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

Xiangzhi Wei* Siqi Qiu* Lin Zhu Ruiliang Feng Yaobin Tian Juntong Xi Youyi Zheng†

Abstract—Minimizing support structures is crucial in reducing 3D printing material and time. Partition-based methods are efficient means in realizing this objective. Although some algorithms exist for support-free fabrication of solid models, no algorithm ever considers the problem of support-free fabrication for *shell* models (i.e., hollowed meshes). In this paper, we present a skeleton-based algorithm for partitioning a 3D surface model into the least number of parts for 3D printing without using any support structure. To achieve support-free fabrication while minimizing the effect of the seams and cracks that are inevitably induced by the partition, which affect the aesthetics and strength of the final assembled surface, we put forward an optimization system with the minimization of the number of partitions and the total length of the cuts, under the constraints of support-free printing angle. Our approach is particularly tailored for shell models, and it can be applicable to solid models as well. We first rigorously show that the optimization problem is NP-hard and then propose a stochastic method to find an optimal solution to the objectives. We propose a polynomial-time algorithm for a special case when the skeleton graph satisfies the requirement that the number of partitioned parts and the degree of each node are bounded by a small constant. We evaluate our partition method on a number of 3D models and validate our method by 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 [1]. Fused deposition modeling (FDM), stereolithography (SLA), Selective Laser Melting (SLM) and Selective Laser Sintering (SLS) are 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. For example, the price of fine plastics used in Stratasys FDM machines is 300-400$/kg, the price of the resins for Object machines is 500-600$/kg; and the price of fine Ti powders used in SLM machines is usually higher than 1000$/kg. Therefore, it is desirable to reduce the amount of materials used in the fabrication process. Note that this is also a critical operation for reducing production time and thus 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.

Autodesk MeshMixer[1] 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 methods have studied various factors that influence the volume of supports, e.g., optimizing the topology of the support structure [2], [3], determining an optimal fabrication direction [4], [5], [6], partitioning any given model into a set of separate parts that satisfy particular geometric properties such as being pyramidal [7], has minimal packing volume [8] or being inter-lockable [9].

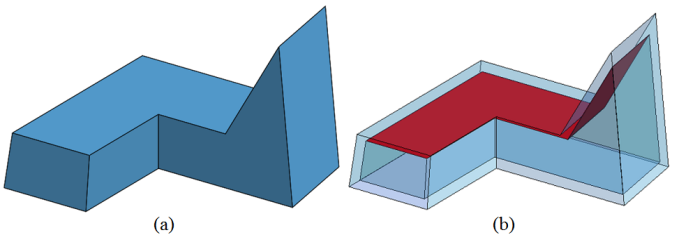


Fig. 1. (a) A solid pyramid used in [7], and (b) a shell version of the model, where the overhang region is highlighted in red. The former can be printed in a support-free manner, while the overhang region of the latter needs support from below.

Although several algorithms exist for partitioning solid models in an (almost) support-free fashion [7] for 3D printing (i.e., no support structures are required to support the overhang regions of the model since the slope of each facet with respect to the build platform is large enough such that a 3D printed layer can prevent its upper layer from falling down), no existing methods ever consider the problem of partitioning a *shell* model which we call, whose boundary contains two meshes: the outer one and the inner one, the component in-between is solid, and the closure of the inner mesh is not solid (see Fig. 1, right), into the least number of parts whose fabrication is free of support structures. Shell models are widely used in many mechanical and artistic lightweight designs where prototypes are needed. Therefore, studying how to fabricate shell models in a support-free manner is particularly useful for lightweight applications.

* *co-first author*
† *corresponding author*

- *X. Wei, S. Qiu, L. Zhu, R. Feng 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,fengruiliang,jtxi}@sjtu.edu.cn*
- *Yaobin Tian is with the Department of Industrial Engineering and Logistics Management of Hong Kong University of Science and Technology, Hong Kong. E-mail: ybtian@ust.hk*
- *Youyi Zheng is with the State Key Lab of CAD&CG, Zhejiang University, China. E-mail: zyy@cad.zju.edu.cn*

1. http://www.meshmixer.com/

There is a great difference between the two types of models in choosing the support-free printing direction: for a solid pyramid, a large facet is usually set as the base; while for a shell one, to find the optimal support-free printing direction, a large face simply does not suffice (see Fig. 1). Minimizing the number of portioned parts corresponds to finding 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 plastics, etc.

This paper addresses the problem of decomposing a 3D shell model into the least number of parts, each of which, when printed in a proper direction, is free of support materials. Finding such optimal partition directly on the surface mesh is challenging as the number of mesh triangles are typically large ($10 \sim 100k$) and the running time of any direct searching algorithm (e.g., randomly start from a face and search for a support-free surface part) could be exponential. We simplify the problem by considering a reduced representation. In particular, we draw inspiration from the 1D skeleton of organic models: the topology variations of a natural model can be well-represented by its skeleton, and a segment of the skeleton corresponds to a mesh part. Furthermore, the mesh part is typically a cylinder-like shape which can be printed free of support structures in most cases if the printing direction is parallel to the skeletal direction (Fig. 2).

We restrict our focus on the models with locally tubular shapes that can be well-described by skeletons. This covers a large range of articulated or organic models [10]. For a shell model, support structures are required for both the interior and exterior surfaces of a mesh model during the 3D printing process. Our approach assumes that the interior of a mesh model is hollow 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 parts, such that each part is represented by a skeletal subgraph and can be fabricated in a good printing direction without using any support structure.

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 partition the skeleton into a minimum set of subgraphs (i.e., the least number of subgraphs), such that each arc in any subgraph subtends to an axis by an angle of no large than θ, the corresponding chunk of the mesh is therefore support-free if printed along this axis (see Fig. 3).

