Near support-free multi-directional 3D printing via global-optimal decomposition

Gao, Yisong; Wu, Lifang; Yan, Dong Ming; Nan, Liangliang

DOI
10.1016/j.gmod.2019.101034

Publication date
2019

Document Version
Final published version

Published in
Graphical Models

Graphical Models 104 (2019) 101034

Contents lists available at ScienceDirect

Graphical Models

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

Near support-free multi-directional 3D printing via global-optimal decomposition

Yisong Gao [a], Lifang Wu [a,*], Dong-Ming Yan [b], Liangliang Nan [c]

[a] Faculty of Information Technology, Beijing University of Technology, Beijing, China
[b] NLPR-LIAMA, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[c] Delft University of Technology, Delft, Netherlands

ARTICLE INFO

Keywords:
3D Printing
Multi-directional
Support-free
Model decomposition
Global optimization

ABSTRACT

In 3D printing, it is critical to use as few as possible supporting materials for efficiency and material saving. Multiple model decomposition methods and multi-DOF (degrees of freedom) 3D printers have been developed to address this issue. However, most systems utilize model decomposition and multi-DOF independently. Only a few existing approaches combine the two, i.e. partitioning the models for multi-DOF printing. In this paper, we present a novel model decomposition method for multi-directional 3D printing, allowing consistent printing with the least cost of supporting materials. Our method is based on a global optimization that minimizes the surface area to be supported for a 3D model. The printing sequence is determined inherently by minimizing a single global objective function. Experiments on various complex 3D models using a five-DOF 3D printer have demonstrated the effectiveness of our approach.

1. Introduction

3D printing, also known as additive manufacturing, has been widely used for both rapid prototyping and small batch production, producing objects ranging from microstructures used in biomedical to large-scale complex parts applied in the aerospace industry. The proliferation of 3D printing techniques is driven by the requirements of flexibility and efficiency. Compared to traditional manufacturing methods such as carving and molding, 3D printing is capable of fabricating objects with arbitrarily sophisticated shapes such as hollow structures.

Conventional 3D printers are limited to 3 degrees of freedom (DOFs), i.e., *X, Y*, and *Z*. The major limitation of such printers is that the material accumulating direction (i.e., fabrication direction) once set cannot be changed in every single printing task. Therefore, it typically requires auxiliary supporting structures to be added and printed to prevent the growing objects from deforming and even collapsing under the effect of gravity. Introducing supporting structures have a few drawbacks. For example, it consumes more materials (and thus less cost-effective), prolongs printing time, restricts printing flexibility, and requires additional post-processing to remove the supporting structures that may harm the surface quality of the printed objects.

To be flexible for objects with increasing complexity, various fabrication systems with more DOFs have been developed. The extra DOFs provide the capability to adjust the direction of material accumulation during the fabrication, which provides more flexibility to the printing system and meanwhile reduces both material and energy consumption. In principle, more DOFs lead to more flexibility but lower accuracy in the printed objects due to the accumulation of the positioning errors of the servo motors. As suggested by Dai et al. [1], a 5-DOF computer numerical control (CNC) system can provide higher printing accuracy than using a 6-DOF robotic arm system. In this work, we exploit a CNC 3D printing system (i.e., translation in *X, Y*, and *Z* directions to drive the print heads plus two additional rotational axes to change the orientation of the platform) designed by ourselves. Instead of exploiting the capability of the extra freedoms (i.e., the two continuous rotational axes), we rely on smart model partition techniques and discrete rotation of the extra freedoms to ensure both flexibility and high printing accuracy. The system works in a "discrete-rotational" style: the model is decomposed into several parts by planes and the parts are printed one by one (the subsequent parts are printed directly on top of the previously printed parts). The printing process of each part is similar to the conventional 3-axis 3D printing while the platform can rotate to the direction suiting the next part when the previous part is finished. Fig. 1 shows the printing process of the Stanford bunny model using our system. With the discrete-rotational printing strategy, highly-accurate traditional CNC subtractive manufacturing movements and existing matured tool-path planning methods can be directly applied to each part of the model. So both the moving velocity of the system and printing precision can reach that of conventional 3D printers and machine

* Corresponding author.
E-mail addresses: ysgao@emails.bjut.edu.cn (Y. Gao), lfwu@bjut.edu.cn (L. Wu), yandongming@gmail.com (D.-M. Yan), liangliang.nan@gmail.com (L. Nan).

https://doi.org/10.1016/j.gmod.2019.101034
Received 24 February 2019; Received in revised form 8 May 2019; Accepted 28 May 2019
Available online 11 June 2019

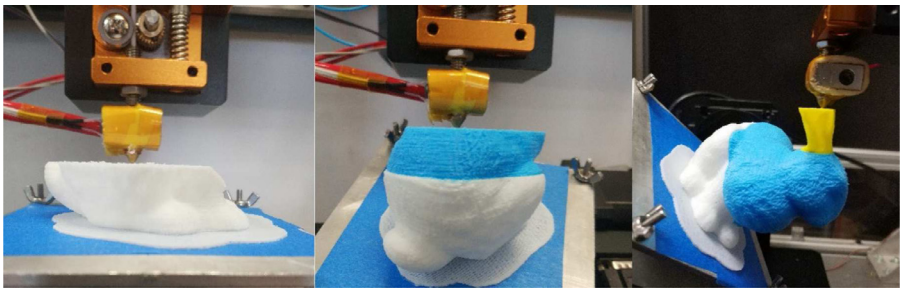


Fig. 1. Our system printing the Stanford bunny model.

tools. Moreover, unidirectional geometry optimization methods such as support-free hollowing [2], and topology optimization methods such as surface remeshing with segmentation [3], can also be applied to each part of the model.

When a multi-DOF printing system is enhanced by model decomposition, achieving collision-free motion becomes more challenging because both printing order and tool-path planning are highly dependent on each other. Existing approaches decompose models into multiple solid parts [4] or line segments [5,6] and print each part in a specific order. This allows each part to be printed in a way similar to the conventional printers with only 3 DOFs. Efforts have also been made to decompose a model into curved surface layers [1], further increasing the flexibility and making supporting structures unnecessary. In our work, we focus on planar-layered fabrication and our goal is to achieve better decomposition results for multi-directional printing.

Given an arbitrary 3D model, our method decomposes it into several parts with an optimal printing order, with which the model parts can be printed one by one and finally builds up the complete object in a single printing pass using the least supporting materials. We comprehensively analyze the collision situations and convert the support-free conditions from a constant direction to varying directions. We propose a heuristic decomposition method using cutting planes to avoid the collision. We minimize the surface area to be supported using global optimization with which the printing sequence can be determined inherently. In addition to the printability with least supports, our method improves surface quality as well by reducing the number of model parts and compensating refinement along the part orientations. We test our algorithm on various complex models to demonstrate its effectiveness and we compare our method to the previous work [4] of the same fabrication style to show its merits.

