 1: FieldGovernedCFA

Input: The voxel representation of a solid model, $\tilde{\mathcal{H}} = \{v_{i,j,k}\}$, and the governing field, $\bar{F}(\cdot)$.
Output: An indication-field $G(\cdot)$ with value defined on every voxel of $\tilde{\mathcal{H}}$.

```
Adding all voxels adjacent to the platform T to the first layer, L_1, as a set of voxels;
Set L_1 as the current working layer L_c and C_prev = ∅;
Set the layer index τ = 1 and the threshold as f_τ = Δf;
while f_τ ≤ F̄_max do
    while L_c ≠ ∅ do
        Add all voxels of L_c into the already processed set, V;
        Compute the new convex-front by the convex hull of C_prev, L_c and T as C_c = C(C_prev ∪ T ∪ L_c);
        Set L_next = ∅ and τ = τ + 1;
        foreach v_{i,j,k} ∈ L_c do
            foreach v_{r,s,t} ∈ N(v_{i,j,k}) do
                if v_{r,s,t} NOT inside C_c then
                    if v_{r,s,t} ∉ V AND v_{r,s,t} ∉ L_next then
                        if F̄(c(v_{r,s,t})) ≤ f_τ then
                            Add v_{r,s,t} into L_next;
                        end
                    end
                end
            end
        end
        Compute the reduced set of L_next for shadow-region prevention by Algorithm AdaptiveRefinementShadowPrevention;
        foreach v_{r,s,t} ∈ L_next do
            Assign the field-value as G(c(v_{r,s,t})) = τ;
        end
        Set L_c = L_next and C_prev = C_c;
    end
    f_τ = f_τ + Δf;
end
```

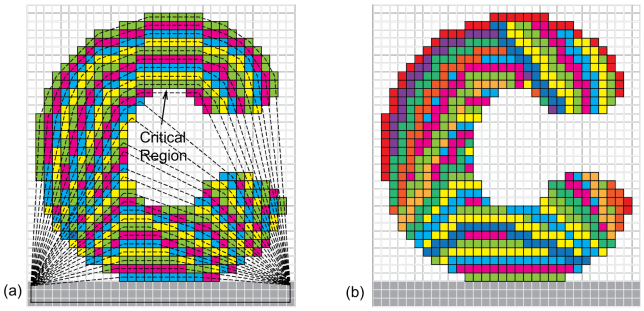


Fig. 6. An illustration of convex-front peeling and the peeling-governed convex-front advancing. (a) Isolated components will be generated when the voxels in the critical region are removed together with other voxels in the same layer. However, such problem on the peeling field $F(\cdot)$ will be automatically avoided when shadow-region preserved convex-front advancing is conducted – see (b) for a result.

This inverse field is used to guide the previous algorithm, still using shadow prevention.

Intuitively, this peeling heuristic helps reduce the apparition of shadowed voxels by encouraging the growth to proceed uniformly and progressively towards the outer object surface, curving the layers ahead of time. Without this heuristic, the algorithm grows roughly flat until a 'downward' feature is suddenly encountered (see the case in Fig.4(a)). The orientation must locally be changed to fill the feature, creating many potential shadowed voxels. On the same case (Fig.6(b)) the layers already curve and align with the surface thanks to the peeling order. Thus, fewer voxels are shadowed when the feature is filled. This strongly reduces the difficulty of the shadow checks; for example, the total computing time for the Armadillo in Fig.3 is reduced from 304 down to 61 minutes.

4.4.1 Field-Governed Growing Algorithm. The field $\bar{F}(\cdot)$ generated by inversely peeling provides a very good guidance for the sequence of material accumulation. We revise the growing strategy to follow the field values of $\bar{F}(\cdot)$, controlling the "speed" of the convex-front growth in different regions. Specifically, we progressively increase a threshold f_τ of the field values in $\bar{F}(\cdot)$ and only advance shadow-prevented convex-fronts into regions where the field values $\bar{F}(\cdot)$ are smaller than f_τ. The algorithm introduces an outer loop above the greedy growing scheme (in Section 4.2) to control the speed of material accumulation w.r.t. $\bar{F}(\cdot)$ as follows:

(1) Initialize the first layer $\mathcal{L}_1$ by the voxels adjacent to the printing platform $\mathcal{T}$ and the threshold $f_\tau = \Delta f$.
(2) Apply the shadow-prevented greedy convex-front advancing in the set of unprocessed voxels $\{v_{i,j,k}\}$ that satisfy $\bar{F}(\mathbf{c}(v_{i,j,k})) \le f_\tau$.
(3) Let $f_\tau = f_\tau + \Delta f$ and go back to Step (2) until $f_\tau > \bar{F}_{\max}$ with $\bar{F}_{\max} = \max_{\forall v \in \bar{\mathcal{H}}}(\bar{F}(\mathbf{c}(v)))$.

Note that while advancing the convex-front all the constraints – support-free, accessibility and shadow-free – should be satisfied. However, by using the inverse peeling field $\bar{F}(\cdot)$ as a guiding heuristic, we observe much better performance from the shadow prevention sub-routine. Pseudo-code of the field-governed convex-front advancing is given in **Algorithm** *FieldGovernedCFA*.

