Toward Support-free 3D Printing: A skeletal Approach for Partitioning Models

Xiangzhi Wei Siqi Qiu Lin Zhu Juntong Xi Youyi Zheng†

Abstract—We propose an algorithm for partitioning a 3D model into the least number of parts for 3D printing without using any support structure. Minimizing support structures is crucial in reducing 3D printing material and time, partition-based methods are efficient means in realizing this objective. Although there exists some algorithm for support-free fabrication of solid models, no algorithm ever considers the problem of support-free fabrication for 3D printed shell models. Any partition will inevitably induce seams and cracks on the assembled model, which affects the aesthetics and strength of the finished surface. In this paper, to achieve support-free fabrication while minimizing the effect of the seams, we put forward an optimization system with the minimization of the number of partitioned components and the total length of the cuts, under the constraints of support-free printing angle. Our approach is particularly efficient for shell models, and it can be applicable to solid models as well. We show that the optimization problem is NP-hard and propose a Monte Carlo method to find an optimal solution to the objectives. We applied our partition method on a number of 3D models. Finally, we validate our method by carrying out a serial of 3D printing experiments.

Index Terms—3D printing, skeleton, model partition, support-free.

1 INTRODUCTION

3D printing, or additive manufacturing, has drawn growing interests from researchers in computer graphics. Fused deposition modeling (FDM), stereolithographic (SLA), Selective Laser Melting (SLM) and Selective Laser Sintering (SLS) and the four most popular means of 3D printing techniques. Although 3D printing has seen its applications in producing arbitrarily intricate 3D models, the price of the printing materials, especially for those with high quality, are still outrageously high. Therefore, it is desirable to reduce materials used in the fabrication process. Note that this is also a critical operation for reducing production time and therefore the total production cost. For this purpose, an efficient method is to minimize the support structures, which are removed in the post-processing phase of the fabrication task.

As for minimizing support structures, Autodesk MeshMixer provides a semiautomatic orientation optimization tool to minimize support volume, support area, structural strength, or a combination of these three attributes. However, it requires professional experience in setting the geometric parameters manually. A number of literatures have studied various factors that influence the volume of supports, e.g., optimizing the topology of the support structure [1], [2], determining an optimal fabrication direction [3], [4], [5], partitioning any given model into a set of separate parts that satisfy particular geometric properties such as being pyramidical [6], has minimal packing volume [7] or inter-lockable [8]. However, the partition results of these methods do not simultaneously respect the geometric properties and the support-free printability of the portioned parts. Further, no existing methods ever consider the problem of partitioning a boundary-represented mesh into the least number of parts whose fabrication is free of support structures. The least number of portioned parts corresponds to the least number of seams on the final assembled model, which ensures a nice aesthetics preservation of the model surface; and the support-free fabrication saves material to the most extent, which is particularly helpful in printing objects made of metal powder, resin, or nice plastics, etc.

This paper addresses the problem of decomposing a 3D model into the least number of parts, each of which, when printed in a proper direction, is free of support materials. Unfortunately, this problem is identical to the multi-container packing problem, which has been shown to be NP-hard [9]. Our solution of the problem draws inspiration from the 1D representation of organic models, i.e., skeletons: the topology variations of a natural model can be well-encoded by its skeleton, and a segment of the skeleton corresponding to a chunk of a mesh; further, the mesh chunk is typically a cylinder-like shape which can be printed free of support structures if the printing direction is parallel to the skeletal direction.

We restrict our focus on organic or man-made models since those shapes merit well-defined skeletons. Further, support structures are required for the interior and exterior surface of a mesh model during the 3D printing process. Our approach assumes that the interior of a mesh model is hollowed and the mesh model is shelled with a printer-friendly thickness. Therefore, our objective is to partition a model according to the growth of its skeleton into a set of sub-parts, such that each sub-part is represented by a skeletal subgraph which can be fabricated in a good printing direction without any support structure. In the meanwhile, we also look for a best set of partitioned sub-parts which induce the minimal partition length.

Formally, given a printing direction, if the angle between a facet and the printing direction is less than or equal to θ which is a printer-dependent value, then the facet can be printed without using any support structure. This inspires us to compute a minimum set of subgraph of the skeleton, such that each edge in any subgraph subtends to an axis (the printing direction) by an

† *corresponding author*

- *X. Wei, S. Qiu, L. Zhu and J. Xi are with the Department of Intelligent Manufacturing and Information Engineering, School of Mechanical Engineering,Shanghai Jiao Tong University, China.*
  E-mail: {antonwei,siqiqiu,zhulin0728,jtxi}@sjtu.edu.cn
- *Youyi Zheng is with ShanghaiTech University, China.*
  E-mail: zhengyy@shanghaitech.edu.cn

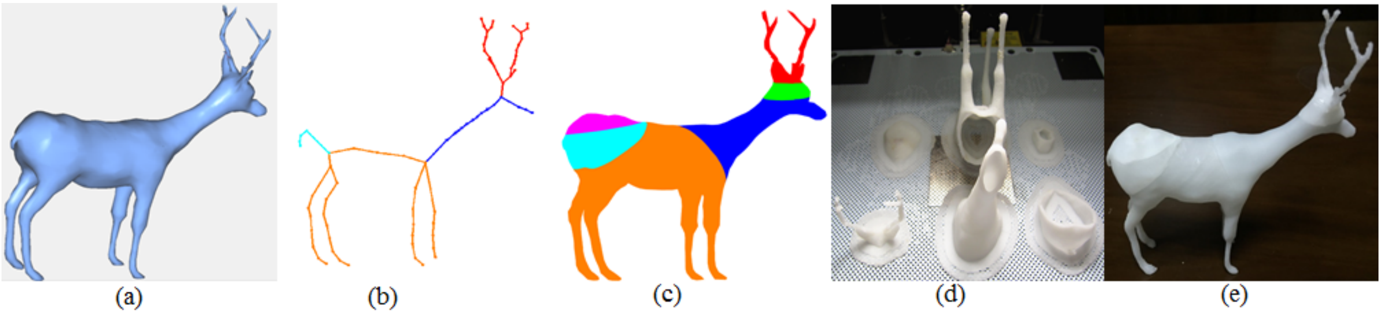


Fig. 1. A deer model is partitioned into support-free parts using our algorithm: (a) a mesh model (hollow); (b) skeleton partition by our approach; (c)mesh partition by our approach;the support-free printing of the parts; (d) the assembly result of the printed parts.

angle of no larger than θ, the corresponding chunk of the mesh is therefore support-free as printed along this axis. In general, a cone of axes satisfies this angle constraint.