In short, our work makes the following contributions:

- A set of collision-free and near support-free conditions for multi-directional fabrication;
- A heuristic decomposition method using cutting planes to avoid collision between the extruder and the printed parts;
- A method based on a single-objective global optimization that can simultaneously optimize for both the decomposition and printing order.

2. Related work

2.1. Multi-DOF hardware system

In terms of hardware, multi-DOF printers are being constantly explored starting with an attempt on proof-of-concept printing using a 6-DOF robotic arm [7]. A 5-axis motion system similar to 5-axis CNC machining was proposed to fabricate simple shape on an existing model [8]. The multi-directional additive manufacturing was presented in the work [9] using 6-DOF parallel kinematic Stewart platform. After that, multi-DOF systems with shape decomposition tool-path planning for multi-directional fabrication was built up one after another such as [4–6,10,11], tremendously increasing the flexibility of printing and freedom of design.

2.2. Model decomposition

On the software side, model decomposition and deformation are the two main approaches to improve printability. Model decomposition is favored for its multifunction and preservation of the original shape. Shapira et al. [12] decomposed models according to the shape and volume diameter to achieve coherent results with different gestures. Luo et al. [13] focused on decomposing large models into printing volume with a consideration of printability, number of parts, feasibility, structural soundness, and aesthetics. Hu et al. [14] decomposed models into approximate pyramidal parts, which could be printed without supports. Herholz et al. [15] presented methods using a height field for the printability in a support-free manner. Chen et al. [16], cater to printing efficiency problem, decomposed model into parts and packed them in the printing volume so all the parts could be produced in one pass. Song et al. [11] built coarse internal base structures within the given 3D object and attach thin 3D-printed parts onto the base to recover the fine surface details. Wang et al. [17] improved overall surface quality by decomposing and optimizing the printing directions of each part to avoid the staircase effect that harms surface quality. Wei et al. [18] decomposed models guided by the skeletons for support-free printing especially towards shell models. Chen et al. [19] manufactured the inner of the objects using universal building blocks and fabricated outer shells with pyramidal decomposition which can realize support-free printing.

Most of the methods mentioned above produce objects in a decompose-and-assemble manner, which consumes extra time to assemble the parts together. Wu et al. [4] firstly decomposed solid model into support-free parts which can be fabricated one by one (directly accumulated on the printed parts) in one pass by a 6-DOF robotic system. Models are finally sliced into planar layers along different orientations and printed in a ‘discrete-rotational’ style. Wu et al. [20] improved their previous method [4] by partitioning models using planes so that the method could work with models with complex topology on either 4-DOF or 5-DOF systems. One of the most state-of-the-art methods is proposed to decompose a model into curved surface layers [1], further increasing the flexibility and making support structures unnecessary in most cases. This novel work firstly realized “3D printing” with the full use of DOFs.

3. Methodology

Considering printability, material consumption, quality, and efficiency, achieving optimal model decomposition is challenging. Hard constraints, i.e., collision-free and support-free, are unlikely to be fully satisfied and with which the tool path planning become computationally expensive. In this work, we transform the necessary constraints to a form that simplifies the calculation, and we relax unnecessarily strict constraints for robustness considerations.

We decompose a 3D model using a set of planes (i.e., cutting planes) for planar-layered fabrication. Every time a cutting plane is applied to a model component, the model component is partitioned into two parts. During printing, the part will be printed on top of the cutting plane layer by layer along the normal direction of the cutting plane.

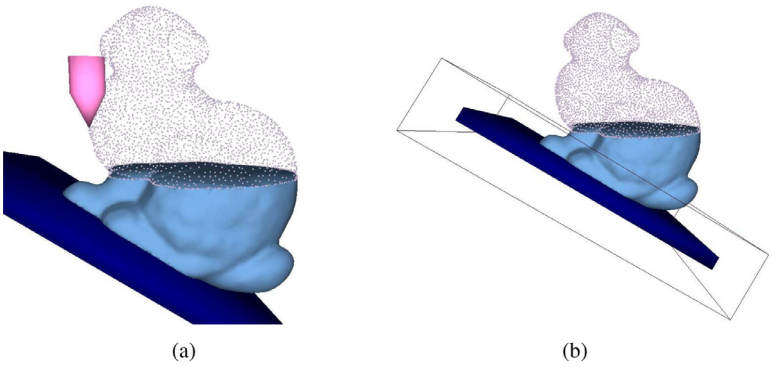


Fig. 2. Extruder-platform collision. (a) Collisions may occur between the extruder (in pink) and the platform (in dark blue) when printing the object parts. (b) The expanded platform box is used for simpler collision detection. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

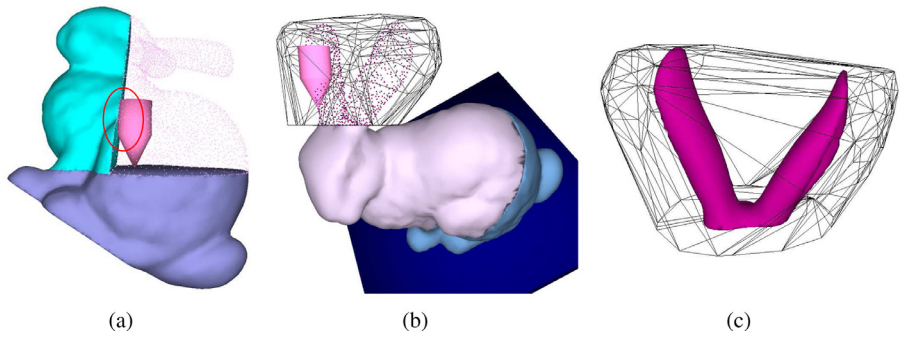


Fig. 3. Extruder-object collision. (a) An example of the extruder-object collision. (b) (c) The expanded convex hull of the model part is used to simulate the workspace of the extruder when printing the part.

3.1. Constraint analysis

Constraint I: Collision-free. To ensure printability, no collision can occur during the printing process. Normally, the following two types of collisions are considered:

Extruder-Platform. In the previous work [4], the workspace of the multi-DOF printer is restricted to the upper half space above the horizontal plane. We observe that the restriction of rotational angular range is unnecessarily strict and may miss collision situations when considering the actual volume of the extruder (i.e., the nozzles that melt and extrude the materials). We seek for an approximate solution that guarantees collision-free between the extruder and the platform. Specifically, we check if collision occurs between each model part and the expanded platform box (i.e., a cuboid obtained by expanding the platform to a distance of the radius of the extruder along all the major axes. See Fig. 2 (b)), allowing the platform to rotate without limiting the angular range (e.g., $[-90°, 90°]$ in [4] while the platform can rotate up to 180° from its horizontal position with the extra rotational DOFs) without collision. In our work, we set the angular range to be $[-135°, 135°]$ (reduced 45° considering the self-supporting angle of the materials).