4.5 Curved layer extraction

Given the growing field $G(\cdot)$ with values defined on every voxel of $\bar{\mathcal{H}}$, working surfaces of curved layers are extracted from $G(\cdot)$ as isosurfaces at different isovalues. Assuming that the size of a voxel is w and the diameter of the printer-head's nozzle is d, the working surfaces are extracted at the isovalues $i = 1, \ldots, g_i, \ldots, \lceil \max(G(\cdot))/d \rceil$ with:

$$g_i = (i - \frac{1}{2})\frac{w}{d}. \tag{2}$$

We first construct a narrow-band grid around the isosurface of $G(\mathbf{p}) = g_i$ by using the voxels which field-values fall within the interval $[\lfloor g_i d/w \rfloor, \lceil g_i d/w \rceil]$. We also add their neighboring voxels. Then, a polygonal mesh surface $\mathcal{S}_i^*$ for this isosurface can be extracted by using the *Dual Contouring* (DC) [Ju et al. 2002] or the *Marching Cubes* (MC) [Lorensen and Cline 1987] algorithms. In our implementation, we select DC as it generates less polygons. The Hermite information required by DC is obtained by numerical difference on the scalar-field $G(\cdot)$.

Since the boundary $\partial\mathcal{S}_i^*$ of $\mathcal{S}_i^*$ being extracted from a voxel grid, it imperfectly matches the actual boundary of $\bar{\mathcal{H}}$. A surface $\mathcal{S}_i$ with accurate boundary is obtained by trimming $\mathcal{S}_i^*$ with the input polygonal model $\mathcal{H}$ (using, e.g., [Zhou et al. 2016]). This produces a correct result as long as $\partial\mathcal{S}_i^*$ is always *outside* $\mathcal{H}$, which we ensured by using a conservative sampling when constructing the voxel representation $\bar{\mathcal{H}}$ of $\mathcal{H}$. An illustration is given in Fig.7.

5 TOOL-PATH PLANNING FOR FABRICATION

Once the geometry of the curved layers is obtained, each has to be covered with toolpaths for material deposition. The basic requirements on the *curved* tool-paths $\{\mathcal{P}_j\}_{j=1,\ldots,m}$ covering a surface $\mathcal{S}$ are: path continuity, orientation continuity and pose continuity.

Our system is similar to FDM printers: filament is heated, melted into viscoelastic material and extruded from a printing nozzle through a small hole. This principle makes it difficult to quickly switch extrusion on and off, and therefore a continuous deposition path is demanded. However, position-continuity alone is not sufficient using a multi-axis AM system. We also have to take into account the variations of orientation. Orientation-continuity is crucial for the fabrication process as it determines the smoothness – and hence quality – of material accumulation. Finally, pose-continuity is necessary to avoid poor dynamic behavior in the motion of the 6DOF robotic arm. Three requirements are addressed in three steps, using respectively a Fermat spiral curve for continuous toolpaths, a direction optimization for orientation continuity, and a graph-based optimization for pose continuity. We detail each in the following.

5.1 Position-continuity

Fused materials in FDM are difficult to control due to the compressibility of molten materials. Existing FDM printing software relieves this problem by generating smooth and continuous tool-paths. A

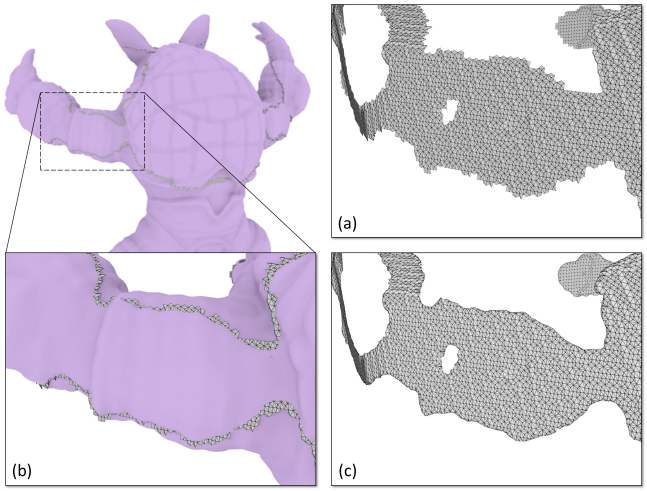


Fig. 7. An illustration for extracting a working surface with accurate boundary: (a) the isosurface $\mathcal{S}_i^*$ generated by dual contouring, (b) $\mathcal{S}_i^*$ is trimmed by $\mathcal{H}$, to obtain (c) the resultant working surface $\mathcal{S}_i$ with its boundary exactly located on the boundary of $\mathcal{H}$.

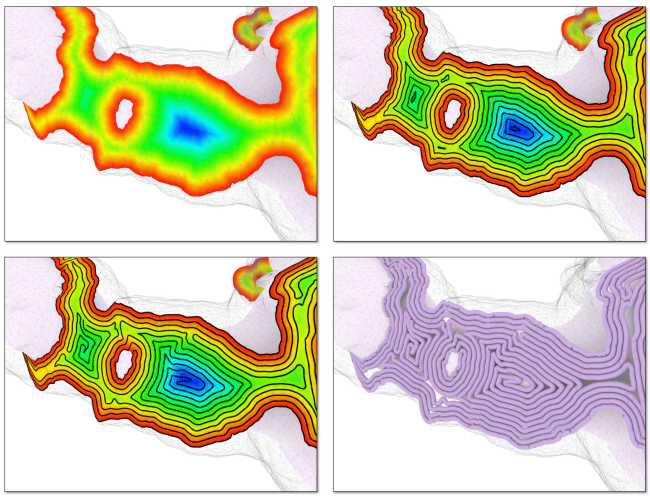


Fig. 8. An example for illustrating the generation of continuous Fermal spiral tool-path: (Top Row) The geodesic distance field is generated by the FWP-MMP method and the iso-contours are extracted on the mesh surface. (Bottom Row) The iso-contours at different iso-values are connected to form the tool-path.