Our method handles this issue in a unified Monte Carlo framework which simultaneously looks for the best set of subgraphs which is support-free and with minimal partition length.

In short, our method makes the following contributions:

- We partition any given mesh model into the least number of parts that are printable free of support structures; meanwhile, we preserve the aesthetics of the surface with the least number of seams whose length is minimized.
- We propose a Monte Carlo graph partition method based on the guide of 1D Laplacian skeleton of a given mesh model.

2 Related Work

3D Printing. As cutting-edge 3D printing devices continue to emerge, various 3D printing techniques start to evolve. In computer graphics, a number of literatures have focused on the fabrication of 3D models using 3D printers. Optimization works have been devoted to structural designs with emphasis on saving printing materials while preserving certain strength [10], [11], [12], [13], [14]. The modeling of some particular features have also been studied, for example, deformation behavior [15], animated mechanical characters [16], [17], articulated models with mobile joints [18], [19], models spinnable motions [20], and self-balancing [21].

Model Partition for 3D Printing. A 3D printer cannot directly print a model whose size is larger than the printer's working space. To overcome this practical limitation, Luo et al. [22] proposed a solution to partition a given 3D model into parts for 3D printing and then assemble the parts together. This approach has a few advantages: (i) it is cost-effective in the sense that we only need to print a replacement part for a corresponding broken part; (ii) it is convenient for storage and transportation; (iii) changing some parts of a model allows innovative designs. Along this line of research, Hao et al. [23] partitioned a large complex model into simpler 3D printable parts by using curvature-based partitioning. Hildebrand et al. [4] addressed the directional bias issue in 3D printing by segmenting a 3D model into a few parts each of which is assigned an optimal printing orientation. Vanek et al. [7] reduced the time and material cost of 3D printing by hollowing a 3D model into shells and breaking them into parts, a number of parameters including the total connecting area and volume of each segment are considered during the optimization process. Song et al. [8] recently developed a novel voxelization-based approach to construct inter-locking 3D parts from a given 3D model. Without using any glue, Xin et al. [24] and Song et al. [25] take a 3D interlocking approach to construct and connect printed 3D parts to form an object assembly. Hu et al. [6] proposed an nice algorithm for decomposing a solid model into the least number of pyramids. However, their algorithm is only workable for volumetric models.

Shape Decomposition. In geometry processing, decomposing a shape into parts is a fundamental problem. Many research efforts have been devoted to the problem of model decomposition. An excellent survey can be found in [26]. A large body of research are focusing on partitioning a given 3D model into meaningful parts which agree with human perception [27], [28], [29], [30], [31], [32], [33], among which geometric features captures shape concaveness are mostly exploited in accordance with the minima rules [34], [35]. Other approaches are more application-oriented. For example, texture mapping techniques often require the input shape being decomposed into flat charts which can be flatten with image textures [36], [37]. Our method of partition are based on shape skeletons. While there have been previous work on skeleton based shape decomposition [38], [39], [40], none of them are tailored towards support-free fabrication.

3 Overview

In nature, most organic models ubiquitously contain cylindrical parts [41], e.g., arms, legs, etc. Moreover, an cylindrical part can be well represented by its corresponding 1D skeleton. Based on this property, a chunk of a mesh model can be fabricated free of support if its corresponding skeleton piece subtends to a printing direction by an angle of no large than θ, where θ is a printer-dependent threshold value determined by experiments. Hence, our problem naturally becomes partitioning a given 1D skeleton into sub-skeletons such that each sub-skeleton merits the desirable support-free property. In the remainder of the paper, by model we mean an organic mesh model of natural life form or articulated figures preserving nice topology features of real lives.

Choise of skeleton. Compared to natural skeletons, the medial axis can describe the topology of a mesh model more precisely [42]. However, a medial axis of a 3D mesh model is a 2D surface which cannot be conveniently applied to describe the critical topology changes of the model. Additionally, the medial axis consists of intersecting pieces of planes and conic surfaces, presenting significant complications to algorithms that attempt to construct 3D medial axes. Reeb graph provides a possible choice for 1D skeleton. During the generation process of any reeb graph, the slicing direction and the position of the representative node on each slide (a connected region) seriously influence the choice of critical points and therefore generation of the Reeb graph. However, the determination of suitable slicing direction and representative nodes is an intractable problem. We resort to the 1D Laplacian skeleton proposed by [40] which is extracted by shrinking the mesh model using Laplacian smoothing, such 1D Laplacian skeleton provides an excellent choice for reasonably describing the geometrical and topological variations of any 3D model.

Problem Statement. Our goal is hence to decompose the 1D Laplacian skeleton of the model into the least support-free subgraphs leads to a partition of the model into the least printable parts free of support structures and cracks on the final assembly model. In addition, since support structures result in bumpy supported areas, support-free fabrication also means a nice preservation of the surface quality of the parts. Further, a minimization of the number of cuts and the total cutting length means a minimum amount of seams and their lengths on the assembled model. Therefore, we focus on these two problems in this paper.

Unfortunately, finding such a decomposition is a non-trivial task. Consider a simple example of 1D Laplacian skeleton that is a fork with n vectors sharing a common origin; our objective is partitioning the fork into the least number of sub-forks such that each sub-fork can be packed into a cone of angle 2θ in order to make the sub-fork support-free when fabricated in a given direction. This problem is exactly the problem of packing n items with weights $w_1, ..., w_n$ into bins of capacity c such that all items are packed into the fewest number of bins, which has been shown to be NP-hard [9].

we formulate the partition problem with both the objectives of the total number of cuts and the cutting length, under the constraint of printing angle of each branch with respect to the build platform, the angle between a cut plane and the printing direction, the dimension of each printed model with respect to the printable volume of a given printer, and the base area of a printed model. Since the problem is NP-hard, we propose a randomized Monte Carlo method in compliance with a set of carefully designed selection strategy to seek a practical solution to the optimization problem. Figure 1 shows an example of a printed deer model (shell), our partitioned result based on Laplacian skeleton of the model, and their assembly effect.