Decomposing a skeleton graph into a minimal set of subgraphs that satisfy the support-free constraints is a non-trivial task. In this paper, we first rigorously show that the partition problem is NP-hard by reducing it to a known NP-Complete problem and we also show that the optimal solution can be found in polynomial time in special cases. Then, we present a unified stochastic framework to handle the general cases by simultaneously looking for the best set of subgraphs that are both support-free and having minimal partition length while matching a set of fabrication constraints. In short, our method makes the following contributions:

- We devise a practical solution to the problem of support-free mesh partition by formulating it as graph partition problem with constraints;
- We show the graph partition problem with support-free constraints is NP-hard;
- We offer a viable and very efficient solution to tackle the graph partition problem using a semi-greedy stochastic algorithm.

The remainder of the paper is organized as follows: Section 2 provides a review of the related works, section 3 provides a closed-form solution to the problem, section 4 provides a stochastic method for the problem, section 5 provides the experimental results of our proposed methods, section 6 concludes the paper with some discussions.

2 Related Work

Our work is focused on model partition, with the particular purpose for support-free fabrication. Here we briefly review a set of most relevant recent works.

Computational Fabrication. Recently, an increasing body of research work has been devoted to computational fabrication using 3D printing as the emergence of advanced 3D printing devices. 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 [11], [12], [13], [14], [15], [16], [17], [18]. The printability of critical structures (e.g., bridges, spikes, holes, etc.) during the FDM processes has been investigated [2], [19]. The modeling of some particular features have also been studied, for example, surface quality [20], deformation behavior [21], animated mechanical characters [22], [23], articulated models with mobile joints [24], [25], models spinnable motions [26], and self-balancing [27].

Shape Decomposition. In geometry processing, decomposing a shape into meaningful parts is a fundamental problem [28]. Many efforts have been devoted to the problem of model decomposition. An excellent survey can be found in [29]. A large body of research is focusing on partitioning a given 3D model into parts which agree with human perception [30], [31], [32], [33], [34], [35], [36], among which geometric features captures shape concaveness are mostly exploited in accordance with the minima rules [37], [38]. Other approaches are more application-oriented. For example, texture mapping techniques often require the input shape being decomposed into charts which can be flatten to match image textures [39], [40]. Our method of partition is based on shape skeletons. Although there has been previous work on skeleton-based shape decomposition [41], [42], [43], none of them is designed for support-free fabrication.

Fabrication-driven Model Partition. 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. [44] 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. [45] partitioned a large complex model into simpler 3D printable parts by using curvature-based partitioning. Hildebrand et al. [5] 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. [8] 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

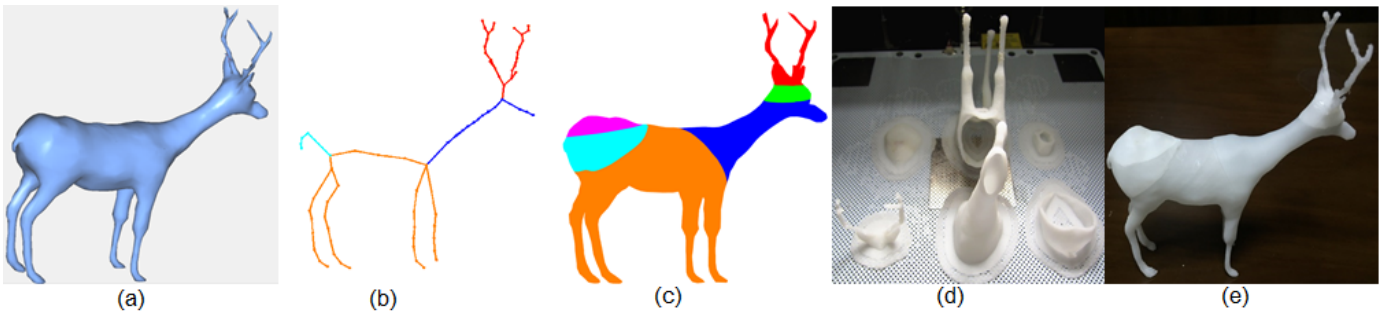


Fig. 2. A deer model is partitioned into support-free parts using our algorithm: (a) a shell model; (b) skeleton partition by our approach; (c)mesh partition by our approach; (d)the support-free printing of the parts; (e) the assembly result of the printed parts.

volume of each segment are considered during the optimization process. Attene et al. [46] introduced a method for decomposing a model into parts which can be packed in a box. Yao et al. [47] further investigated the problem of optimized partition and packing by considering multiple metrics and integrated level-set methods into a unified framework. Song et al. [9] recently developed a novel voxelization-based approach to construct interlocking 3D parts from a given 3D model. In their latest work, Song et al. [48] combined 3D printing and 2D laser cutting for cost-effective fabrication of large objects. Without using any glue, Xin et al. [49] and Song et al. [50] took a 3D interlocking approach to construct and connect printed 3D parts to form an object assembly. None of these methods considers the problem of support-free fabrication. Most recently, Hu et al. [7] proposed an nice algorithm for decomposing a solid model into the least number of pyramids, each of which can be fabricated in a support-free manner. However, their algorithm is only designed for volumetric models while our algorithm aims for both shell and solid models.

3 A Closed-form Solution

In nature, most organic models ubiquitously contain cylindrical parts [51], e.g., arms, legs, etc. Moreover, a cylindrical part can be well represented by its corresponding curved skeleton. Based on this property, a chunk of a mesh model can be fabricated free of support if all the arcs of its corresponding skeleton subtend to a printing direction by an angle of no large than θ (Fig. 4). Hence, our problem becomes partitioning a given 1D skeleton graph into a set of subgraphs such that each subgraph has 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.