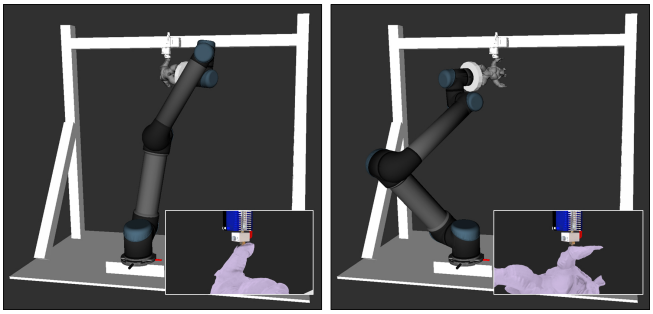


Fig. 9. A same sample point $\mathbf{c}_j$ with a printing orientation $\mathbf{t}_j$ along a tool-path can be realized by the robotic arm using a variety of poses (top row), determined by inverse kinematics. From the closeups, it can be seen that the Armadillo model is actually rotated around the vertical axis of the nozzle between both poses, while the same location is being fabricated.

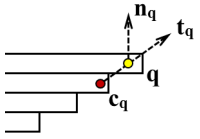
recent effort can be found in [Zhao et al. 2016] to cover a planar domain by a continuous tool-path in Fermat spiral. We adopt the same strategy but extend the algorithm to a curved polygonal surface $\mathcal{S}$. The computation in our algorithm relies on a geodesic metric, which is far more difficult to evaluate than the Euclidean distance used in [Zhao et al. 2016]. Our algorithm is applied to each isolated regions of $\mathcal{S}$ separately and consists of three steps:

- First, we build an exact geodesic boundary distance-field over $\mathcal{S}$ by the *Fast-Wavefront-Propagation* (FWP) based on the *Mitchell-Mount-Papadimitriou* (MMP) method [Xu et al. 2015]. Polyline sources instead of point sources are adopted in order to improve the accuracy of the boundary distance-field.
- Then, we construct iso-contours (i.e., closed-curves having the same iso-value) over the surface mesh $\mathcal{S}$.
- Finally, a 3D Fermat spiral tool-path is generated by connecting iso-contours at different iso-values [Zhao et al. 2016].

An example is shown in Fig. 8 for the working surface generated in Fig. 7.

5.2 Orientation-continuity

The Fermat spiral tool-path generated over the mesh surface $\mathcal{S}_i$ consists of many line-segments having different lengths. Each tool-path is uniformly re-sampled into consecutive points with 1mm distance. Then, the printing orientation at each sample point has to be determined.



The surface normal $\mathbf{n_q}$ at a point $\mathbf{q} \in \mathcal{S}_i$ may seem a natural choice of orientation; however it is not optimized for stable adhesion. We improve the printing orientation as follows. We first find the closest point $\mathbf{c_q}$ of $\mathbf{q}$ among the surfaces of the curved layers printed before $\mathcal{S}_i$:

$$\mathbf{c_q} = \arg\min_{\forall \mathbf{p} \in \mathcal{S}_k} \|\mathbf{p} - \mathbf{q}\|. \quad (\forall k < i)) \tag{3}$$

The vector, $\mathbf{t_q} = \mathbf{c_q q}$, provides a better candidate for orientation, as it is consistent with previous layers. However, orientations separately determined on consecutive samples may have large variations. We thus apply a low-pass filter on those samples with orientation-variation larger than 10°. As a result, orientation continuity can be improved while having better material adhesion.

When the tool-path passes across a crest region, large variation of orientations between two neighboring samples $\mathbf{q}_i$ and $\mathbf{q}_{i+1}$ may also be observed. In such cases we subdivide $\mathbf{q}_i\mathbf{q}_{i+1}$ into smaller line segments and compute the orientations for the newly generated sample points by quaternion interpolation. The subdivision significantly improves the dynamic performance of the robotic arm when printing across the crest.

5.3 Pose-continuity

Our 3D printing system is built around a 6DOF robotic arm. Therefore there is an additional DOF available to the arm when moving along the tool-paths with the target orientation. We exploit this additional DOF to optimize the continuity of poses and therefore the dynamic behavior of the robot motion.

Given a list of points with orientations along a tool-path denoted as $\{(\mathbf{c}_j, \mathbf{t}_j)\}$ $(j = 1, \ldots, m)$, we consider the problem of determining corresponding poses of the 6DOF robotic arm in a joint-angle space. As we are using a hardware setup with a fixed printer head, the position and orientation of material accumulation at a point is defined in the frame $\mathbb{B}$ of the end-effector on the robotic arm with the origin located at the center of the working platform.

We first convert $(\mathbf{c}_j, \mathbf{n}_j)$ into p possible poses of $\mathbb{B}$ in the Euclidean space, by rotating $\mathbb{B}$ around the axis of the nozzle (i.e., the z-axis along which to accumulate materials in FDM). In our current implementation, $p = 30$ is used for the sampling rate. This provides a good trade-off between computation time and quality.

For each pose of $\mathbb{B}$, an analytical inverse kinematics solver is applied to determine all possible configurations in the joint-angle space, denoted as $\{\mathbf{a}_{j,k}\}$. A configuration will be excluded when it

Fig. 10. The results of our algorithm for 3D printing an Armadillo model and a Woman-Pully model by multi-axis motion.