Extruder-Object. Decomposition of the model using cutting planes typically introduces sharp corners in the printing process (see Fig. 2(a)). For the corner regions, collision usually cannot be avoided even by switching the printing order of the parts. The previous work [4] does not allow intersections between cutting planes to occur within the volume of the object to be printed. We found it too strict and relax it to allow a certain type of cutting plane intersections. Our observation is that collision can be avoided when parts with higher printing priority lie under the base plane of the subsequent parts. With this observation, we propose a heuristic method to decompose a 3D model into parts with planes and meanwhile obtain proper printing priorities for collision-free printing (see Fig. 7 for an example and Section 3.2 for details).

In the general case, extruder-object collisions can be detected by using an expanded convex hull of each part of the model. Similar to the expanded platform box, we calculate the convex hull of each part and expand it by the radius of the extruder (expand every vertex along its normal's component perpendicular to the printing direction. See Fig. 3(b)). Intersection detection is then applied between the expanded hull and the parts with higher printing priority for collision detection.

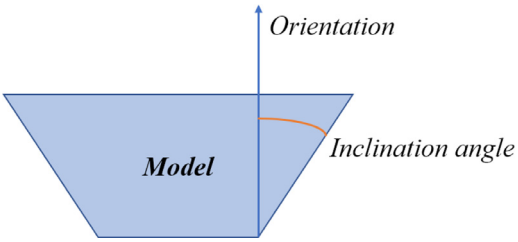


Fig. 4. Inclination angle (illustrated in 2D).

Constraint II: Support-free. Given the fact that there is still no theoretical proof of the existence of a strict support-free decomposition of arbitrary 3D shapes [1], we relax the support-free requirement to a soft constraint. Instead of minimizing the material cost for the supporting structure, we attempt to minimize the area of the regions that require supports to ensure printability. In our problem setting, the supporting analysis is much more sophisticated than conventional model decomposition methods because of the varying printing orientations for the decomposed parts. We consider three situations in multi-directional printing and discuss them in details as follows.

Overlarge inclination angle: We define the angle between the model surface and its printing orientation as inclination angle, as shown in Fig. 4. In FDM-based fabrication, materials can be accumulated without extra supporting structures (i.e., the model can support itself without causing deformation) when the surface inclination angle is smaller than a certain value. This threshold angle is called the maximal self-support angle as used in [1,2,4,14], and this threshold angle is usually set to be 45° or larger depending on the stiffness of the material. In printing, deformation could be observed and the printing object may even collapse under the effect of gravity if the actual inclination angle is greater than this threshold. Normally, the surface regions with an exceeding

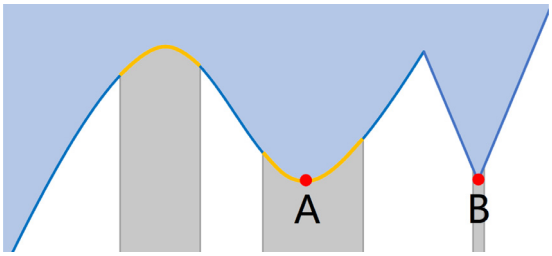


Fig. 5. Two types of local minimum point. For the local minimum point A, supporting structures (shaded regions) are needed for its neighboring faces. For B, support is needed only at point B (with a certain radius) while it is not necessary for its neighboring faces because of the small inclination angles.

inclination angle (called overhanging regions) should be supported using additional structures to ensure printability. In this work, we aim to minimize the area of the overhanging regions. When the area of the overhanging regions can be reduced to zero, this support-free constraint can be satisfied.

Local minimum points: The surface point with the smallest distance to the platform (or the base plane of each part in the multi-directional case), compared with all its neighbors (except those lying on the base plane), is called the local minimum point. Obviously, regions around a minimum point are overhanging and supports are required. To reduce supporting structures, we decompose the model in a way such that the least number of minimum points presents. If the adjacent faces of a local minimum point do not need supports (i.e., the inclination angle is small. See the point B in Fig. 5), we only need a cylinder-like structure that supports the minimum point. In such cases, the supporting surface area is defined as the cross-sectional area of the supporting cylinder.

Fragile regions: In 3D printing with model decomposition, parts are printed in a predefined order and one part is always (except for the first one) printed on top of other previously printed parts. Such a process requires to check if there exist too fragile regions that cannot afford other parts to be printed on top of them. In 3D printing, typical fragile regions are thin fins or bridges on the previously printed parts. Inspired by the work of Luo et al. [13], we detect fragile regions by checking if there exists a region whose distance to the base plane of the following part is smaller than a specified threshold and meanwhile whose normal is sufficiently close to the normal of the base planes (see Fig. 6). Here, we set the distance threshold to 5 mm and the maximal allowed angle deviation is set to 10°. Fragile regions can be detected and handled more efficiently in such an approximate manner, compared to accurate analysis-based approaches such as finite element analysis (FEA, which is fairly time-consuming). In addition, precisely determining whether a region requires supports and how much material it will have consumed are rather complicated. Due to these reasons, we strictly disallow any fragile regions, for which we try simply moving the cutting plane a certain distance along its normal direction in case of fragile regions.

3.2. Model decomposition

Avoiding extruder-object collision. In the previous work of Wu et al. [4], collision is avoided by strictly prohibiting intersections that occur within the interior of the model, which is computationally expensive. Besides, due to such a strategy exposes too much constraint on the location and orientation of a cutting plane, better decompositions are likely to be ignored. We observe that if no printed object exists on top of a cutting plane, there would not be any collision between the extruder and the object because the motion of the extruder is completely on/above the cutting plane. Based on this fact, we utilize a heuristic method to segment a model into collision-free parts. Given the cutting planes, the model is partitioned using a method similar to binary space partitioning (BSP) [21] (we call it 'plane cut'). Specifically, the cutting planes are applied reversely in a top-down segmentation order, which is contrary to the printing priority (see Fig. 7 for an example). This decomposition strategy ensures that the number of parts increases by one for every partition, leading to a simpler printing priority determination. As a result, the model is decomposed into $N+1$ parts without extruder-object collisions by given N cutting planes. However, in another cutting case that only one branch is chosen to decompose the model if the plane intersects the model at several branches (called 'branch cut'), extruder-object collision detection is still indispensable. As we observed, the 'branch cut' is more suitable for tree-like models while the 'plane cut' suits models with ring-like structures better (applying only one 'branch cut' on ring-like structures may lead to unseparated results).

Overhanging detection. In practice, due to the cohesive and elastic forces of the plastic material itself, the object would not collapse immediately when printing the severely sloping regions without supports. Fig. 8 shows such an example. However, deformation should not be neglected as the inclination angle increases for a certain distance. Such a distance (we call it *safe distance*) defines a *safe region* for which supporting structure is not necessary. Experiments show that the safe distance can be up to 10 mm for a 60° inclination angle. Due to the wide distribution of the inclination angles and the cutting plane orientations, it is difficult (and also not necessary) to accurately compute a safe distance for every face in the model. In our work, we set the safe distance to be half of the maximum distance observed in our experiments, i.e., 5 mm. With this safe distance, the detection of overhanging regions is detailed in Algorithm 1 and illustrated in Fig. 9.

Algorithm 1 Detection of overhanging surface regions.