Choice of skeleton. Compared to natural skeletons, the medial axis can describe the topology of a mesh model more precisely [52]. 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 graphs [53] can be used to approximate the topologies of the mesh models. During the generation process of any Reeb graph, the slicing direction and the position of the representative node on each slice (a connected region) seriously influence the shape of the graph. But, how to determine the slicing direction and the representative nodes such that the shape of the resulting graph captures the geometric changes of the mesh is a difficult problem. There are quite a few methods that generate curve skeletons from a 3D model. For example, medial surface based method which contracts the medial axis surface of the model [52]; and the generalized field method which traces out curves seeded at critical points along high-divergence directions [54]; and contraction methods which shrink the mesh into curves [43], [55]. It is hard to analytically compare the properties of the curve skeletons due to the lack of a unanimously accepted formal definition of skeleton. For a complete survey of these methods, the readers are referred to a survey in [55]. To carry out our methodology of partitioning the model based on curve skeletons, we resort to the Laplacian skeleton proposed by [43] which is extracted by shrinking the inner mesh of the shell model using Laplacian smoothing. Such a curved skeleton provides an excellent choice for reasonably describing the geometrical and topological variations of any 3D model. The Laplacian skeleton represents the models very well if the following two conditions are met: (1) each critical topological feature of the model is captured by the skeleton; (2) the skeleton is dense enough to capture the geometric variations of the original shape.

For condition (1), we use the Laplacian skeleton provided by the authors of [43], which has been shown to be capable of segmenting complicated models in a nice way. For condition (2), the density of the skeleton, one can always add arcs (line segments) into the skeleton and obtain a finer representation of a smaller strip of the mesh model. In special cases, a portion of the mesh is not represented properly by a Laplacian skeleton arc if it is a tinny detail that has been shrunk into a point during just a few iterations of the Laplacian smoothing process. We find that these details can usually be printed out in a support-free manner if the overhang angle is not that small (e.g., larger than 10 degrees) with

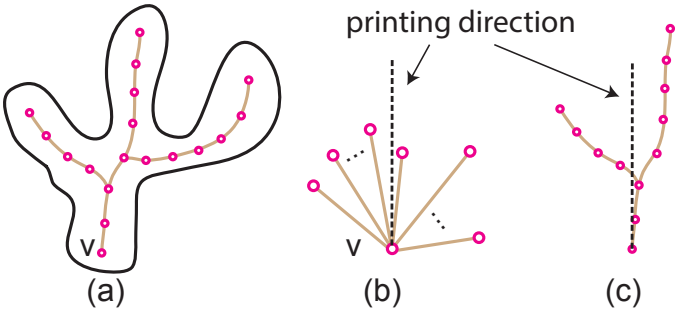


Fig. 3. A segment (a) of a skeleton corresponds to a fork (b) originated from a node. (c) A subgraph which can be 3D printed in a support-free manner.

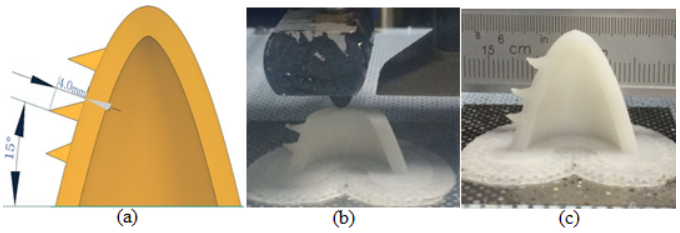


Fig. 4. Support-free printing of a model with some spikes on it: the length of the spikes is $4mm$, the overhang angle is 15 degrees, the model is fabricated by a Zortrax M200 3D printer.

respect to the building platform, and that the overhang length is within a safe distance (e.g., 1-5mm) when a nozzle jumps from one point to another. See Fig. 4 for an illustration. We remark that this distance is long enough to address most cases when small details are not captured by a skeleton arc: it can also be printed out by any prevailing desktop 3D printer in a support-free manner due to the cohesive and elastic forces that are inherent to the plastic material itself.

Problem Statement. Our goal is hence to decompose the Laplacian skeleton of the model into a proper number of support-free subgraphs leading to a partition of the model into the least number of printable parts free of support structures and seams on the final assembled model. A subgraph can be a common graph with multiple loops or it can be a tree without any loop. 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.

Finding a decomposition of the skeleton into the least pieces of support-free subgraphs is a non-trivial task. As a simple example, suppose for simplicity that the least number of cuts on the skeleton corresponds to the least number of cuts on the mesh model. Refer to Fig. 3 for a Laplacian skeleton that is a fork with n arcs sharing a common origin. In this example, our task is to partition 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, where θ is the safe angle for support-free fabrication.

Theorem 1*: Partitioning a graph of n arcs into the least number of support-free subgraphs is NP-hard.*

Proof: We shall complete the proof by transforming an instance of a known NP hard problem into our problem in polynomial time. First we need to show that our problem is in NP: The certificate is a set of arc-disjoint and rooted subgraphs partition of the input graph (skeleton), a certifier checks in polynomial time (i.e., $O(n^2)$) that the number of subgraphs is at most the given bound K, and that the rooted subgraphs satisfy the angle constraint of 2θ. We shall reduce the Clique Cover problem to the skeleton partition problem, where the Clique Cover problem is covering a graph with the least number of cliques (complete graphs), which has been shown to be NP-hard [56]. We now show that Clique Cover $\leq p$ Skeleton Partition (i.e., Clique Cover is polynomial-time reducible to Skeleton Partition). It suffices to show that there exists an instance of Skeleton Partition problem which is NP-hard. Let us consider the following instance of Clique Cover: an arbitrary planar $G(V, E)$ such that $|V| = n$ and each pair of nodes (v_i, v_j) is connected by an arc if $||v_iv_j||$ is no larger than a given bound D. We construct a skeleton S of n arcs rooted at a common node (a fork) as follows: select any triplet of nodes v_i, v_j, v_k from G and construct a triplet of unit arcs $\{e_i, e_j, e_k\}$ in S, the angle between each pair of arcs $\{e_i, e_j\}$, denoted as $A(e_i, e_j)$, is defined as $A(e_i, e_j) = 2\theta||v_iv_j||/D$. See Fig. 3, given this triplet of arcs as the basis, we can construct each of the remaining arcs of S, denoted as e_x, which represents a node v_x in G: the relative position of e_x with respect to each element of $\{e_i, e_j, e_k\}$ is defined as the distance from v_x to each element of $\{v_i, v_j, v_k\}$. This transformation takes $O(n^2)$ time.