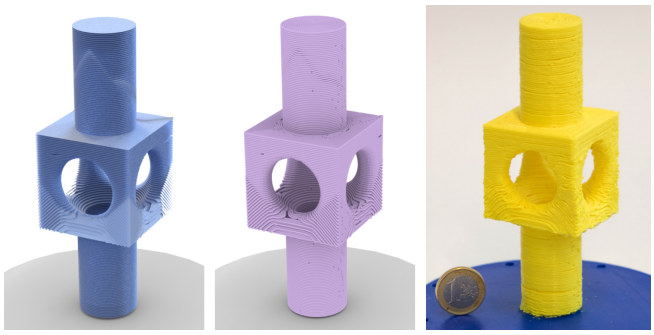


Fig. 11. The result of our algorithm for a mechanical part by multi-axis 3D printing – from left to right, the working surfaces, the tool-paths and the printed model.

leads to self-collision or collision with environmental obstacles. As shown in Fig.9, poses for realizing a sampling point $(\mathbf{c}_j, \mathbf{t}_j)$ of the tool-path can be significantly different from each other. Therefore, an optimization is taken to generate a sequence of continuous poses $\hat{\mathbf{a}}_j$ that minimizes:

$$\sum_j \|\hat{\mathbf{a}}_j \hat{\mathbf{a}}_{j+1}\|_1 \quad (\exists \hat{\mathbf{a}}_j \in \{\mathbf{a}_{j,k}\}), \tag{4}$$

where $\|\cdot\|_1$ denotes a L^1-norm. The problem can be solved on a directed graph by using the Dijkstra's algorithm. More details can be found in supplementary document.

6 RESULTS AND DISCUSSION

We now present results of our approach, implemented in C++. The robotic arm and printing setup is described in supplemental material. We present experimental results in Section 6.1 and discuss limitations and future work in Section 6.2.

6.1 Experimental Results

We tested our approach on a variety of models. The first example is the hollowed Bunny model shown in Fig. 1, discretized in 97.5k voxels. The second and third models are a Candelabra (186.7k voxels) and a hollowed Armadillo (540.7k voxels), shown in Figs. 2 and 10 respectively. We also tested on models with higher-genus topology: the hollowed Woman-Pully model (185.8k voxels) shown in Fig. 10 and the Mech-Part model shown in Fig. 11, which has relatively regular shape but multiple topological handles. Physical printouts are shown in the corresponding figures and Fig. 12. As can be seen, our approach successfully exploits the multi-axis motion of the robotic arm to fabricate regions with large overhang without any additional support structures.

Performance data for processing the models are reported in Table 1. They are obtained on a DELL desktop with an Intel Xeon E5 1630 3.7GHz Quad Core CPU, 32GB RAM, running Ubuntu 14.04.

We now compare the different strategies we discussed for curved layer decomposition, on both the Armadillo model and the Woman-Pully model. We report in Fig. 13 both the computing time and the number of missed voxels. We compare the following strategies:

- *GCFA*: the primary greedy CFA,
- *SP-GCFA*: the shadow-prevented greedy CFA,

Fig. 12. The 3D printing results of all examples shown in this paper.

- *PG-SP-GCFA*: the peeling-governed and shadow-prevented greedy CFA.

As seen from Fig. 13, the PG-SP-GCFA scheme provides the best trade-off between computation speed and quality. Figure 14 shows a failure case of our algorithm on the Fertility model. Neither *SP-GCFA* nor *PG-SP-GCFA* generates a sequence that covers the whole model – i.e., both cannot reach voxels in the chin, although the result of PG-SP-GCFA is obtained much faster and only misses 0.7% of the total voxels. For models like this, a few additional support structures would need to be added for those uncovered regions. See Fig.14(c) for the result by adding support to the missed regions.

Since the curved layers are collision–free (convex front), the tool-paths can be directly computed on them without checking for collisions. The tool-path generation for the examples shown in this paper completes in one to ten minutes. The bottle-neck of our approach is the step checking for shadowed voxels, which can take up to 89.7% of the total time for curved layer decomposition (e.g., the Woman-Pully example). As a result, the computation of curved layer decomposition can take up to hours on some large models. Of course, the time spent on 3D printing a model remains much longer than that of tool-path planning.

Table 1. Computational statistics of our multi-axis volume printing approach

Model	Figure	Total Voxel #	Time (sec.) of Curved Layer Decomposition			Missed Voxel #	Working Surf. Time† (sec.)	Tool-Path Time (sec.)	Fabrication Time (min.)
			Peeling-Field	Shadow Prev.	Total Time				
Bunny	1	97,532	7.24	171.57	203.90	null	888.72	51.09	119.55
Candelabra	2	186,735	9.92	158.83	233.90	null	204.01	131.84	484.81
Armadillo	3, 10	540,689	34.18	3,035.76	3,639.21	null	1,930.95	567.53	760.11
Woman-Pully	10, 13	185,815	10.18	436.07	537.06	null	1,258.14	167.75	419.61
Mech-Part	11	186,723	n/a	1,200.05	1,252.43	null	444.82	126.72	387.72
Fertility	14	77,064	3.79	207.04	232.76	511	-	-	-

†The time reported for working surface extraction includes both mesh polygonization and trimming.