Input:
Model part $part_i$ and its orientation $Orientation_i$;
Output:
The overhanging regions in $part_i$;
1: Compute the inclination angles, relative to $Orientation_i$, for all faces in $part_i$ and mark the potential overhanging faces (ignore regions with small areas);
2: Project the border lines of each overhanging region $area_j$ onto the plane $Plane_i$ (that passes through the mass center of the region and is orthogonal to $Orientation_i$);
3: Calculate the mass center $Center_j$ of the lower boundary of each region and project it onto $Plane_i$;
4: Compute an enlarged offset of the projected closer boundary (the side that is closer to the project region center) with the specified *safe distance* on $Plane_i$. The offset regions are within the safe distance and are support-free, while the remaining regions require supporting structures.
5: Map the 2D regions that require supports back to the 3D model, resulting in the overhanging regions for $part_i$.

4. Implementation

4.1. Objective function

With the aforementioned analysis of the constraints, collision detection and approximation, and overhanging detection, below we detail our objective function. As stated in Section 3.1, we relax the support-free constraints requirement to a soft requirement and we minimize the area of the regions that require supports (instead of minimizing the actual material cost for the supporting structure) for any arbitrary input models. Given N cutting planes, we partition the input model into $N+1$ components by minimizing the following objective function

$$\min \sum_{i=0}^{N} area_i$$
$$\text{s.t. } Orientation_i \cdot Orientation_0 \geq -0.5, i = 1, 2, \ldots, N;$$
$$part_i \cap Box_{platform} = \emptyset, i = 1, 2, \ldots, N;$$

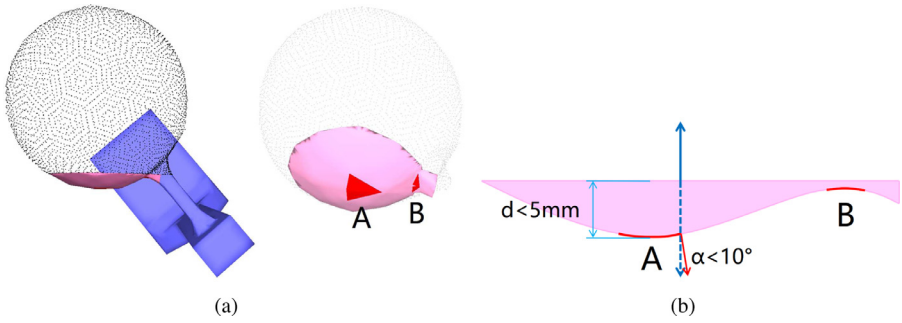


Fig. 6. (a) An example of fragile regions in the hanging-ball model. Region A is a fin and region B is a bridge. (b) Illustration of fragile regions in 2D.

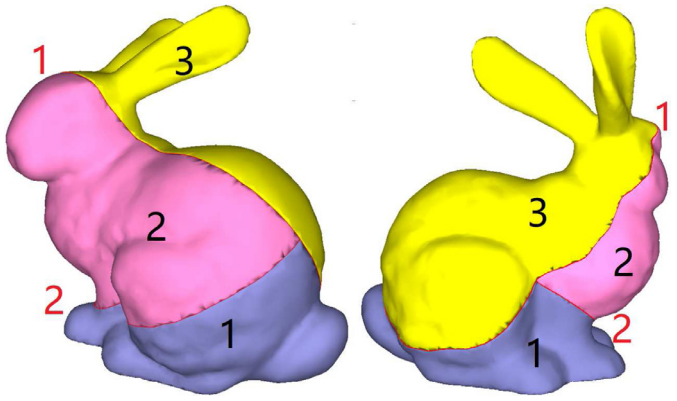


Fig. 7. The decomposition of the Stanford bunny model into three parts using two cutting planes. The red numbers denote the actual cutting order and the black numbers denote the printing priority. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$$\begin{aligned} &ThinRegion_i = \varnothing, i = 0, 1, \ldots, N-1; \\ &Hull_i \cap \bigcup_{j=0}^{i-1} part_j = \varnothing, i = 1, 2, \ldots, N(optional). \end{aligned} \tag{1}$$

where $area_i$ depicts the projected area of the overhanging regions of $part_i$ that require supporting structure. $\sum_{i=0}^{N} area_i$ denotes the total area of the supporting regions. $part_i$ is the decomposed part with the ith plane as its base plane and will be printed along the plane's normal direction $Orientation_i$. $Box_{platform}$ represents the expanded platform box, and $ThinRegion_i$ denotes the fragile regions. $Hull_i$ is the expanded convex hull of $part_i$ and $\bigcup_{j=0}^{i-1} part_j$ represents the printed object parts, which are only needed when applying 'branch cut'. By minimizing Eq. (1), we obtain the parameters for each cutting planes.

4.2. Optimization

We combine genetic algorithm (GA) with elitism, adaptive probabilities of crossover and mutation and simulated annealing for better convergence rate and constraints implementation. The genetic algorithm can be seen as a Markov process and GA with the elitist model (i.e., preserving the best individual in each generation) has been proven to be probabilistic convergent [22], which guarantee the solution is approximately global optimal. Adaptive probabilities of crossover and mutation realize the twin goals of maintaining the diversity of the population and sustaining convergence capacity [23]. Simulated annealing is used to refuse the solutions out of constraints and improve convergence efficiency. See Algorithm 2 for a better understanding of the optimization process.

Parameter setting. The number of cutting planes N is specified by the user taking into account the complexity of the model. The size of the population N_{pop} is set to 200 and the user-specified terminating threshold N_{term} is set to 100.

We use 5 variables to determine the parameters of each cutting plane and the branch that will be cut by this plane, including a point $\mathbf{p} = (a, b, c)$ and two rotational angles α, β. The point $\mathbf{p}$ is used to determine the specific branch to be cut (i.e., if 'branch cut' is allowed, the branch with the shortest distance to the point is chosen) and the two rotational angles determine the orientation of the plane. We concatenate the variables of all cutting planes into a high dimensional vector $input\_set = \{a_1, b_1, c_1, \alpha_1, \beta_1, a_2, b_2, c_2, \alpha_2, \beta_2, \ldots\}$. This vector is then encoded in binary mode for the convenience of genetic operations.

Fitness. We set the negative value of the total projected area of the overhanging regions as fitness (GA always aims to maximize *fitness*). The total projected area includes the projected area of each final overhanging region and the cross-sectional area of the supporting cylinder (i.e., $csarea = \pi R^2$. R represents the radius of the cylinder. Here we use $R = 1$ mm) of each local minimum point. For each iteration in the optimization, planes that do not satisfy the collisions and fragility constraints are rejected by using simulated annealing.