4 Algorithm

Skeleton Partition. Let M denote the mesh model, and let S denote the Laplacian skeleton obtained via the algorithm provided in [40]. We propose an algorithm for partitioning S into a minimum set of *disjoint* subgraphs, each of which can be fabricated in a 3D printer without using support structures. Decomposing S into two pieces can be done by duplicating a node v; and partitioning M at node v requires the determination of the position and normal of a cutting plane. To guarantee an aesthetical look on the resulting surface with shortest seams, we need a constraint to minimize the peripheral length of the cut in terms of the position and normal of the cutting plane. Note that the orientation of the cutting plane affects the printing direction and thus the shape of the subgraph while on the contrary, the shape of the subgraph limits the orientation of the cutting plane. Hence, this is an essential *chicken-and-egg* problem. In addition, we also need to consider the printing volume constraint, i.e., the volume of the printing model should be within the working volume of a given 3D printer.

To summarize, our objectives are the minimization of: (1) the number of partitioned components N; (2) the total peripheral length of each cut L_i, i.e., ΣL_i. The constraints of the problem are as follows: (i) Each arc of the partitioned subgraph H_i subtends to an axis by an angle of no larger than θ, where θ is a printer-dependent parameter obtained from experiments. This guarantees that the corresponding mesh component is support-free during the printing process; (ii) Let the normal direction of the cut c (pointing to the exterior of the partitioned part) be denoted as $n(c)$; as the directed arcs of H_i are translated to a common origin, they form a fork, the fork has a central ray, which is the middle axis of the minimum cone that encloses the fork (see Figure 2), let $cone(H_i)$ be the cone, let the central ray of $cone(H_i)$ be denoted as $r(H_i)$; in order to guarantee support-free fabrication for the boundary of the cut, the angle between $n(c)$ and $r(H_i)$ should satisfy: $angle(n(c), r(Hi)) \leq \pi/2 + \theta$; (iii) The base of a printing model should be large enough to gather sticky force from the building platform, such that the model is not deformed during the building process. Formally, let $b(H_i)$ denote the area of the base of the mesh component corresponding to H_i, which is approximately equal to the peripheral length of the base times the thickness of the shell. let τ be a user-defined threshold value, then we have $b(H_i) \geq \tau$. Here τ can be determined empirically; Finally, (iv) Each cut partitions a single subgraph.

We have the following optimization system:

$$\underset{x}{\text{minimize}} \quad N \quad \text{and} \quad \sum_{i=1}^{N} L_i, \quad \text{subject to:}$$

$$A(e, r(H_i)) \leq \theta, \ i = 1, \ldots, m, \forall e \in H_i, \quad (1)$$

$$A(n(c), r(H_i)) \leq \pi/2 + \theta, \quad (2)$$

$$b(H_i) \geq \tau, \quad (3)$$

$$c \cap S = c \cap H_i, \quad (4)$$

where all H_i-s constitute a partition of the original graph to be cut off. A direct exploration of all possible partitions over the graph G could quickly leads to exponential complexity. The key here is to quickly explore potential good partitions in a way that subsequent exploration of the graph is limited to those which leads to a less value of the optimization function. Hence, we employ a randomized Monte Carlo algorithm.

The idea is to separate the minimization of the two target terms (number of subgraphs and the total cutting length) sequentially rather than simultaneously. To minimize the number of subgraphs, we use a randomized exploration strategy, but give a higher probability to explore the graph deeply. In particular, assume that we are given a function $Tirm\_BFS(v, G, \theta)$ which traverses G from v in a breath first search manner to progressively collect arcs which satisfy the constraints. The following algorithm sketches the idea of the skeleton decomposition algorithm. The main idea is to randomly search for candidate subgraphs using Monte Carlo Method, which randomly chooses a node of G to start traversing and randomly grows the subgraph while paying attention to the constraints.

Next we shall show how $Tirm\_BFS(v, G, \theta)$ works to find a locally maximal subgraph starting at v that satisfies the angle constraint. Let H be the current subgraph obtained so far. When an arc e of G is visited, we need to determine whether it should be included into H. If the start of each outgoing arc of H is moved to a common origin, then the arcs form a fork of rays (Figure 2). A naive method to judge whether e should be included is to

Algorithm 1 $SkeletonMeshDecomposition(S, M)$

Input: The Laplacian skeleton S of a mesh model M;
Output: The decomposition of S into a set of the least pieces of subgraphs T, each arc of which subtends to an axis by an angle of no larger than θ;

$T = \emptyset$; $min = \inf$; $count = 0$; max_iter = a user defined large constant;
while $count < max_iter$ **do**
  $G = S$; $U = \emptyset$;
  while $G \neq \emptyset$ **do**
    for $i = 1; i < |S|; i++$ **do**
      $H = Trim\_BFS\_P(v_i, G, \theta)$;
      $H = S/H$;
      $U = U \cup H$;
      if $|U| < min$ **then**
        $T = U$;
        $min = |U|$;
      end if
    end for
  end while
  $count = count + 1$;
end while
return T;

Algorithm 2 Algorithm: $Tirm\_BFS(v, S, \theta)$

Input: A node v of Laplacian skeleton S, an angular value θ.
Output: A maximal subgraph H rooted at v and its corresponding mesh component that meet the constraints $(Eq.1-4)$.

1: starting from v, initialize F($\hat{e_i}$, $\hat{e_j}$); $H = \emptyset$;
2: **while** the current arc e_k of S picked by the BFS process is not empty **do**
3:   **if** $A(\overrightarrow{oc_{i,j}}, \hat{e_k}) \leq A(\overrightarrow{oc_{i,j}}, \hat{e_i})$ **then**
4:     $H = H \cup e_k$;
5:   **else if** $A(\overrightarrow{oq}, \hat{e_k}) \leq \pi - 2\theta$ **then**
6:     $q = O(\hat{e_k}, c_{i,j}) \cap B_{i,j}$;
7:     $\hat{e_i} = \hat{e_k}$;
8:     $\hat{e_j} = \overrightarrow{oq}$;
9:     update $B_{i,j}$ and $c_{i,j}$;
10:     $H = H \cup e_k$;
11:   **end if**
12: **end while**
13: call the cutting scheme for M;
14: $M = M/M_H$;
15: **return** H and M_H;