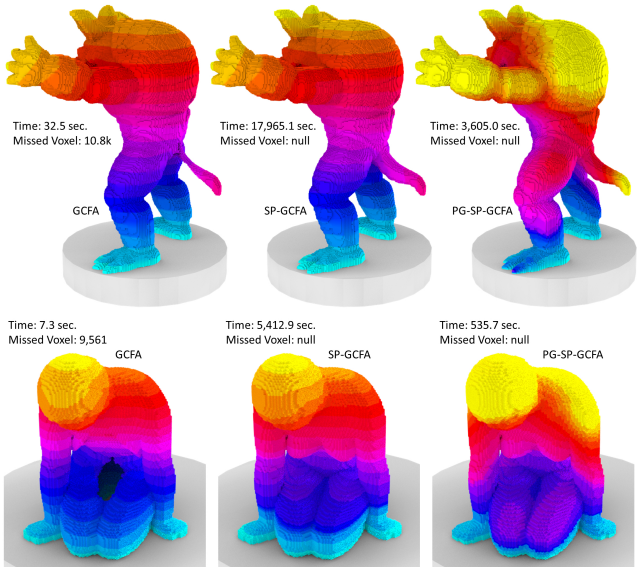


Fig. 13. The results of different strategies for generating the growth field $G(\cdot)$ on an Armadillo model (with 540k voxels) and a Woman-Pully model (with 185k voxels) – note that both models are hollowed. From left to right, the following schemes are tested: (i) the primary greedy CFA (GCFA), (ii) the shadow-prevented greedy CFA (SP-GCFA) and (iii) the peeling-governed and shadow-prevented greedy CFA (PG-SP-GCFA). The time of computation (in sec.) and the number of missed voxels are also reported. Here the time of PG-SP-GCFA includes the step of generating a peeling field. It can be found that with the help of peeling-governed field the computation of shadow-prevented CFA can be much faster (i.e., 4.98× and 10.1× speedup respectively).

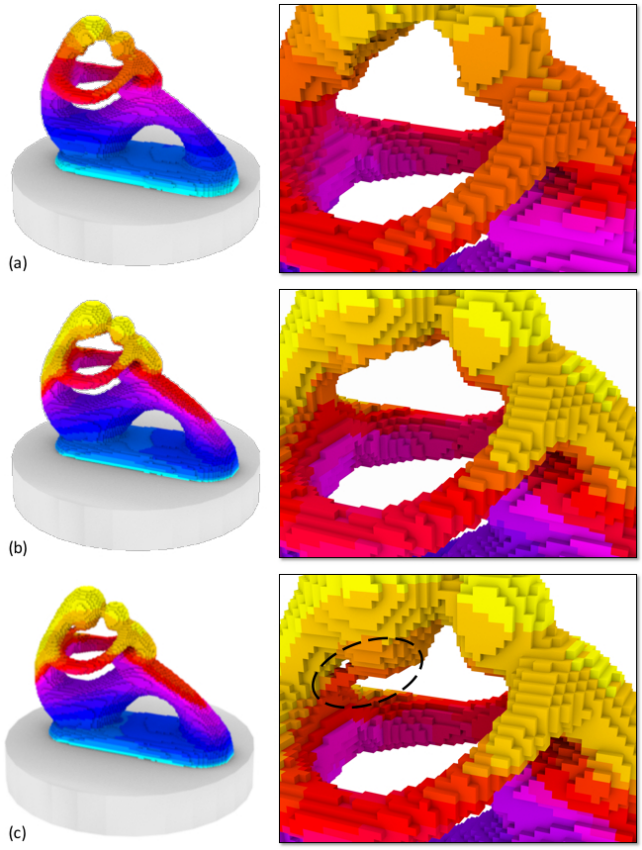


Fig. 14. An example of failure case – a Fertility model with 77, 064 voxels: (a) the result of SP-GCFA scheme (2,093 seconds with 1,376 voxels missed) and (b) the result of PG-SP-GCFA scheme (142 seconds – 14.7× faster but still have 511 voxels missed). After detecting the missed region and adding supports (encircled by dash line), the growing field can be successfully computed to cover the whole model (c) – PG-SP-GCFA scheme is used here (112 seconds).

6.2 Discussion and limitations

Objects fabricated by our system exhibit artifacts. The main reasons are based on hardware position error, non-uniform layer thickness and gaps between tool-paths, which are discussed below.

6.2.1 Discretization error. Our approach processes input solids discretized in voxel grids. The aliasing error along the boundary is avoided when extracting curved layers by trimming the extracted iso-surfaces by the original mesh surface (Section 4.5). However, the space in-between curved layers varies due to the discrete nature of the growing field. We consider how the actual separation distance between layers differs from the ideal, uniform thickness. Figure 15 shows the histogram of distance variations, evaluated by first sampling every curved layer into points and then computing the point-to-surface distances with the PQP library [Gottschalk et al. 1996]. It is found that the variation of distance is relatively small.

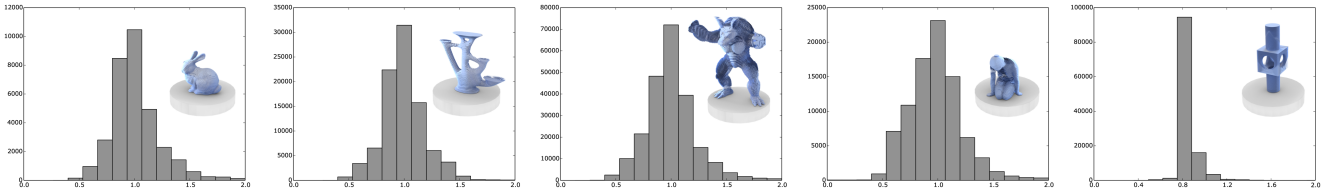


Fig. 15. The histogram of distance variation between working surfaces on example models shown above (all with voxel width 0.8).

In practice, during fabrication the thickness variations are compensated by controlling the feed-rate of material extrusion [Song et al. 2017].