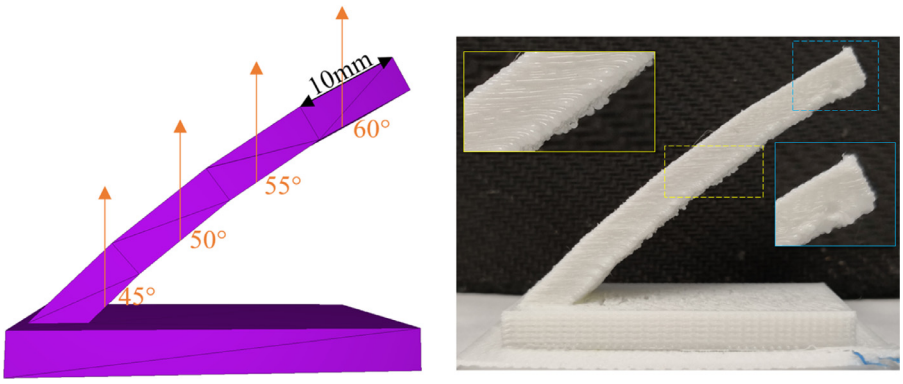
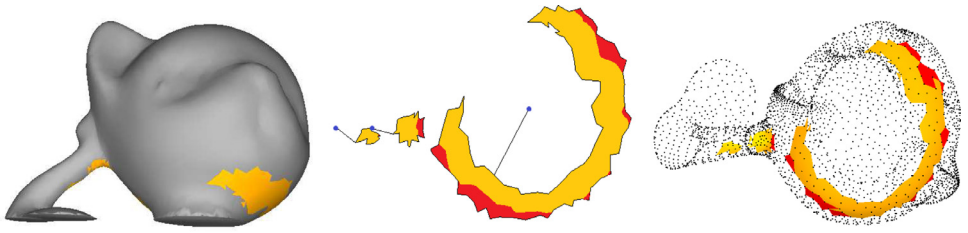


Fig. 8. A printed object with increasing inclination angles.



(a) (b) (c)

Fig. 9. Overhanging detection. (a) Potential faces (marked in orange) with overlarge inclination angles. (b) The projection of the potential regions in 2D. The red regions denote the areas require supporting structures. (c) Mapping the overhanging regions to 3D. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Algorithm 2 Minimizing the total overhanging area $\sum_{i=0}^{N} area_i$.

Input:
The input model M and number of cutting planes N;
Output:
The decomposition of the model $part_i (0 \le i \le N)$ with printing order;

$gen_{\max} \Leftarrow 100$
$gen \Leftarrow 0$
Initialization returns the best *fitness* of the initial population;
while $(gen < gen_{\max})$ and $(fitness_{gen}/fitness_{gen-20} > 0.99)$ and $(fitness_{gen} \neq 0)$ **do**
  $gen++$;
  Preserve the individual with the best *fitness*;
  Selection;
  Crossover;
  Mutation;
  Calculate *fitness* of the new generation;
  Simulated Annealing;
  Replace the worst individual by the preserved one;
end while
Decompose the model using the solution with the best *fitness*. The solution reveals both the cutting planes and the cutting sequence.

Initialization. Given the number of partition planes N (specified by the user), solutions are initialized by randomly sampling the parameter space under collision-free constraints and overhanging test.

Selection. The classical roulette wheel selection is applied, i.e., individuals with higher fitness values have higher priority to be selected. The probability P_s of each individual $Indv_i (i = 1, 2, \ldots, N_{pop})$ is calculated as

$$P_s(Indv_i) = \frac{fitness(Indv_i)}{\sum_{j=1}^{N_{pop}} fitness(Indv_j)}. \tag{2}$$

Every two selected individuals are then paired as parents $p_k (k = 1, 2, \ldots, N_{pop}/2)$.

Crossover: Uniform crossover is used. Each pair of bits in the encoded genes of the two paired individuals is exchanged by the crossover probability P_c. Here P_c is set adaptively for faster convergence:

$$P_c(p_k) = \begin{cases} 1.0 \times \frac{fitness_{max} - fitness(p_k)}{fitness_{max} - fitness_{avg}}, & fitness(p_k) \ge fitness_{avg} \\ 1.0, & fitness(p_k) < fitness_{avg} \end{cases} \tag{3}$$

where $fitness_{\max}$ and $fitness_{avg}$ denote the maximal and the average fitness of the generation and $fitness(p_k)$ is the higher fitness of p_k, meaning that the individuals with higher fitness values are likely to be preserved.

Mutation: Adaptive mutation is utilized to accelerate convergence. The probability of mutation P_m is defined similarly to that of crossover:

$$P_m(Indv_i) = \begin{cases} 0.5 \times \frac{fitness_{max} - fitness(Indv_i)}{fitness_{max} - fitness_{avg}}, & fitness(Indv_i) \ge fitness_{avg} \\ 0.5, & fitness(Indv_i) < fitness_{avg} \\ 0.05, & P_m(Indv_i) < 0.05 \end{cases} \tag{4}$$

Simulated Annealing: The probabilistic acceptance process is similar to that of conventional simulated annealing, which is applied when $fitness(Indv_i) \le fitness_{avg}$. In addition, an extra rejection condition is added to exclude the marked individuals that do not meet the constraints.

Elitism: After simulated annealing, we replace the worst individual with the best one from the previous generation to prevent the *fitness* from decreasing as the number of generations increases.

Termination conditions: If the generation has not exceeded the maximum termination generation N_{term}, the optimization terminates only if the *fitness* reaches 0 (or an acceptable threshold), or the relative change in consecutive 20 generations does not exceed a specified percentage (1% in our implementation), i.e., $|\frac{fitness_i - fitness_{i-20}}{fitness_{i-20}}| \le 0.01$. For the latter case, it requires that the user increases the number of cutting plane N (and repeat the optimization) or introduces supporting structures to ensure printability.

Supporting structure: To reduce the material consumption, we use the tree-like column supporting structures similar to Vanek et al. [24]. A column starts from an overhanging surface region and ends up at other printed parts or the platform. The workspace is partitioned using Binary Space Partitioning (BSP) by the cutting planes. We allow the columns to incline no more than 30° from the cutting plane's normal direction. Column candidates starting and ending up within the same BSP space are given higher priority to be chosen to reduce the complexity of supporting structures.

5. Results and discussions

We have tested our multi-directional printing system on various challenging 3D models. Fig. 1 shows our system in the printing of the Stanford bunny model.

Model decomposition. Figs. 7, 10 (right), and 15 (right) show different decomposition results of the Stanford bunny model. The advantage of our model decomposition strategy is that it provides control over the number of the resulted parts and meanwhile minimizes the material consumption. A negative correlation between N and the total area to be supported can be discovered, i.e., the best total area decreases (or remains constant) when N increases. To determine N, we use a brute-force strategy (iterative increment of N) with a global-optimal method to gain the best results of N planes. The iteration terminates when the area becomes small enough (as a suggestion, no overhanging region but maybe several local minimum points. See the overhanging areas in Table 1, Figs. 12 and 13 for examples) or the N exceeds an expecting number (set by the user). If the decomposition results cannot reach near support-free, an objective function can be used to find a balance between N and the area to be supported

$$\min \quad \alpha \cdot area_{Normalized} + (1 - \alpha)\left(\frac{N}{N_{expect}}\right)^2 \tag{5}$$

where we use a quadratic (or even higher) term to describe N, simulating the booming calculational cost when increasing N. The parameter α can be set according to the expecting number N_{expect}.