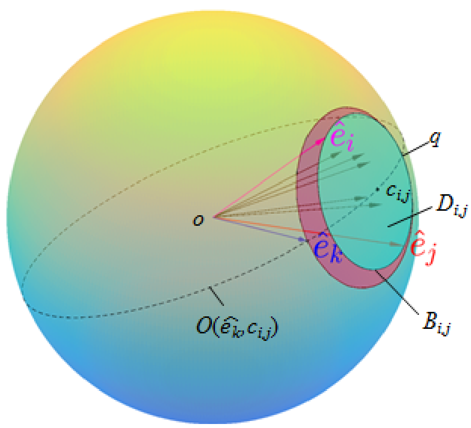


Fig. 2. Illustration of unit vectors, unit sphere, spherical disks, and the determination of taking a new edge in $TrimBFS$.

move the start of e to the origin of the fork, and compute the angle between e and each arc of the fork, e is included if the maximal angle between e and each arc of the fork does not exceed π. However, this would lead to $O(K^2)$ time complexity, where K is the size of S. To speed up this process, we keep the pair of vectors which form the largest angle and judge whether a new vector expands the angle of the fork. See in Figure 2, let $\hat{e_i}$ and $\hat{e_j}$ be the units of these two vectors obtained so far. For simplicity, we denoted by $F(\hat{e_i}, \hat{e_j})$ the fork with the starts of all unit vectors converging at the origin of the coordinate frame, where $\hat{e_i}$ and $\hat{e_j}$ are the pair of unit vectors that form the largest angle in the fork. Let $\hat{e_k}$ be the unit of a new vector to be processed next, if $\hat{e_k}$ penetrates through the blue circle, then no change need to be made to the fork; otherwise, let $D_{i,j}$ denote the spherical disk that passes through the endpoints of $\hat{e_i}$ and $\hat{e_j}$ whose central axis is collinear with $\hat{e_i} + \hat{e_j}$, let $c_{i,j}$ be the center of $D_{i,j}$, let $B_{i,j}$ be the boundary circle of $D_{i,j}$. The circle passing through $\hat{e_k}$ and $c_{i,j}$, denoted as $O(\hat{e_k}, c_{i,j})$, intersects $B_{i,j}$ at two points, let q the point further away from the endpoint of $\hat{e_k}$, then $\overrightarrow{oq}$ and $\hat{e_k}$ are the two extreme vectors that to be used in the next iteration. To summarize, a new arc e_k is taken by Function $Tirm\_BFS$ if and only if one of the following two conditions is met: (1) the angle between $\overrightarrow{oc_{i,j}}$ and $\hat{e_k}$, denoted as $A(\overrightarrow{oc_{i,j}}, \hat{e_k})$, satisfies $A(\overrightarrow{oc_{i,j}}, \hat{e_k}) \leq A(\overrightarrow{oc_{i,j}}, \hat{e_i})$; (2) $A(\overrightarrow{oc_{i,j}}, \hat{e_k}) \leq \pi - 2\theta$, where $q = O(\hat{e_k}, c_{i,j}) \cap B_{i,j}$.

Algorithm 2 demonstrates the growing process. As the order of the chosen arcs influences the shape of the final BFS subgraph, in line 2, the BFS process randomly chooses an arc incident to v to proceed on. In order to guarantee a greater chance of converging to the optimal result in a short time, we apply a training-and-learning procedure for the first k (say 1000) runs. Formally, let N_v be the number of times an arc is chosen as the exit arc when node v is visited. Given the data of the first k runs, as a node v is visited, the probability of choosing an arc e as an outgoing arc in the subsequent runs is, $P(v, e) = N_v/k$. To further speed up the process of $Trim\_BFS$, we assign a mark that stores the minimum number of subgraphs obtained so far, such that the current branching can be terminated if its output number of subgraphs is larger than the mark. Further, as the grow of $Trim\_BFS$ is greedy thus the growth is towards a local maximal subgraph that satisfies the angle constraints, the resulting partition of Algorithm 1 might not be a global optimum. In order to ensure more possibilities to find a global optimum, we employ a probability-based exploring scheme. In particular, we set the probability of stop the growing of $Trim\_BFS$ as k/K where k is the size of the current subgraph and K is the size of S. This gives high probability for the $Trim\_BFS$ to grow in larger sizes while also provides the probability of exploring subgraphs of smaller sizes. We denote $Trim\_BFS\_P$ as the probability based probing. We implement it using rejection sampling.

Mesh Partition. The skeleton partition tells us a rough sketch of the mesh partition, i.e., the cutting plane should be in the vicinity of each node v incident to two distinct subgraphs. Yet we need to determine the exact positions and orientations of the cutting planes. For each node v that is incident to at least two distinct subgraphs H_i and H_j, we process it using the following cutting principle.

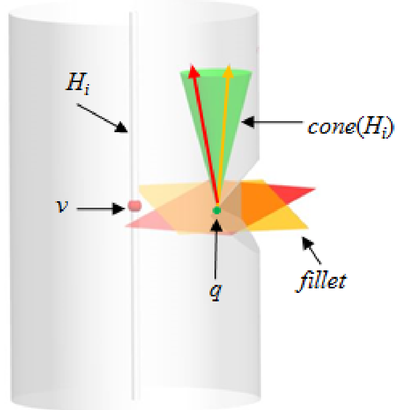


Fig. 3. Illustration of a cone and its corresponding fillets, where two cutting planes in $F(H_i)$ and their associated vectors in $cone(H_i)$ are marked by distinct colors.

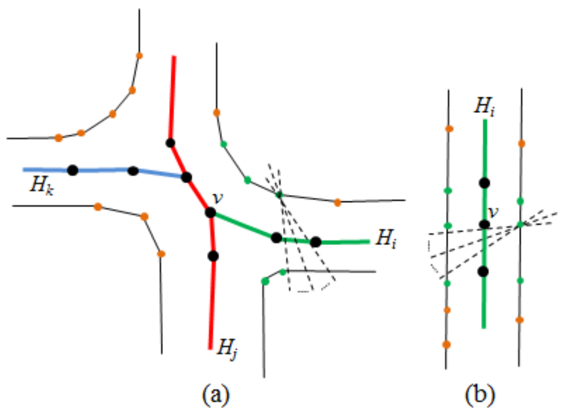


Fig. 4. 2D illustration of R(v) and the cuts, all vertices in R(v) are shown in green, and the cuts associated with a vertex are shown in dashed lines:(a) the case of cuts around concave points; (b) the case of cuts on a cylindrical part.