Interestingly, the mean of distance variation is not the same as the width of one voxel. This is due to the fact that voxels neighboring by faces and edges are considered in the same way during the front advancing in $G(\cdot)$ – i.e., the field value is increased by one in both cases. As this is a systematic error caused by voxel discretization, the ratio of layer-distance change w.r.t. the width of voxels is expected to remain constant. Experimentally, a ratio of 1.25× appears in most tests – see Fig.15. Through this experimental calibration, when a printer head with nozzle's diameter d is employed, the computation should be taken on voxels with width $0.8d$. For the Mech-Part example, this ratio can also be used although the resultant layer thickness is closer to $0.8d$ instead of d – i.e., the layer thickness $0.8d$ of 3D printing is realized by controlling the feedrate of material extrusion through a nozzle with diameter d.

6.2.2 Hardware. Our approach successfully handles a variety of models including those with high-genus topology and large overhangs; however, we did not explicitly optimize our approach to prevent the generation of thin-features. As a consequence, the quality of material deposition at those thin-features is not very reliable. This is considered as the major limitation on our current FDM-based hardware platform; although this will not be a problem when applying our method on some other platforms (e.g., to fabricate metal parts by arc welding). Another hardware oriented limitation is that the positioning accuracy of the UR5 robotic arm used in our system is relatively low – only with 0.1mm for the repeatability and with around ±1mm for the positioning error in low speed motion [Kruit 2013], which can be significantly improved when using other high-end systems (e.g., high precision 5-axis table tilting motion system as what is used in 5-axis CNC machining). Figure 16 shows a comparison of the Bunny model fabricated on our setup using different tool-paths. It can be found that the artifacts occurs for both results – i.e., positioning inaccuracy on hardware is a major source for printing error.

Our system relies on a fixed printer-head. As a result, large rotations are applied to the parts under printing, which need to be tightly attached to the end of the robotic arm. Simply using a sticky paper as with the conventional 3D printing method does not work well, as gravity alone can detach the object under fabrication. Our current solution is to first fabricate a working plate using the same material (i.e., PLA in our tests) and fix this plate onto the end of the arm by bolts (see the blue plate in Figs.1 and 12). The objects are then directly printed onto this PLA plate. As the same material is used, the adhesion is strong enough to hold the part. However, the part has to be cut out from the platform after printing. Optionally the platform could be printed in water dissolvable PVA.

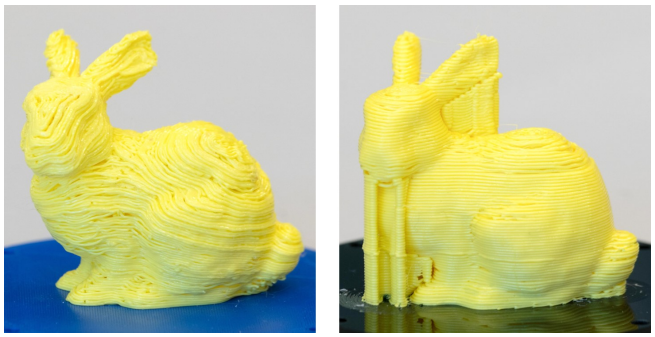


Fig. 16. A comparison for the Bunny model fabricated on the same robotic system by using different tool-paths: (left) the spatial tool-paths generated by our method and (right) the planar paths generated by conventional slicer for 3D printing.

6.2.3 Tool-path and motion planning. As a problem already identified in [Zhao et al. 2016], Fermat spiral tool-paths show small gaps near the medial axis. We leave it as a future work to study other filling patterns that can result in smooth tool-paths for covering a surface patch more completely.

When computing the printing orientations based on the consideration of best material adhesion (Section 5.2), the determined orientation may drive the printer-head locally gouging into the already printed model. In our current implementation, the printing orientations are checked and corrected into gouging-free ones locally. As future research we plan to develop a better scheme to compute continuous and optimized gouging-free printing orientations along an input tool-path. Besides, the current implementation of motion-planning is preliminary – i.e., the dynamic efficiency has not been optimized. More advanced motion-planning methods will be developed in future research.

7 CONCLUSION

Different from conventional 3D printing that deposits materials layer by layer in planes, we propose a new system of 5DOF 3D printing that can fabricate solid models along variational directions. A FDM hardware system with multi-axis motions provided by a robotic arm is employed to fabricate solid models according to the 5DOF 3D printing tool-paths generated by our algorithm. The most challenging part of this system is how to efficiently and effectively

compute the feasible sequence of material deposition so that the fabrication can be performed in a support-free way.

We introduced a methodology to compute advancing fields for material accumulation by always performing material deposition along the surfaces of convex hulls – therefore, the printing process is guaranteed to remain collision-free. Our algorithm first produces a scalar field that represents the growth of the shape during the AM process. After that, we extract from the field a sequence of curved layers, and tool-paths are generated to cover each curved layer by incorporating the constraints from hardware systems. Both computational and physical experiments have been conducted to verify the output of our algorithm.

The results of our experimental tests are very encouraging. Models with large overhangs and high-genus topology can be successfully fabricated by our 5DOF 3D printing system without any supporting structures. We believe that our tool-path generation algorithm will be widely used in 5DOF 3D printing systems to enable a variety of new applications.

ACKNOWLEDGMENTS

C. Dai and C.C.L. Wang are supported by the grant of IDE faculty at TU Delft, and G. Fang is supported by the Chinese Scholarship Council. This work is partially supported by the National Science Foundation of China (Grant No.: 61725204, 61432003, 61521002 and 61628211). S. Lefebvre is supported by ERC grant ShapeForge (StG-2012-307877). The fertility model in Figure 14 was taken from the AIM@SHAPE shape repository.