Hardware settings and Surface quality. For the hardware parameters, the positioning accuracy of each translational axis is 0.005 mm for

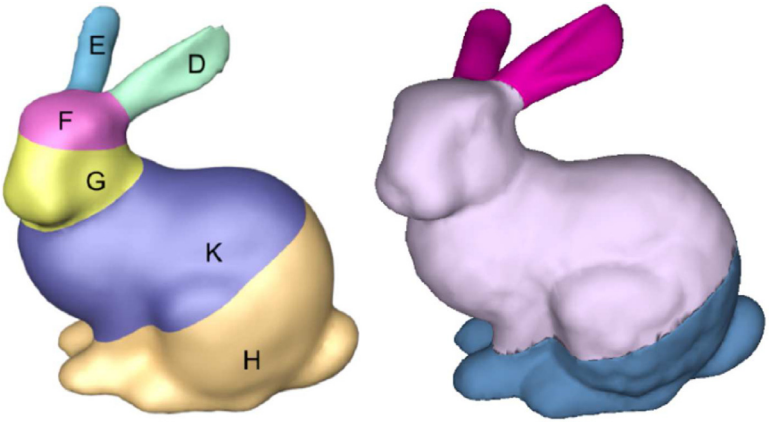
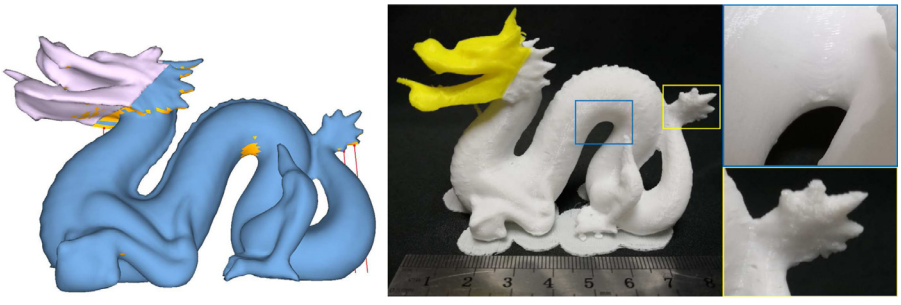


Fig. 10. Decomposition result of the Stanford bunny model. Left: the result of Wu et al. (5 cutting planes and 6 parts). Right: Our result (2 cutting planes and 3 parts).

Fig. 11. Surface quality of the printed bunny object using our printing system. Left: The regions with overlarge inclination angles are marked in yellow. Middle and right: the zoom-ins of the details in the two printed regions. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) (b)

Fig. 12. (a) Decomposition and support-free analysis of the dragon model. The safe regions are marked in orange and three local minimum points are marked in red points and lines (indicating the orientation). (b) The surface quality of the printed object using our printing system. Left: The printed object. One safe region is marked in blue and one region with two local minimum points is marked in yellow. Right: the zoom-ins of the details in the two printed regions. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 1
Statistic of our partition results.

Model	Faces	N	Initialization (in *sec*)	Optimization time(in *sec*)	Overhanging area (mm^2)
Kitten	10000	1	93	463	20.85
		2	87	60	0
Bunny	13026	1	80	240	87.28
		2	81	491	0
H-ball	11540	2	53	355	0
Dragonstand	24598	1	182	780	4π
Armadillo	9998	1	99	442	7π

the repeatability and 0.058 mm per 300 mm for the positioning error, the positioning accuracy of each rotational axis is 0.01° for the repeatability (no exact overall accuracy). The diameter of the nozzle is 0.4 mm, the line width and the layer thickness of material extrusion are 0.2 mm. The highest printing speed is set to 50 mm/s.

To reduce the material consumption, our method does not introduce the supporting structure at the safe regions (with overlarge inclination angles but can be ignored). Experiments show that the printing quality of the safe regions (even with large inclination angles) are satisfactory. As can be seen from Fig. 11 and the regions marked in blue in Fig. 12(b), the surface quality loss in the safe regions can be neglected.

Besides, on the dragon model (see Fig. 12), we only add supporting structure to one local minimum point under the mouth while leaving the points on the tail unsupported. It can be seen that without adding supports to the local minimum points, the deformation becomes serious (the region marked in yellow in Fig. 12(b)), which indicates the necessity of the analysis on the local minimum points.

Compare to the curved-surface-layered printing style of Dai et al. [1], the overall surface quality of the objects generated by the planar-layered manner printing performs better thanks to the flatness and the same thickness of each layer (which contribute to the material accumulation). Our results are decomposed by planes and sliced into planar layers so

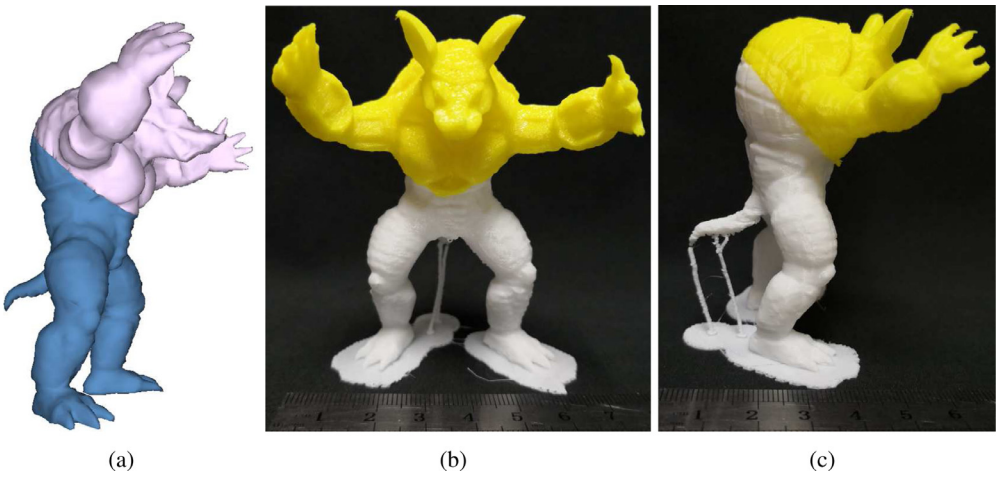


Fig. 13. (a) Decomposition of the armadillo model. (b) (c) The printed object in two views. Only several supporting structures are used to finish the object. The overhanging points and regions cannot be optimized due to the extruder-platform collision restriction.