In Figure 3, at the position of node v, we want to find a surface vertex $q \in M$ around v which the cutting plane should pass through, yet the cutting length is minimized. Let us denote the set of all planes which pass through a surface vertex p and are orthogonal to vectors in $cone(H_i)$ (originated from p) as $F(H_i)$, which we term as a *fillet*. Then if the node v is incident to two subgraphs H_i and H_j, we have two plane sets $F(H_i)$ and $F(H_j)$ at point p, respectively. Depending on the relative position of p, we have the following two cases: *case (i)*: $F(H_i) \cap F(H_j) \neq \emptyset$. In this case, we shall randomly sample a set of cutting planes from $F(H_i) \cap F(H_j)$, and determine the one achieving the minimum cutting length; *case (ii)*: $F(H_i) \cap F(H_j) = \emptyset$. In this case, two cuts are required in order to separate the mesh into support-free subparts. However, care must be taken as the angle between c_1 and c_2 should be constrained by $A(n(c_1), n(c_2)) \leq \pi/2 + \theta$. If this constraint is violated, one more cut in between c_1 and c_2 is required; while the base of each partitioned component should be constrained by $b(H_i) \geq \tau$ and $b(H_j) \geq \tau$. If any of the constraints is not satisfied, we shall translate the fillets along the opposite printing direction until the constraints are satisfied.

In either case, a cut that nicely follows the geometry features is demanded to preserve aesthetic appearance in the final assembled object. However, the positions of the skeleton nodes may not locally reflect the geometric features such as concave areas on the mesh surface, therefore they may be insufficient for a nice partition of the mesh. To compensate this, we exploit the concave vertices of the mesh that are incident to the skeleton nodes and take those that significantly concave into a candidate set of pivots for the cuts. More precisely, let $R(v)$ be the set of concave vertices on M that are incident to v during the Laplacian shrinking process [40], we truncate $R(v)$ such that the non-significant concave vertices are removed away. Here, given a vertex v_i and any of its neighbor v_j, v_i is concave if $(v_i - v_j)(n_j - n_i)$ is nonnegative, the significance of a concave vertex can be quantified as the magnitude of $(v_i - v_j)(n_j - n_i)$ [43], denoted as τ_i. We collect vertices whose τ_i is greater than a threshold δ. See Figure 3(c-d), for each vertex in $R(v)$, we shall process a pair of fillets as done for vertex p above.

Next, we proceed to find a cutting plane around v. We first extend the region $R(v)$ by merging all $R(u)$ where u is an adjacent node of v in S. Given all concave vertices in the new $R(v)$, each concave vertex defines a set of feasible cutting planes in accordance to its $fillet$. By feasible we require that the cutting plane does not cut through other subgraph except for H_i. We then exhaustively go through all feasible cutting planes and find the one whose cutting length is minimized. In case of cylindrical parts which do not merit good concaveness, we reduce the threshold value δ by half and repeat the process until a feasible minimal length cutting plane is found. Figure 4 is an illustration of the above process.

In order to detect whether a cutting plane cut through any subgraph other than H_i, we take advantage of the correspondence between the mesh vertices and the skeleton nodes. Since each vertex of M is mapped to a single node of S [40], as a cut goes through the mesh surface, the endpoints of the cut edges of M give us the information of the cut through. Approximately, if a cut c goes into an edge whose endpoints are incident to a subgraph H_j, then c cuts through H_j.

Sometimes, it is possible for a mesh component which is not printable free of support even though its corresponding skeleton is detected to be printable free of support. This is because the skeleton piece that shares a junction node with another subgraph may compromise its topology locally [40], and therefore cannot precisely describe the local topology feature of the partitioned component. In this case, we search along the skeleton and further partition it with an additional cut. Refer to Figure 5 for an illustration. Empirically, we found this rarely happened.

5 RESULTS

Partition Examples: Without any partitioning operation, the inner and outer boundary of of a shell model needs to be filled by a huge amount of support materials in order to guarantee a fine surface quality. See Figure 6 (a-b) for an illustration of the sculpture model, both its inner and outer surface require significant amount of support in order not to be collapsed during the printing process; while our approach only keeps all cylindrical shells that are free of support (Figure 6 (c-d)).

We have run our algorithm on various natural or man-made models, and some of the results are presented as follows:

Experimental Validation: We validated our approach by a set of printing experiments on Zortrax desktop printer, a kind of FDM machine that allows a printing layer thickness of 0.09mm , this is also the layer thickness we used in the printing experiments. The

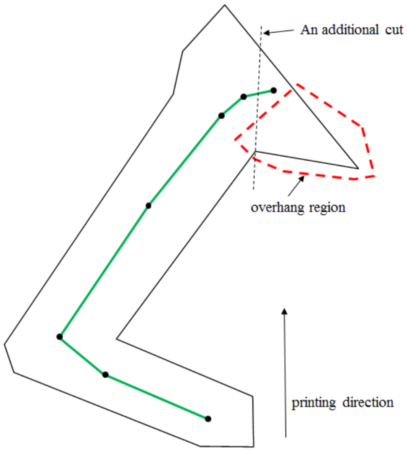


Fig. 5. Illustration of a mesh component with an overhange that requires support while its corresponding skeleton is support-free

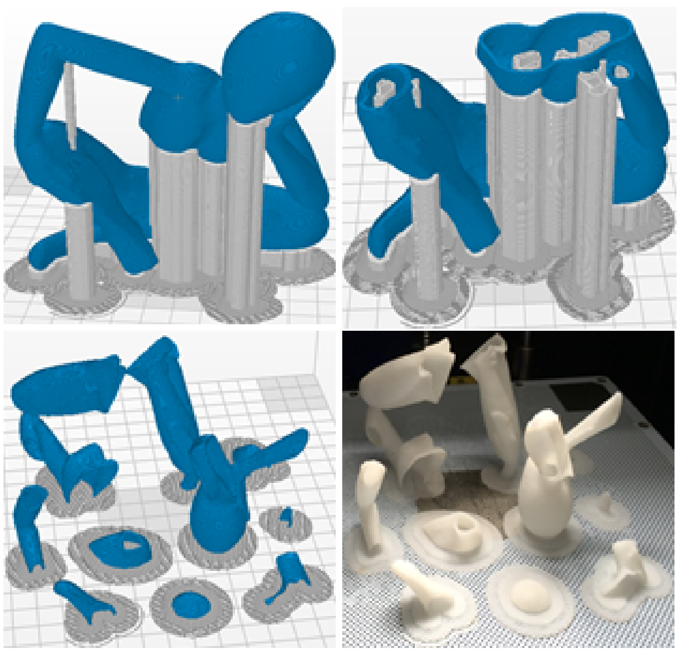


Fig. 6. An illustration of the sculpture model under the 3D printing software Z-suite as $\theta = 20°$; (a) the full model; (b) an intermediate step of the simulation; (c) the simulation result of our partitioned shell models; (d) the 3D printed shell parts free of support.

experiments are based on the choice of $\theta = 70°$, i.e., all overhangs with an angle of no larger than $20°$ with respect to the build platform are given support structures. Zortrax provides a built-in 3D printing software called *Z-suite* can automatically counts the filament of the print material (in meters) and an estimate of the weight of the print material. The following table summarizes the printing material and time costs by the original models and the partitioned models. Our approach significantly reduces both the printing material and time as the skeleton-based partition reduces both the supported materials inside and outside the models.