Now we claim that there is a clique cover in G of size K if and only if there is a skeleton partition of S into K support-free subgraphs (the angle between each pair of arcs is no larger than 2θ). For if there is clique cover in G of size K, then each clique in G corresponds to a rooted subgraph of S that satisfies the angle constraint. Conversely, if S is partitioned into K support-free subgraphs, then every pair of arcs in each subgraph satisfy the angle constraint, by the mapping relation, their corresponding nodes in G form a clique. This completes the proof. □

This tells the difficulty of solving the problem of partitioning a 3D model into the least number of support-free pieces. In the following, we showcase the problem in different scenarios and show that there exists polynomial solutions when the graph satisfies certain conditions.

Tree case. If the topology of the skeleton S is a tree such that the degree of each node is bounded by a constant d, and the least number of support-free subgraphs is smaller than a constant c, we show that the problem of partitioning S into the least number of support-free subgraphs (satisfying the angle constraint) can be computed in polynomial time. In general, for a tree structure with arbitrary c and d, we have then following theorem.

Theorem 2*: Given two integer numbers c and d, let S be a tree structure such that the degree of each node is bounded by d, then whether S has c support-free subgraphs can be determined in $O(2^{cd}n^{2c})$ time, and a partition instance can be reported within the same time bound.*

Proof: We shall prove the theorem by construction. In taking a subgraph from a tree structure S, we can duplicate a node v and take a subset of arcs incident to it. Given a node with d arcs incident to v, then the number of subsets of arcs incident to it is $C(d, 1) + C(d, 2) + ... + C(d, d-1) = O(2^d)$. In order to construct a partition of S into c subgraphs, we need to choose at most c nodes from S (it is possible that multiple subgraphs are derived from a common node). More precisely, we need to choose i nodes from n nodes of S for $1 \leq i \leq c$. Since there are $O(2^d)$ choices for each node, the number of all possible partitions is bounded by $O(\sum_{i=1}^{c} C(n, i) * C(i * 2^d, c)) = O(2^{cd}n^c)$. For the resulting partition, we need to make sure that it is a valid partition, i.e., no arc is contained in more than one subgraph. This can be done in $O(n)$ time by counting the total number of arcs in the resulting partition: if it is equal to $(n-1)$, then the partition is valid. For each valid partition, we need to determine whether a subgraph is support-free. For this purpose, in each valid partition, for a subgraph of size n_i, it takes $O(n_i)$ time to check whether the subgraph is a support-free one given a node as the root. Therefore it takes $O(n_i^2)$ time for processing all nodes in the subgraph. In sum, it takes $O(\sum_{i=1}^{c} n_i^2) = O(n^2)$ time to check whether a partition is a support-free one. Therefore, for all $O(2^{cd}n^c)$ partitions, it requires $O(2^{cd}n^c * n^2) = O(2^{cd}n^{2c})$ time. This completes the proof. □

As a result, if c^2 is a constant, then partitioning S into c support-free subgraphs can be done in polynomial time. Further, the least number of subgraphs can be determined by a standard binary operation on c. More precisely, if c is sufficient to obtain a feasible support-free partition, then we can try $c/2$, and then $c/2^2$ if $c/2$ is also sufficient, and so on so forth, finally, a half interval is added back, and the number before and after this resulting number are used to locate the final value of the smallest number. The construction process in the proof of Theorem 1 directly suggests a polynomial time algorithm for computing the least number of support-free subgraphs when S is a tree structure with constant c and d.

General case. Next we consider the general case when the graph is not a tree (i.e., contains cycles). If the genus (number of handles) of a model is h , then its corresponding skeleton contains h disjoint cycles, where h can be determined by the Euler formula on 3D mesh models; it requires at least h splitting nodes to split G into a tree structure (which has $C(n, c) = O(n^c)$ choices); thereafter, Theorem 2 addresses the remaining splitting issue. To summarize, we have the following theorem for a general graph G.

Theorem 3*: Given three integer numbers c, d and h, let S be a general graph with a genus of h such that the degree of each node is bounded by d, then whether S has c support-free subgraphs can be determined in $O(2^{cd}n^{2c+h})$ time, and a partition instance can be reported within the same time bound.*

We formulate the partition problem with the objectives of both 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 cutting 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.

Even though we have shown cases when the problem has polynomial time solutions, exhaustive search is still computationally prohibitive in complex cases (e.g., when the graph contain multiple cycles and the value of c and d are large). Next, we propose a randomized stochastic method in compliance with a set of carefully designed selection strategy to seek a practical solution.

4 Stochastic Method

Let M denote the mesh model, and let S denote the Laplacian skeleton obtained via the algorithm provided in [43]. We propose an algorithm for partitioning M into a minimum set of *disjoint* components, each of which can be fabricated by a 3D printer without using any support structure. Decomposing S into two pieces can be done by duplicating a node v; while partitioning M at node v requires the determination of the position and normal of a cutting plane. Particularly, the angle between the normal of the cutting plane and the printing direction should be less than or equal to θ in order to guarantee the requirement of support-free fabrication. To guarantee an aesthetical look of 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 constrains the orientation of the cutting plane. In addition, we need the assumption that the volume of the printing model should be within the working volume of a given 3D printer.

2. c might be proportional to the number of skeleton arcs (for example on a shape consisting of a union of spring-link parts).