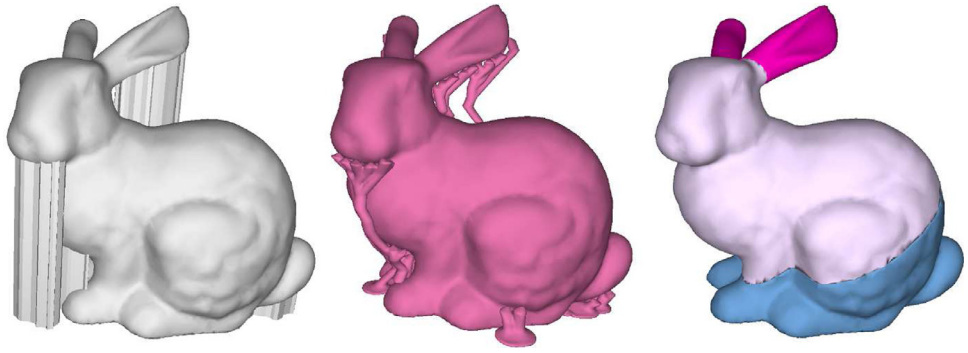


Fig. 14. Different types of supporting structures. Left: cylinder-like supporting structures. Middle: Branch structures [25]. Right: Our method results in support-free printing.

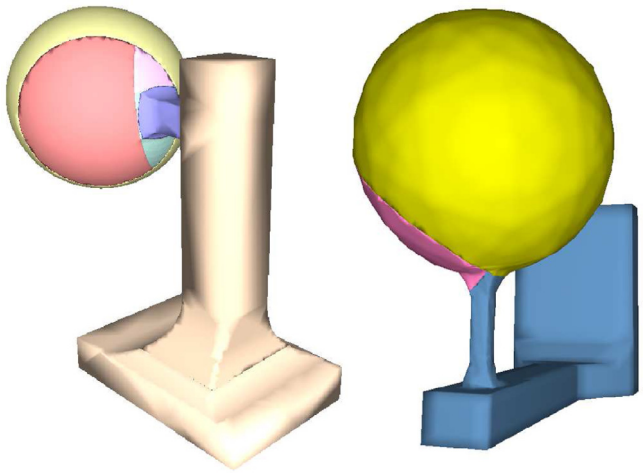


Fig. 15. Decomposition results of the hanging-ball model. Left: the result of Wu et al. (5 planes, 6 parts). Right: out result (2 planes, 3 parts).

that the printing quality could be superior to the curved-surface-layered printing results under the same hardware conditions.

Material saving. The motivation of our work is to achieve near support-free 3D printing. To evaluate this, we measure the volume of the printed objects to record the material cost by assuming that all the models are printed in the solid form. We choose two common support structure generation strategy, one completely filling the overhanging regions and the other one generating branch structures. Fig. 14 shows the bunny model printed with different types of supporting structures. It costs 8.8% extra material with the first strategy (Fig. 14 (left)) and 3.4% for the branch type structures. By minimizing the total supporting area of the overhanging regions, our algorithm can achieve support-free printing for this model, which means the reduction of support material used is 100%).

Timing. Our algorithms were implemented in C++ with OpenMP parallelization enabled on a laptop with Intel Core i7-7770HQ CPU (with 4 cores / 8 threads) and 8 GB RAM. The running times and overhanging areas using different numbers of cutting planes are reported in Table 1. Since the rotational motion of the platform between the printing processes of two parts only takes up several seconds, the printing time is mainly influenced by the total length of the tool-paths, which is increased by adding extra supporting structures. Therefore, the printing time can be saved by reducing the use of supports. In our experiment, while the material cost reduces about 10%, the time consumption can reduce about 20% if the decomposition result reaches support-free (the saving percentage depends on models).

Comparison. We firstly compare our algorithm with the previous work of Wu et al. [4] since the decomposition and printing style of two methods are similar. For the bunny model in Fig. 10 and the hanging-ball model in Fig. 15, both methods successfully decomposed the model. However, our decomposition results have a fewer number of parts and shorter seam length (see Fig. 10). With fewer part, both the cumulative error of the multi-DOF system and the total length of the connecting seams between parts can be reduced. Besides, their coarse decomposition method is based on the shape-diameter analysis. Thus it is only applicable to models that can be abstracted by a skeleton structure and may fail for more general objects, especially for ring-like models. For example in the kitten model shown in Fig. 16, the shape-diameter-based method results in unfeasible decomposition results (see the green part in the left sub-figure) even with a subsequent fine tuning step. In contrast, our optimization-based method is capable of decomposing the model into parts respecting printability.

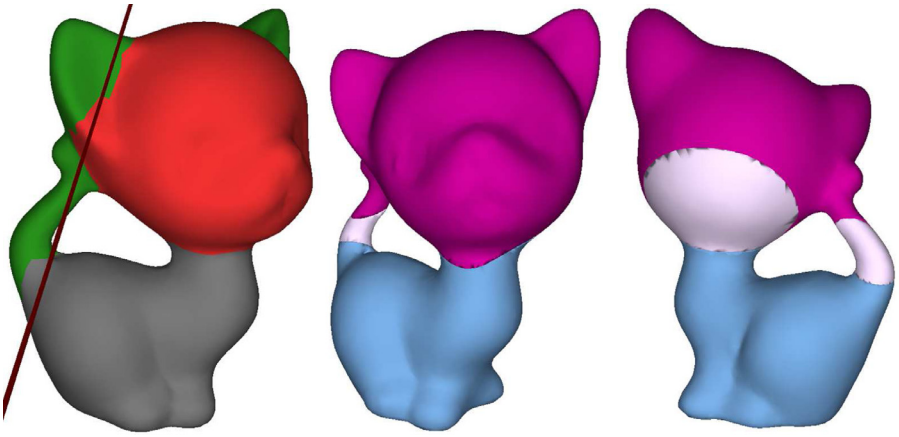


Fig. 16. Decomposition results of the kitten model. Left: the coarse decomposition result of Wu et al. [4] (the green part is hard to modify for printability). Middle and right: our results in two views. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

(a) (b) (c)

Fig. 17. The printed bunny: (a) The result of Wu et al. [4] (5 planes). (b) The result of Dai et al. [1] (curved-layered). (c) Our result (2 planes).

Fig. 17 shows a comparison of the printed bunny model. The number of parts and total seam length of our result are superior to the competing method of Wu et al. [4].

Compare to the method of Dai et al. [1], our scheme is inferior on the motion flexibility but superior on the complexity of implementation. Methods toward conventional 3D printing, such as hollowing, slicing and tool-path planning, can be directly applied on each part of our results. We have realized support-free or near support-free printing on many models (see the results of armadillo model in Fig. 13 and bunny model in Figs. 10 and 17 as comparisons). Our printed objects perform better benefitting from both higher accuracy of the printing system and planar-layered printing style.

6. Conclusions and limitations

We propose a global optimization-based model decomposition method for discrete multi-directional 3D printing, which is capable to achieve near support-free printing. We proposed a heuristic method to partition the model into parts free of collisions. The cutting planes and decomposition order are determined by solving an optimization problem that is formulated to minimize the total surface area that requires supporting structures. The advantage of our model decomposition algorithm is that it always results in a fewer number of parts and therefore reduces the accumulation of mechanical system error as well as the total length of seams between part.

Though our method can achieve near support-free 3D printing, it still has limitations. First, the optimal cutting planes are determined using a genetic algorithm which is computationally inefficient, especially when the number of cutting planes is large. Second, the user has to specify the number of cutting planes, which is typically a trial and error process relying on user experiences. Theoretically, the optimal number of cutting planes can be automatically determined by brute force search (i.e., running the same optimization with different N values). The computation may become unaffordable when N becomes large.