Figure 8 shows the comparison of the printing effects of the original models and our partitioned models. Although the FDM printing technique requires a small supporting bed for holding the printed model on the printing platform, other printing techniques such as SLA, SLM and SLS may avoid the use of these supporting

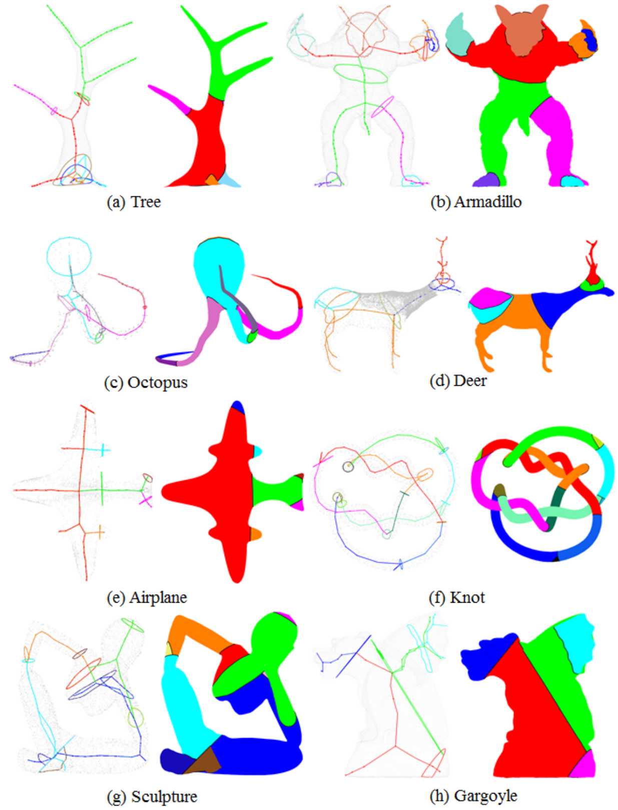


Fig. 7. Partition examples as $\theta = 70°$. The printing direction of each part is orthogonal to its base (shown in the same color as the part)

beds. Therefore, our approach guarantees support-free to the most extent for all exiting printing techniques.

[Evaluations: how close we are to the global optimum][Limitations: discussion about 1. the strength of our interior, – future work. 2. cut through salient regions – hard to balance. etc]

Since the problem of partitioning the skeleton graph into the least number of support-free subgraphs is NP-hard, it is impossible to obtain an optimal result in polynomial time. However, we can evaluate the effectiveness of our approach by judging the topology of some simple models and evaluate how close our partition is from a potential optimal result. For example, from Figure 1, under the requirement of $\theta = 70°$, ignoring the tinny detailed geometric features we can observe that an optimal partition of the deer model requires at least 4 cuts: the tail, the chunk including the legs, the neck, and the head and the horn. Our approach provides a partition of 4 cuts, which is almost near the optimal. Further, from the appearance of the gargoyle model (the last row of Figure 8), we can see that an optimal cutting should have the following 4 parts: the head, two wings, and the remaining parts. Our partition results in 5 parts, which is very close to the potential optimal partition.

Our approach is devoted to shell models, it can also be applied to cutting solid models without any problem. However, our approach suffers from a few limitations: for shell models, the thickness of the shells need to be large enough such that no serious deformation is caused during the assembling process. However, the problem of setting the minimum thickness of the model for various parts of the model is non-trivial, and our current work is restricted to a uniform setting of the shell thickness whose value is determined by an error-and-trial process. Further, the strength guarantee is not elaborated in this work, a possible solution is

Models	Print material (original)	Print time (original)	Number of parts (partition)	Print material (partition)	Print time (partition)	Material save (%)	Time save(%)
Tree	9.07m (22g)	5h 5min	8	5.19(12g)	4h 2min	42.7784	20.6557
Armadillo	9.85m (23g)	5h 56min	10	4.65(11g)	3h 30min	52.7919	41.0112
Octopus	14.58m (35g)	8h 7min	9	8.75(21g)	6h 51min	39.9863	15.6057
Deer	14.24m (34g)	8h 46min	6	8.89(21g)	5h 25min	37.5702	38.2129
Airplane	9.86m (23g)	4h 15min	7	5.24(12g)	3h 10min	46.856	25.4902
Knots	26.96m (64g)	17h 44min	15	12.38(29g)	11h 20min	54.0801	36.0902
Sculpture	29.17m (69g)	15h 12min	10	16.35(39g)	11h 23min	43.9493	25.1096
Gargoyle	7.88m (19g)	4h 34min	5	5.16(12g)	3h 48min	34.5178	16.79

TABLE 1
Statistics showing the print material, print time and partition number of the printed models.

to use the algorithm in [12] that distributes the least amount of materials for constructing a truss frame beneath the skin. Finally, our approach may allow a cut that passes through a salience region, which may hurt the appearance of the model. We found that it is difficult to make a balance between the saliency and the minimal cutting length as well as the minimal cutting numbers. A potential future research is to take care of salience region during graph partition. Finally, as a trade-off, a spatially curved cut might be a consideration to alleviate the salience problem.

6 Conclusion

In this paper, we present a skeleton-based approach for partitioning a shell model into parts which are free of supporting structure when fabricated. We formulate the model partition problem as a constrained graph partition problem which is particularly tailored toward 3D fabrication. To tackle the NP-hardness of the problem, we exploit a randomized Monte Carlo method which adaptively searches for better partition possibilities while avoiding local minima. Compared with existing partition-based methods, the advantages of our partition method are as follows:

- The models are support-free, especially for the 3D printing techniques including SLA, SLM and SLS. For FDM technique, it requires a bed of support that consumes very little volume of materials.
- The seams on the assembled model are minimized in terms of cut number and cut length.