REFERENCES

Debapriya Chakraborty, B. Aneesh Reddy, and A. Roy Choudhury. 2008. Extruder Path Generation for Curved Layer Fused Deposition Modeling. *Computer-Aided Design* 40, 2 (2008), 235–243. https://doi.org/10.1016/j.cad.2007.10.014

Jérémie Dumas, Jean Hergel, and Sylvain Lefebvre. 2014. Bridging the Gap: Automated Steady Scaffoldings for 3D Printing. *ACM Trans. Graph.* 33, 4, Article 98 (2014), 98:1–98:10 pages. https://doi.org/10.1145/2601097.2601153

Gershon Elber. 1994. Accessibility in 5-Axis Milling Environment. *Computer-Aided Design* 26, 11 (1994), 796–802. https://doi.org/10.1016/0010-4485(94)90093-0

Ben Ezair, Saul Fuhrmann, and Gershon Elber. 2018. Volumetric Covering Print-Paths for Additive Manufacturing of 3D Models. *Computer-Aided Design* 100 (2018), 1–13. https://doi.org/10.1016/j.cad.2018.02.006

Wei Gao, Yunbo Zhang, Diogo C. Nazzetta, Karthik Ramani, and Raymond J. Cipra. 2015a. 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 (UIST '15)*. ACM, New York, NY, USA, 437–446. https://doi.org/10.1145/2807442.2807476

Wei Gao, Yunbo Zhang, Devarajan Ramanujan, Karthik Ramani, Yong Chen, Christopher B. Williams, Charlie C. L. Wang, Yung C. Shin, Song Zhang, and Pablo D. Zavattieri. 2015b. The Status, Challenges, and Future of Additive Manufacturing in Engineering. *Computer-Aided Design* 69 (2015), 65–89. https://doi.org/10.1016/j.cad.2015.04.001

Ian Gibson, David W. Rosen, and Brent Stucker. 2015. *Additive Manufacturing Technologies: 3D Printing, Rapid Prototyping, and Direct Digital Manufacturing*. Springer.

S. Gottschalk, M. C. Lin, and D. Manocha. 1996. OBBTree: A Hierarchical Structure for Rapid Interference Detection. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96)*. ACM, New York, NY, USA, 171–180. https://doi.org/10.1145/237170.237244

Philipp Herholz, Wojciech Matusik, and Marc Alexa. 2015. Approximating Free-form Geometry with Height Fields for Manufacturing. *Comput. Graph. Forum* 34, 2 (May 2015), 239–251. https://doi.org/10.1111/cgf.12556

Kailun Hu, Shuo Jin, and Charlie C. L. Wang. 2015. Support Slimming for Single Material Based Additive Manufacturing. *Computer-Aided Design* 65 (2015), 1–10. https://doi.org/10.1016/j.cad.2015.03.001

Ruizhen Hu, Honghua Li, Hao Zhang, and Daniel Cohen-Or. 2014. Approximate Pyramidal Shape Decomposition. *ACM Trans. Graph.* 33, 6, Article 213 (Nov. 2014), 12 pages. https://doi.org/10.1145/2661229.2661244

Yijiang Huang, Juyong Zhang, Xin Hu, Guoxian Song, Zhongyuan Liu, Lei Yu, and Ligang Liu. 2016. FrameFab: Robotic Fabrication of Frame Shapes. *ACM Trans. Graph.* 35, 6, Article 224 (2016), 224:1–224:11 pages. https://doi.org/10.1145/2980179.2982401

Oleg Ilushin, Gershon Elber, Dan Halperin, Ron Wein, and Myung-Soo Kim. 2005. Precise Global Collision Detection in Multi-Axis NC-Machining. *Computer-Aided Design* 37, 9 (2005), 909–920. https://doi.org/10.1016/j.cad.2004.09.018

Tao Ju, Frank Losasso, Scott Schaefer, and Joe Warren. 2002. Dual Contouring of Hermite Data. *ACM Trans. Graph.* 21, 3 (July 2002), 339–346. https://doi.org/10.1145/566654.566586

Steven Keating and Neri Oxman. 2013. Compound Fabrication: A Multi-Functional Robotic Platform for Digital Design and Fabrication. *Robot. Comput.-Integr. Manuf.* 29, 6 (2013), 439–448. https://doi.org/10.1016/j.rcim.2013.05.001

Yong-Joon Kim, Gershon Elber, Michael Bartoň, and Helmut Pottmann. 2015. Precise Gouging-free Tool Orientations for 5-axis CNC Machining. *Computer-Aided Design* 58 (Jan. 2015), 220–229. https://doi.org/10.1016/j.cad.2014.08.010

C. J. Kruit. 2013. A Novel Additive Manufacturing Approach Using a Multiple Degrees of Freedom Robotic Arm. Master Thesis of TU Delft. (2013).

Ligang Liu, Ariel Shamir, Charlie Wang, and Emily Whiting. 2014. 3D Printing Oriented Design: Geometry and Optimization. In *SIGGRAPH Asia 2014 Courses*. Article 1. https://doi.org/10.1145/2659467.2675050

Min Liu, Yu-Shen Liu, and Karthik Ramani. 2009. Computing Global Visibility Maps for Regions on the Boundaries of Polyhedra Using Minkowski Sums. *Computer-Aided Design* 41, 9 (2009), 668–680. https://doi.org/10.1016/j.cad.2009.03.010

Thomas Llewellyn-Jones, Allen Robert, and Trask Richard. 2016. Curved Layer Fused Filament Fabrication Using Automated Tool-Path Generation. *3D Printing and Additive Manufacturing* 3 (2016), 236–243. https://doi.org/10.1089/3dp.2016.0033

William E. Lorensen and Harvey E. Cline. 1987. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *SIGGRAPH Comput. Graph.* 21, 4 (Aug. 1987), 163–169. https://doi.org/10.1145/37402.37422