In a precise 3D printer with an error of 16-100um (e.g., with SLA and FDM techniques), we find that the deformation on the matching interfaces is subtle and can be ignored. Although the length of the cuts accounts for the aesthetic of the assembled model, we find that the number of cuts has a stronger impact: it begs for a precise matching and gluing. If one matches two parts with eyes, matching error and deformation on the resulting assembly is unguaranteed. The larger the number of parts, the more efforts it requires to assemble a model in a nice-looking manner. In addition, a cut of small peripheral length may not be easy to glue if its shape is very complicated; further, it is also very fragile to the breaking forces. Therefore, we put the number of parts in a higher priority than the total cutting length. To summarize, our objectives are the minimization of: (1) the number of partitions N of the mesh model; (2) the total peripheral length ΣL_i, where L_i is the peripheral length of cut c_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) As the directed arcs of H_i are translated to a common origin, they form a fork (see Fig. 5). Let a and b be the pair of farthest arcs in the fork, i.e., the angle between a and b, denoted as α, is the largest. Let the central axis of the minimum cone that encloses the fork be denoted as $r(H_i)$, then $r(H_i)$ is collinear with $a+b$. Then the feasible directions of printing H_i without using any support is bounded by a solid cone centered at $r(H_i)$ with an apex angle of $2\theta - \alpha$. Therefore, let $b(H_i)$ be the base cut of the mesh part that corresponds to the root of H_i (by base cut we mean the cutting boundary which adheres to the printing platform when the part is being printed, see Fig. 2 (d)), then $b(H_i)$ is required to be orthogonal to some directed arc $\hat{e}$ in $cone(H_i)$.
- (iii) The base of a partitioned 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 $area(b(H_i))$ denote the area of the base cut 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 experimentally;
- Finally, (iv) each cut partitions a single subgraph.

We have the following optimization system:

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

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

$$b(H_i) \perp \hat{e},\ \textit{for some}\ \hat{e} \in cone(H_i), \quad (2)$$

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

$$c_i \cap S = c_i \cap H_i, i = 1, \ldots, m. \quad (4)$$

where all H_i-s constitute a partition of the original graph. Again, a direct exploration of all possible partitions over the graph G could quickly leads to exponential complexity. The key here is to quickly find potentially good partitions in a way that subsequent exploration of the graph is limited to those which leads to a smaller

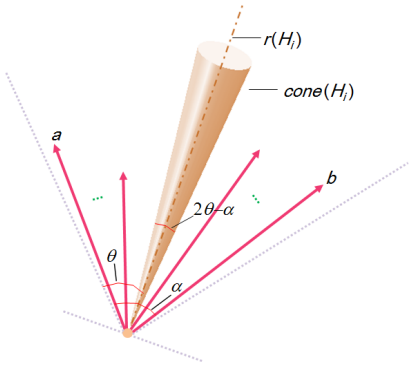


Fig. 5. Illustration of $r(H_i)$ and $cone(H_i)$. $Cone(H_i)$ encapsulates all valid directions w.r.t. all the arcs in H_i in order to make a support-free printing starting from the base.

value of the optimization function. We employ a stochastic method detailed below.

We separate the minimization of the two target terms sequentially, i.e., the number of mesh components first and then the total cutting length. In the following, we shall discuss how the skeleton and mesh are partitioned in details.

Skeleton Partition. To minimize the number of mesh components, we need to provide a proper set of partitioned subgraphs from S. For this purpose, we use a randomized exploration strategy. We give higher probabilities to exploring the graph broadly. In particular, assume that we are given a function $Trim\_BFS(v, G, \theta)$ which traverses G from v in a breath-first search manner to progressively collect arcs which satisfy constraint (i); and a function $MeshPartition(U, M)$ that cuts M based on the set of partitioned subgraphs U, and returns a partition of M, denoted as $M^{'}$, the number of cuts N and the total cutting length L. Algorithm 1 sketches the idea of the skeleton decomposition by minimizing N. The main idea is to randomly search for candidate subgraphs using the similar idea of a stochastic method, which randomly chooses a node of G to start traversing and randomly grows the subgraph while paying attention to the aforementioned constraints.

Next we shall show how $Trim\_BFS(v, G, \theta)$ works to find a locally maximal subgraph starting at v that satisfies the angle constraint. Let H be the current subgraph obtained so far, initially H is a single directed arc. 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 directed arcs form a fork (Fig. 5). A naive method for judging whether e should be included is to 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 maximum angle between e and each arc of the fork does not exceed 2θ. However, this would lead to $O(n^2)$ time complexity of $Trim\_BFS(v, G, \theta)$, where n is the size of G. To speed up this process, we apply a randomized algorithm (upon a random permutation of the sequence of insertions) that dynamically maintains the smallest disc to enclose all the inserted directed arcs (For a set of $|H|$ directed arcs, all the smallest enclosing discs can be computed in expected $O(|H|)$ time [57]).

Thereafter, as a new directed arc e is visited, we compute the acute angle between e and $Sum(H)/num(H)$, if it is no larger than θ, then it is safe to take it into H. By a backward analysis, each of such operations takes O(1) time in amortized sense.

In growing H, the order of the chosen arcs influences the shape

Algorithm 1 $Skeleton\_Mesh\_Decomposition(S, M)$

Input: The Laplacian skeleton S and mesh model M.
Output: The decomposition of M into the least number of pieces of components that are free of support.

$M_{opt} = \emptyset$; $min\_N = \inf$; $min\_L = \inf$; $count$ = 0; max_iter = a user defined large constant;
while $count < max\_iter$ **do**
  $U = \emptyset$;
  $X = S$;
  while $X \neq \emptyset$ **do**
    pick a random node $v \in X$
    $H = Trim\_BFS^*(v, X, \theta)$;
    $X = X/H$;
    $U = U \cup H$;
  end while
  $(M^{'}, N, L) = MeshPartition(U, M)$;
  if $N < min\_N$ **then**
    $min\_N = N$;
    $min\_L = L$;
    $M_{opt} = M^{'}$;
  else if $N == min\_N$ **and** $L < min\_L$ **then**
    $min\_L = L$;
    $M_{opt} = M^{'}$;
  end if
  $count = count + 1$;
end while
return M_{opt};

of the final subgraph. 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 introduce probabilities of choosing each arc of S. We denote this algorithm as $Trim\_BFS^*$, the main idea of $Trim\_BFS^*$ is as follows. We apply a procedure of choosing the arcs by learning history record from the first k (say 1000) times. 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 times, when a node v is visited, the probability of choosing an arc e as exit in the subsequent times is, $P(v, e) = n_v/k$.