The support-free feature of our partition approach saves a significant amount of time and printing materials both inside and outside the model of a shell model. Finally, our method is efficient and applicable to a large set of natural and man-made models.

Acknowledgments

The authors would like to thank...

This work was supported in part by Shanghai Sailing Program No.16YF1405500, Shanghai Jiao Tong University Grant No. AF0200163, and State Key Laboratory of Mechanical Systems and Vibration Grant no. MSVZD201505.

References

[1] J. Dumas, J. Hergel, and S. Lefebvre, “Bridging the gap: automated steady scaffoldings for 3d printing,” *ACM Trans. Graph.*, vol. 33, no. 4, pp. 98:1–98:10, 2014.

[2] J. Vanek, J. A. G. Galicia, and B. Benes, “Clever support: Efficient support structure generation for digital fabrication,” *Comput. Graph. Forum*, vol. 33, no. 5, pp. 117–125, 2014.

[3] X. Zhang, X. Le, A. Panotopoulou, E. Whiting, and C. C. L. Wang, “Perceptual models of preference in 3d printing direction,” *ACM Trans. Graph.*, vol. 34, no. 6, pp. 215:1–215:12, 2015.

[4] K. Hildebrand, B. Bickel, and M. Alexa, “Orthogonal slicing for additive manufacturing,” *Computers & Graphics*, vol. 37, no. 6, pp. 669–675, 2013.

[5] N. Padhye and K. Deb, “Multi-objective optimisation and multi-criteria decision making in sls using evolutionary approaches,” *Rapid Prototyping Journal*, vol. 17, no. 6, pp. 458–478, 2011.

[6] R. Hu, H. Li, H. Zhang, and D. Cohen-Or, “Approximate pyramidal shape decomposition,” *ACM Transactions on Graphics, (Proc. of SIGGRAPH Asia 2014)*, vol. 33, no. 6, pp. 213:1–213:12, 2014.

[7] J. Vanek, J. A. G. Galicia, B. Benes, R. Mech, N. A. Carr, O. Stava, and G. S. P. Miller, “Packmerger: A 3d print volume optimizer,” *Comput. Graph. Forum*, vol. 33, no. 6, pp. 322–332, 2014.

[8] P. Song, Z. Fu, L. Liu, and C. Fu, “Printing 3d objects with interlocking parts,” *Computer Aided Geometric Design*, vol. 35-36, pp. 137–148, 2015.

[9] A. S. Fukunaga and R. E. Korf, “Bin completion algorithms for multicontainer packing, knapsack, and covering problems,” *J. Artif. Int. Res.*, vol. 28, no. 1, pp. 393–429, Mar. 2007.

[10] O. Stava, J. Vanek, B. Benes, N. A. Carr, and R. Mech, “Stress relief: improving structural strength of 3d printable objects,” *ACM Trans. Graph.*, vol. 31, no. 4, pp. 48:1–48:11, 2012.

[11] Q. Zhou, J. Panetta, and D. Zorin, “Worst-case structural analysis,” *ACM Trans. Graph.*, vol. 32, no. 4, pp. 137:1–137:12, 2013.

[12] W. Wang, T. Y. Wang, Z. Yang, L. Liu, X. Tong, W. Tong, J. Deng, F. Chen, and X. Liu, “Cost-effective printing of 3d objects with skin-frame structures,” *ACM Trans. Graph.*, vol. 32, no. 6, pp. 177:1–177:10, 2013.

[13] N. Umetani and R. Schmidt, “Cross-sectional structural analysis for 3d printing optimization,” in *SIGGRAPH Asia 2013 Technical Briefs*, ser. SA ’13, 2013, pp. 5:1–5:4.

[14] L. Lu, A. Sharf, H. Zhao, Y. Wei, Q. Fan, X. Chen, Y. Savoye, C. Tu, D. Cohen-Or, and B. Chen, “Build-to-last: strength to weight 3d printed objects,” *ACM Trans. Graph.*, vol. 33, no. 4, pp. 97:1–97:10, 2014.

[15] M. Skouras, B. Thomaszewski, S. Coros, B. Bickel, and M. H. Gross, “Computational design of actuated deformable characters,” *ACM Trans. Graph.*, vol. 32, no. 4, pp. 82:1–82:10, 2013.

[16] S. Coros, B. Thomaszewski, G. Noris, S. Sueda, M. Forberg, R. W. Sumner, W. Matusik, and B. Bickel, “Computational design of mechanical characters,” *ACM Trans. Graph.*, vol. 32, no. 4, pp. 83:1–83:12, 2013.

[17] D. Ceylan, W. Li, N. J. Mitra, M. Agrawala, and M. Pauly, “Designing and fabricating mechanical automata from mocap sequences,” *ACM Trans. Graph.*, vol. 32, no. 6, pp. 186:1–186:11, 2013.

[18] M. Bächer, B. Bickel, D. L. James, and H. Pfister, “Fabricating articulated characters from skinned meshes,” *ACM Trans. Graph.*, vol. 31, no. 4, pp. 47:1–47:9, 2012.

[19] J. Calì, D. A. Calian, C. Amati, R. Kleinberger, A. Steed, J. Kautz, and T. Weyrich, “3d-printing of non-assembly, articulated models,” *ACM Trans. Graph.*, vol. 31, no. 6, p. 130, 2012.

[20] M. Bächer, E. Whiting, B. Bickel, and O. Sorkine-Hornung, “Spin-it: optimizing moment of inertia for spinnable objects,” *ACM Trans. Graph.*, vol. 33, no. 4, pp. 96:1–96:10, 2014.

[21] R. Prévost, E. Whiting, S. Lefebvre, and O. Sorkine-Hornung, “Make it stand: balancing shapes for 3d fabrication,” *ACM Trans. Graph.*, vol. 32, no. 4, pp. 81:1–81:10, 2013.

[22] L. Luo, I. Baran, S. Rusinkiewicz, and W. Matusik, “Chopper: partitioning models into 3d-printable parts,” *ACM Trans. Graph.*, vol. 31, no. 6, p. 129, 2012.

[23] J. Hao, L. Fang, and R. E. Williams, “An efficient curvature-based partitioning of large-scale stl models,” *Rapid Prototyping Journal*, vol. 17, no. 2, pp. 116–127, 2011.

Fig. 8. A comparison between printing of original shell models and our partitioned models.