Acknowledgments

This work is partially supported by the Beijing Natural Science Foundation (J170001 and L182059), the National Natural Science Foundation of China (61772523, 61702022, 61802011 and 61620106003), China Postdoctoral Science Foundation (2018T110019), “Ri xin” Training Programme Foundation for the Talents by Beijing University of Technology, and the Construction Project for National Engineering Laboratory for Industrial Big-data Application Technology (312000522303).

References

[1] C. Dai, C.C.L. Wang, C. Wu, S. Lefebvre, G. Fang, Y.-J. Liu, Support-free volume printing by multi-axis motion, ACM Trans. Graph. 37 (4) (2018) 134:1–134:14, doi:10.1145/3197517.3201342.

[2] W. Wang, Y.-J. Liu, J. Wu, S. Tian, C.C.L. Wang, L. Liu, X. Liu, Support-free hollowing, IEEE Trans. Visual. Comput. Graph. 24 (10) (2017) 2787–2798, doi:10.1109/TVCG.2017.2764462.

[3] D. Khan, D.-M. Yan, F. Ding, Y. Zhuang, X. Zhang, Surface remeshing with robust user-guided segmentation, Comput. Vis. Media 4 (2) (2018) 113–122, doi:10.1007/s41095-018-0107-y.

[4] C. Wu, C. Dai, G. Fang, Y.-J. Liu, C.C.L. Wang, RoboFDM: a robotic system for support-free fabrication using FDM, in: 2017 IEEE International Conference on Robotics and Automation, 2017, pp. 1175–1180, doi:10.1109/ICRA.2017.7989140.

[5] R. Wu, H. Peng, F. Guimbretière, S. Marschner, Printing arbitrary meshes with a 5D of wireframe printer, ACM Trans. Graph. 35 (4) (2016) 101:1–101:9, doi:10.1145/2897824.2925966.

[6] Y. Huang, J. Zhang, X. Hu, G. Song, Z. Liu, L. Yu, L. Liu, FrameFAB: robotic fabrication of frame shapes, ACM Trans. Graph. 35 (6) (2016) 224:1–224:11, doi:10.1145/2980179.2982401.

[7] S. Keating, N. Oxman, Compound fabrication: a multi-functional robotic platform for digital design and fabrication, Rob. Comput.-Integr. Manuf. 29 (6) (2013) 439–448, doi:10.1016/j.rcim.2013.05.001.

[8] Y. Pan, C. Zhou, Y. Chen, J. Partanen, Multitool and multi-axis computer numerically controlled accumulation for fabricating conformal features on curved surfaces, J. Manuf. Sci. Eng. 136 (3) (2014) 031007, doi:10.1115/1.4026898.

[9] X. Song, Y. Pan, Y. Chen, Development of a low-cost parallel kinematic machine for multidirectional additive manufacturing, J. Manuf. Sci. Eng. 137 (2) (2015) 021005, doi:10.1115/1.4028897.
[10] W. Gao, Y. Zhang, D.C. Nazzetta, K. Ramani, R.J. Cipra, 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, 2015, pp. 437–446, doi:10.1145/2807442.2807476.
[11] P. Song, B. Deng, Z. Wang, Z. Dong, W. Li, C.-W. Fu, L. Liu, Cofifab: coarse-to-fine fabrication of large 3d objects, ACM Trans. Graph. 35 (4) (2016) 45:1–45:11, doi:10.1145/2897824.2925876.
[12] L. Shapira, A. Shamir, D. Cohen-Or, Consistent mesh partitioning and skeletonisation using the shape diameter function, Vis. Comput. 24 (4) (2008) 249, doi:10.1007/s00371-007-0197-5.
[13] L. Luo, I. Baran, S. Rusinkiewicz, W. Matusik, Chopper: partitioning models into 3d-printable parts, ACM Trans. Graph. 31 (6) (2012) 129:1–129:9, doi:10.1145/2366145.2366148.
[14] R. Hu, H. Li, H. Zhang, D. Cohen-Or, Approximate pyramidal shape decomposition, ACM Trans. Graph. 33 (6) (2014) 213:1–213:12, doi:10.1145/2661229.2661244.
[15] P. Herholz, W. Matusik, M. Alexa, Approximating free-form geometry with height fields for manufacturing, in: Computer Graphics Forum, vol. 34, 2015, pp. 239–251, doi:10.1111/cgf.12556.
[16] X. Chen, H. Zhang, J. Lin, R. Hu, L. Lu, Q. Huang, B. Benes, D. Cohen-Or, B. Chen, Dapper: decompose-and-pack for 3d printing, ACM Trans. Graph. 34 (6) (2015) 213:1–213:12, doi:10.1145/2816795.2818087.
[17] W.M. Wang, C. Zanni, L. Kobbelt, Improved surface quality in 3d printing by optimizing the printing direction, Comput. Graph. Forum 35 (2) (2016) 59–70, doi:10.1111/cgf.12811.
[18] X. Wei, S. Qiu, L. Zhu, R. Feng, Y. Tian, J. Xi, Y. Zheng, Toward support-free 3d printing: a skeletal approach for partitioning models, IEEE Trans. Visual. Comput. Graph. 24 (10) (2018) 2799–2812, doi:10.1109/TVCG.2017.2767047.
[19] X. Chen, H. Li, C.-W. Fu, H. Zhang, B. Chen, D. Cohen-Or, 3D fabrication with universal building blocks and pyramidal shells, ACM Trans. Graph. 37 (6) (2018) 189:1–189:15, doi:10.1145/3272127.3275033.
[20] C. Wu, C. Dai, G. Fang, Y.-J. Liu, C.C.L. Wang, General support-effective decomposition for multi-directional 3d printing, CoRR (2018) arXiv:1812.00606.
[21] H. Fuchs, Z.M. Kedem, B.F. Naylor, On visible surface generation by a priori tree structures, 1980, pp. 124–133, doi:10.1145/965105.807481.
[22] D. Bhandari, C.A. Murthy, S.K. Pal, Genetic algorithm with elitist model and its convergence, Int. J. Pattern Recognit. Artif. Intell. 10 (06) (1996) 731–747, doi:10.1142/S0218001496000438.
[23] M. Srinivas, L.M. Patnaik, Adaptive probabilities of crossover and mutation in genetic algorithms, IEEE Trans. Syst. Man. Cybern. 24 (4) (1994) 656–667, doi:10.1109/21.286385.
[24] J. Vanek, J.A.G. Galicia, B. Benes, Clever support: efficient support structure generation for digital fabrication, Comput. Graph. Forum 33 (5) (2015) 117–125, doi:10.1111/cgf.12437.
[25] R. Schmidt, N. Umetani, Branching support structures for 3d printing, ACM SIGGRAPH 2014 Studio, 2014, doi:10.1145/2619195.2656293. 9:1–9:1