Further, as $Trim\_BFS^*$ is greedy, 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 of achieving a global optimum, we set a randomized scheme to terminate the growing of a subgraph. The probability to terminate the growing is set to r/n, where r is the size of the current subgraph and n is the size of G. This gives high probability for $Trim\_BFS^*$ to grow in larger sizes while rising the probability of exploring subgraphs of smaller sizes. We implemented it using rejection sampling.

Mesh Partition. In the following section, we shall describe how $MeshPartition(U, M)$ works when a cut is required around a skeleton node v. Here the pseudocode is omitted for brevity. The skeleton partition returns a set of nodes where the mesh partition should occur. In particular, the cutting plane should be in the vicinity of each node v incident to at least two distinct subgraphs. We need to determine the exact positions and orientations of the cutting planes. Here we use a planar cut instead of geodesic cuts which are often used to segment a mesh part since planar cuts are simple and much easier to glue when 3D printed.

During the Laplacian smoothing process, a set of points on the mesh (which typically form a cylindrical shape whose central axis lies on a skeleton arc) collapse into a node on the skeleton. To efficiently cut the mesh, for each node v on the skeleton, we associate with it the set of points on the mesh that collapse into it. To preserve the geometric features as well as the aesthetic of the final assembled model, we shall cut the mesh through a concave point from this set. The detailed procedure is presented in the remainder of the section.

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 scheme. Refer to Fig. 6, at the position of node v, we want to find a surface vertex $p \in M$ around v through which the cutting plane should pass, while the cutting length is minimized. Let us denote the set of all planes which pass through a surface vertex p and are orthogonal to directed arcs in $cone(H_i)$ (originated from p) as $F(H_i)$, which we term as a *fillet*. 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 p, respectively. See Fig. 7, depending on the position of p, we have the following two cases.

Case i: $F(H_i) \cap F(H_j) \neq \emptyset$. See Fig. 7 (a), in this case, we shall sample a set of cutting planes from $F(H_i) \cap F(H_j)$ by using a uniform partition of the angle space, and determine the one achieving the minimum cutting length.

Case ii: $F(H_i) \cap F(H_j) = \emptyset$. See Fig. 7 (b), in this case, two cuts c_1 and c_2 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 $A(c_1, c_2) \leq \pi/2+\theta$. If this constraint is violated, putting one side (e.g., c_1) on the ground will cause the other side (c_2) to become an overhang that requires support from below. See Fig. 7 (c), in this case, one more cut in between c_1 and c_2 (e.g., c_3 along the angular bisector of c_1 and c_2) is required under the constraint of support-free fabrication. Meanwhile, the base area of each partitioned component should be no less than τ. If either of the constraint is not satisfied, we shall translate the fillets along the printing direction in an opposite sense until the constraints are satisfied. Particularly, the translation is done in a conservative manner: the translation first stops at a point where the two fillets are tangent at a point or the base areas of both fillets are larger than the threshold value. In this process, this small fixing may result in a violation of the constraints. To mitigate this, we may allow a relaxation of the constraint. Particularly, around

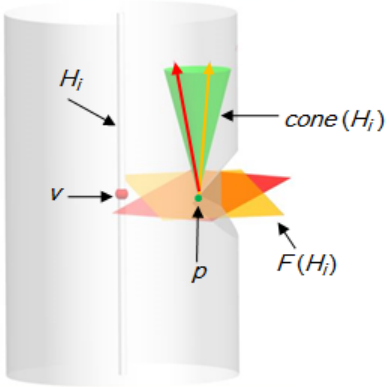


Fig. 6. Illustration of a cone and its corresponding *fillets*, where two cutting planes in $F(H_i)$ and their associated directed arcs in $cone(H_i)$ are marked by distinct colors.

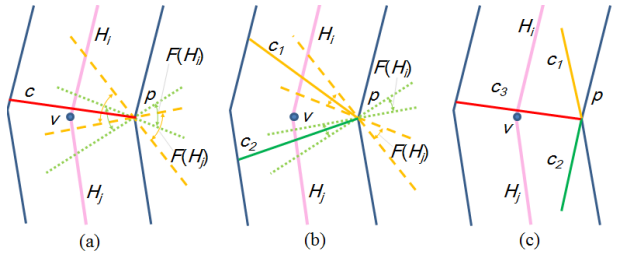


Fig. 7. 2D illustration of partition the mesh for different scenarios of $F(H_i)$ and $F(H_j)$. (a)$F(H_i) \cap F(H_j) \neq \emptyset$, a cut c that realizes the minimum peripheral length is induced; (b)$F(H_i) \cap F(H_j) = \emptyset$, two cuts c_1 and c_2 are required; (c)a cut c_3 along the angular bisector of c_1 and c_2 is induced when $A(c_1, c_2) > \pi/2 + \theta$.

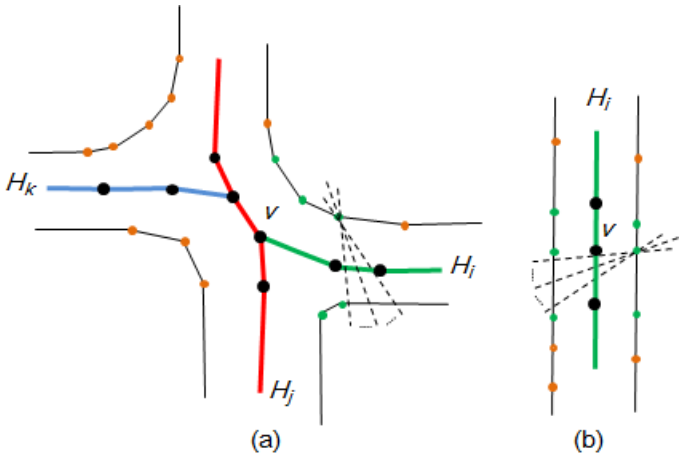


Fig. 8. 2D illustration of $R(v)$ and the cuts, all vertices in $R(v)$ are shown in green, and the cuts associated with a vertex are shown in dashed lines. (a) The case of cuts around concave points; (b) the case of cuts on a cylindrical part.

a cutting region, the choice of a single cut that results in the least amount of support material for both separate components is a nice relaxation, and this can save additional cuts.