[24] S. Xin, C. Lai, C. Fu, T. Wong, Y. He, and D. Cohen-Or, "Making burr puzzles from 3d models," *ACM Trans. Graph.*, vol. 30, no. 4, p. 97, 2011.

[25] P. Song, C. Fu, and D. Cohen-Or, "Recursive interlocking puzzles," *ACM Trans. Graph.*, vol. 31, no. 6, p. 128, 2012.

[26] A. Shamir, "A survey on mesh segmentation techniques," *Comput. Graph. Forum*, vol. 27, no. 6, pp. 1539–1556, 2008.

[27] S. Katz and A. Tal, "Hierarchical mesh decomposition using fuzzy clustering and cuts," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 954–961, 2003.

[28] S. Katz, G. Leifman, and A. Tal, "Mesh segmentation using feature point and core extraction," *The Visual Computer*, vol. 21, no. 8-10, pp. 649–658, 2005.

[29] Z. Ji, L. Liu, Z. Chen, and G. Wang, "Easy mesh cutting," *Comput. Graph. Forum*, vol. 25, no. 3, pp. 283–291, 2006.

[30] R. Liu and H. Zhang, "Mesh segmentation via spectral embedding and contour analysis," *Comput. Graph. Forum*, vol. 26, no. 3, pp. 385–394, 2007.

[31] A. Golovinskiy and T. Funkhouser, "Randomized cuts for 3D mesh analysis," *ACM Transactions on Graphics (Proc. SIGGRAPH ASIA)*, vol. 27, no. 5, Dec. 2008.

[32] X. Chen, A. Golovinskiy, and T. A. Funkhouser, "A benchmark for 3d mesh segmentation," *ACM Trans. Graph.*, vol. 28, no. 3, 2009.

[33] O. van Kaick, N. Fish, Y. Kleiman, S. Asafi, and D. Cohen-Or, "Shape segmentation by approximate convexity analysis," *ACM Trans. Graph.*, vol. 34, no. 1, pp. 4:1–4:11, 2014.

[34] D. D. Hoffman and W. A. Richards, "Parts of recognition," *Cognition*, vol. 18, no. 1, pp. 65–96, 1984.

[35] D. D. Hoffman and M. Singh, "Salience of visual parts," *Cognition*, vol. 63, no. 1, pp. 29–78, 1997.

[36] K. Zhou, J. Synder, B. Guo, and H.-Y. Shum, "Iso-charts: stretch-driven mesh parameterization using spectral analysis," in *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*. ACM, 2004, pp. 45–54.

[37] I. García and G. Patow, "Igt: Inverse geometric textures," *ACM Trans. Graph.*, vol. 27, pp. 137:1–137:9, December 2008.

[38] J.-M. Lien, J. Keyser, and N. M. Amato, "Simultaneous shape decomposition and skeletonization," in *Proceedings of the 2006 ACM symposium on Solid and physical modeling*. ACM, 2006, pp. 219–228.

[39] D. Reniers and A. Telea, "Skeleton-based hierarchical shape segmentation," in *Shape Modeling and Applications, 2007. SMI'07. IEEE International Conference on*. IEEE, 2007, pp. 179–188.

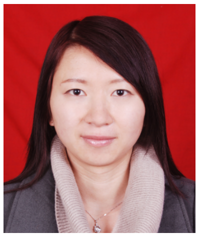
[40] O. K. Au, C. Tai, H. Chu, D. Cohen-Or, and T. Lee, "Skeleton extraction by mesh contraction," *ACM Trans. Graph.*, vol. 27, no. 3, 2008.

[41] Y. Zhou, K. Yin, H. Huang, H. Zhang, M. Gong, and D. Cohen-Or, "Generalized cylinder decomposition," *ACM Trans. Graph.*, vol. 34, no. 6, pp. 171:1–171:14, Oct. 2015.

[42] X. Zhang, Y. Xia, J. Wang, Z. Yang, C. Tu, and W. Wang, "Medial axis tree - an internal supporting structure for 3d printing," *Computer Aided Geometric Design*, vol. 35-36, pp. 149–162, 2015.

[43] O. K.-C. Au, Y. Zheng, M. Chen, P. Xu, and C.-L. Tai, "Mesh segmentation with concavity-aware fields," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 18, no. 7, pp. 1125–1134, 2012.

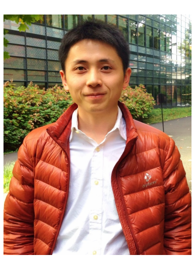
Xiangzhi Wei is an Assistant Professor with the Institute of Intelligent Manufacturing and Information Engineering, School of Mechanical Engineering, Shanghai Jiao Tong University. He received his PhD and MPhil degree in Industrial Engineering and Logistics Management from the Hong Kong University of Science and Technology; and BE degree in Mechanical Engineering from Beijing Jiao Tong University. His current research interests include Additive Manufacturing (3D printing), computational geometry, algorithm design and analysis.



Siqi Qiu is currently an Assistant Professor with the Institute of Intelligent Manufacturing and Information Engineering, School of Mechanical Engineering, Shanghai Jiao Tong University. She received her PhD degree from the Department of Computer Engineering, Compiegne University of Technology, her MSc degree from Department of Electronic Engineering, University of Paris-Sud, and her BSc degree from School of Optical Science and Engineering, Huazhong University of Science and Technology. Her research interests include Monte Carlo methods, System of Systems, and reliability analysis under uncertainty.

Lin Zhu is a PhD Candidate with the Institute of Intelligent Manufacturing and Information Engineering, School of Mechanical Engineering, Shanghai Jiao Tong University. She received her ME degree in Mechanical Engineering from the Xian Jiao Tong University; and BE degree in Packaging Engineering from Xi'an University of Technology. Her current research interests include Additive Manufacturing, Design of Experiments and Analysis.

Juntong Xi received his PhD degree in Mechanical Engineering from Xi'an Jiaotong University. He is a full Professor with the Institute of Intelligent Manufacturing and Information Engineering, School of Mechanical Engineering, Shanghai Jiaotong University, Shanghai, China. His current research interests include digital manufacturing, measurement technology and instruments, and dental CAD.



Youyi Zheng is currently an Assistant Professor at the School of Information Science and Technology, ShanghaiTech University. He obtained his PhD from the Department of Computer Science and Engineering at Hong Kong University of Science & Technology, and his M.Sc. and B.Sc. degrees from the Department of Mathematics, Zhejiang University. His research interests include geometric modeling, imaging, and human-computer interaction.