Linjie Luo, Ilya Baran, Szymon Rusinkiewicz, and Wojciech Matusik. 2012. Chopper: Partitioning Models into 3D-printable Parts. *ACM Trans. Graph.* 31, 6, Article 129 (2012), 9 pages. https://doi.org/10.1145/2366145.2366148

George Marsh. 2011. Automating Aerospace Composites Production with Fibre Placement. *Reinforced Plastics* 55, 3 (2011), 32–37. https://doi.org/10.1016/S0034-3617(11)70075-3

Stefanie Mueller, Sangha Im, Serafima Gurevich, Alexander Teibrich, Lisa Pfisterer, François Guimbretière, and Patrick Baudisch. 2014. WirePrint: 3D Printed Previews for Fast Prototyping. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*. 273–280. https://doi.org/10.1145/2642918.2647359

Jia Pan, Sachin Chitta, and Dinesh Manocha. 2012. FCL: A General Purpose Library for Collision and Proximity Queries. In *2012 IEEE International Conference on Robotics and Automation*. 3859–3866. https://doi.org/10.1109/ICRA.2012.6225337

Yayue Pan, Chi Zhou, Yong Chen, and Jouni Partanen. 2014. Multitool and Multi-Axis Computer Numerically Controlled Accumulation for Fabricating Conformal Features on Curved Surfaces. *ASME Journal Manuf. Sci. Eng.* 136, 3 (2014). https://doi.org/10.1115/1.4026898

Huaishu Peng, Rundong Wu, Steve Marschner, and François Guimbretière. 2016. On-The-Fly Print: Incremental Printing While Modelling. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 887–896. https://doi.org/10.1145/2858036.2858106

Ariel Shamir, Bernd Bickel, and Wojciech Matusik. 2016. Computational Tools for 3D Printing. In *ACM SIGGRAPH 2016 Courses (SIGGRAPH '16)*. Article 9, 9:1–9:34 pages. https://doi.org/10.1145/2897826.2927367

Hai-Chuan Song, Nicolas Ray, Dmitry Sokolov, and Sylvain Lefebvre. 2017. Anti-Aliasing for Fused Filament Deposition. *Computer-Aided Design* 89 (2017), 25–34. https://doi.org/10.1016/j.cad.2017.04.001

Peng Song, Bailin Deng, Ziqi Wang, Zhichao Dong, Wei Li, Chi-Wing Fu, and Ligang Liu. 2016. CofiFab: Coarse-to-fine Fabrication of Large 3D Objects. *ACM Trans. Graph.* 35, 4, Article 45 (July 2016), 11 pages. https://doi.org/10.1145/2897824.2925876

Xuan Song, Yayue Pan, and Yong Chen. 2015. Development of a Low-Cost Parallel Kinematic Machine for Multidirectional Additive Manufacturing. *ASME Journal Manuf. Sci. Eng.* 137, 2 (2015). https://doi.org/10.1115/1.4028897

Helm Volker, Willmann Jan, Thoma Andreas, Piskorec Luka, Hack Norman, Gramazio Fabio, and Kohler Matthias. 2015. Iridescence Print: Robotically Printed Lightweight Mesh Structures. *3D Printing and Additive Manufacturing* 2, 3 (2015), 117–122. https://doi.org/10.1089/3dp.2015.0018

Nan Wang and Kai Tang. 2007. Automatic Generation of Gouge-Free and Angular-Velocity-Compliant Five-Axis Toolpath. *Computer-Aided Design* 39, 10 (2007), 841 – 852. https://doi.org/10.1016/j.cad.2007.04.003

Chenming Wu, Chengkai Dai, Guoxin Fang, Yong-Jin Liu, and Charlie C. L. Wang. 2017. RoboFDM: A Robotic System for Support-Free Fabrication Using FDM. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*. 1175–1180. https://doi.org/10.1109/ICRA.2017.7989140

Rundong Wu, Huaishu Peng, François Guimbretière, and Steve Marschner. 2016. Printing Arbitrary Meshes with a 5DOF Wireframe Printer. *ACM Trans. Graph.* 35, 4, Article 101 (2016), 101:1–101:9 pages. https://doi.org/10.1145/2897824.2925966

Chunxu Xu, Tuanfeng Y. Wang, Yong-Jin Liu, Ligang Liu, and Ying He. 2015. Fast Wavefront Propagation (FWP) for Computing Exact Geodesic Distances on Meshes. *IEEE Trans. Vis. Comp. Graph.* 21, 7 (2015), 822–834. https://doi.org/10.1109/TVCG.2015.2407404

Xiaoting Zhang, Ka-Chun Chan, Charlie C. L. Wang, Kwok-Chuen Wong, and Shekhar-Madhukar Kumta. 2015. Computing Stable Contact Interface for Customized Surgical Jigs. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 6160–6166. https://doi.org/10.1109/ICRA.2015.7140064

Haisen Zhao, Fanglin Gu, Qi-Xing Huang, Jorge Garcia, Yong Chen, Changhe Tu, Bedrich Benes, Hao Zhang, Daniel Cohen-Or, and Baoquan Chen. 2016. Connected Fermat Spirals for Layered Fabrication. *ACM Trans. Graph.* 35, 4, Article 100 (July 2016), 10 pages. https://doi.org/10.1145/2897824.2925958

Qingnan Zhou, Eitan Grinspun, Denis Zorin, and Alec Jacobson. 2016. Mesh Arrangements for Solid Geometry. *ACM Trans. Graph.* 35, 4, Article 39 (July 2016), 15 pages. https://doi.org/10.1145/2897824.2925901