In either case, a cut that nicely follows the geometry features is demanded in order to preserve aesthetic appearance in the final assembled object. We exploit shape concavity to look for a good cut as indicated by the minima rule [37], [38]. As the skeleton nodes may not locally reflect concavity, 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. Let $R(v)$ be the set of concave vertices on M that are incident to v during the Laplacian shrinking process [43], we truncate $R(v)$ such that the insignificant 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, denoted as τ_i, can be quantified as the magnitude of $(v_i - v_j)(n_j - n_i)$ [58]. We collect vertices whose τ_i is greater than a threshold δ. Next, we proceed to find a cutting plane around v. We first extend $R(v)$ by merging each adjacent $R(u)$ into it, where u is an adjacent node of v in S. Given all concave vertices in the new $R(v)$, each vertex defines a set of feasible cutting planes in accordance to its $fillet$. By feasible we require that a cutting plane does not cut through any other subgraph except for H_i. This is to avoid the scenario that a cutting plane cuts through the mesh component of another

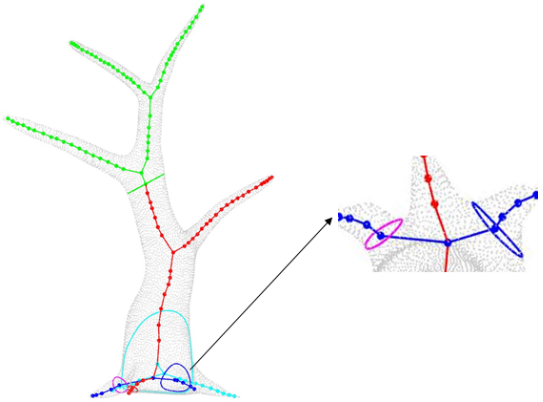


Fig. 9. The Tree model is partitioned into 6 parts according to the 4 subgraphs obtained by the super computer. Two cuts (the blue and purple circles) are required for the cross of the red and blue subgraphs.

subgraph, since each subgraph corresponds to a distinct mesh component. However, in practice this may not be feasible if no cut can avoid cutting into another subgraph and its corresponding mesh component. In this case, a relaxation on this constraint is allowed. We then exhaustively go through all feasible cutting planes and find the one whose cutting length is minimal. In case of a cylindrical part that does not merit good concaveness, we reduce the threshold value δ by half and repeat the procedure until a feasible cutting plane with minimum length is found. Fig. 8 illustrates the process.

In order to determine whether a cutting plane cuts 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 [43], as a cut goes through the mesh surface, the endpoints of the edges of M that are intersected give us the information of the potential subgraph that is 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. Note that it is possible that not all the constraints are met while searching for a solution for the objective, therefore our solution is based on a relaxation of the constraints: if a constraint is not met in a case, then it is relaxed, as little as possible. Fig. 2 shows an example of a deer model (shell), our partition result based on Laplacian skeleton of the model, the printing result and the assembly effect.

Occasionally, there are cases where two subgraphs intersect at some interior node, then their corresponding mesh components need to be separated by an additional cut. See Fig. 9 for an example. For the cross of the red and blue subgraphs, two cuts are required to separate their corresponding mesh components. The additional cut required is equivalent to using one more partitioning vertex hence changing the number of components of the resulting shape.

Note that a cut on the shell mesh may result in two annuluses each of which appears as a new facet to the mesh, for a facet c on a mesh component M_i with base b, c might need some support if its outward pointing normal direction subtends an angle of greater than $\pi/2 + \theta$. In the following, we shall show that this is impossible by the following lemma.

***Lemma**: Given a shell mesh component M_i corresponding to a subgraph H_i returned by Algorithm $Trim\_BFS^*$, each facet c of M_i that results from a non-base cut is free of support for 3D printing.*

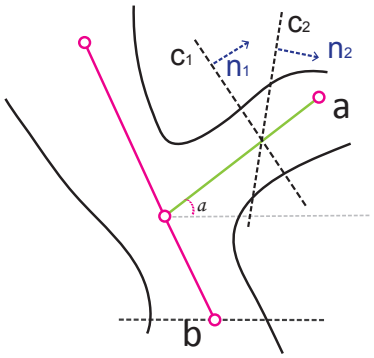


Fig. 10. Proof of Lemma: The colored lines indicate the skeletons, the dashed lines indicate the cuts.

Proof: See Fig. 10 for a 2D illustration, let b the base of M_i that is set on the ground, then the printing direction is vertical. We shall prove the correctness of the lemma by the construction of M_i: each arc of H_i subtends to the horizon by an angle of no smaller than $\pi/2 - \theta$. Let c be a facet (an annulus) induced by a cut on M_i that has an outward pointing normal $n(c)$, if $n(c)$ points upwards, then c is supported by the solid material of the shell of M_i beneath it (e.g., cut c_1 in Fig. 10); on the other hand, $n(c)$ points downwards. In the latter case, let a be the arc of the skeleton that c cuts through, a subtends to the horizon by an angle of $\alpha \geq \pi/2 - \theta$ by Algorithm $Trim\_BFS^*$. If c is parallel to a, then c cannot cut through a; If c subtends to the horizon by an angle of larger than α, then it is greater than $\pi/2 - \theta$, which means that c is also support-free (e.g., cut c_2 in Fig. 10); Finally, if c subtends to the horizon by an angle of smaller than α, it also cuts through a, but it results in a facet whose outward pointing normal is directed upwards (e.g., cut c_1 in Fig. 10) and the fact is again supported by the solid materials below. This is a contradiction and completes the proof. □

5 Results

For a watertight shell model, both its inner and outer boundary need to be supported by a huge amount of materials in order to guarantee a fine surface quality. See Fig. 11 (a-b) for an illustration of the Sculpture model, both its inner and outer surface require a significant amount of support in order not to be deformed during the printing process; while our approach only keeps all cylindrical shells that are free of support (Fig. 11 (c-d)).

We have run our algorithm on various natural and man-made models, and some of the results are presented in Fig. 13. The left column of the figure exhibits the most material-saving orientation for printing the models using the Meshmixer software, a free software provided by Autodesk company. We validated our approach by a set of printing experiments on a 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 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* that can automatically count the filament of the print material (in meters) and an estimate of the weight of the print material. Table 1 summarizes the printing material and time that

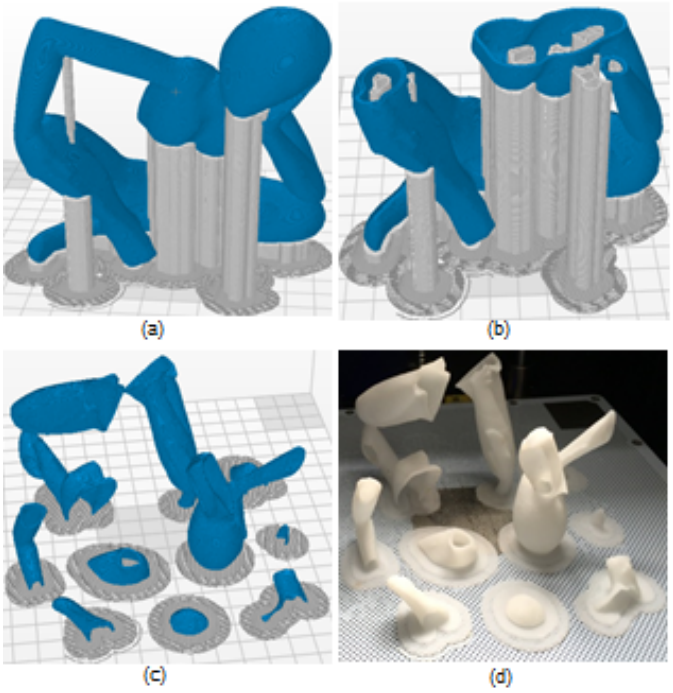


Fig. 11. An illustration of the Sculpture model under the 3D printing software Z-suite as $\theta = 20°$: (a) the full model; (b) an intermediate step of the simulation; (c) the simulation result of our partition for the Sculpture model; (d) the 3D printed result of our partition for the Sculpture model, no support is required except for the bed (limited by FDM technique).

are consumed by the original models and our 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.

The algorithm was implemented with C++ on a PC with Intel i7-4790 and 8 GB RAM. The skeleton partition algorithm was run on each model for 8000 times with the first 1000 times taken as a learning process, i.e., the arcs are chosen by learning the record of the first 1000 times. The running time and the number of vertices of the models are summarized in Table 2. The running time of the algorithm depends on the number of iterations, the number of mesh vertices and the number of skeleton arcs, the topology of the skeletons, the seed nodes of the subgraph used in each iteration, and the positions of the vertices that induce mesh partition. Among these factors, in addition to the numbers, the topology of the skeletons matters a lot: a structure with loops or without loops (i.e., a tree) make a significant difference. As the number of mesh vertices increases, the number of skeleton arcs may not change since a small skeleton arc can represent a large number of vertices, which means that the running time may not change that much. However, even if the number of skeleton arcs increases, the running time still depends on other factors such as the seed nodes and the terminal nodes chosen in each iteration. In particular, if the nodes are chosen properly within the first few iterations, then the algorithm can stop within a short time.

Fig. 13 shows the comparison of the printing effects of the original models and our partitioned models. Due to the limit of the current technology, any FDM printer requires a small amount of 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 beds. Therefore, our approach guarantees support-free to the most extent for all existing printing techniques.

Since the partition of S may have an exponential amount of choices, it is impossible to obtain an optimal mesh partition in polynomial time for complicated models. Particularly, which arcs to be taken into a subgraph and the taking sequence significantly affect the partition result. Thus, it is hard even for humans to determine the optimum solution. In solving such an intricate problem involving various parameters, a stochastic method with a large number of iterations is a good choice. To guarantee that the method can converge to a nice result within limited number of iterations, we apply a learning procedure from history data for partitioning the skeletons, which helps accelerate the searching process. From Table 1, we can conclude that the partition numbers for the skeletons and mesh models are sufficiently small. Further, by running our proposed polynomial-time algorithm for special cases (i.e., the number of partitions and the degree for each node are bounded by a small constant), we can evaluate our skeleton partition result by examining all the above mentioned models on a super computer called π, which has $257,000$ CPU cores and is owned by Shanghai Jiao Tong University. Table 3 summarizes the result of the super computer. Particularly, the result NA means "not applicable" if no result can be obtained within the budget running time. More precisely, for the Armadillo model, only $1/5836917$ of the total combinations are tested; and for the Knots model, after 36 hours, only $1/122437$ of the total combinations are tested. Given a skeleton, the number of CPU cores is chosen based on the principle that the number of cores should be smaller than the number of arcs in the skeleton; and that the super computer assigns an integer number of nodes to the user, where a node contains 16 CPU cores. The experimental results are summarized in Table 3.

Comparing Table 3 with Table 1, we can see that the skeleton partition results of Armadillo and Knots cannot be improved within a long-lasting running time; the results of Octopus, Deer, Sculpture, Airplane, Gargoyle and Bearing in Table 1 are optimal already and cannot be improved further; the result of Tree in Table 1 can be improved by 2. Therefore, we can conclude that our proposed stochastic method with the help of choosing arcs by learning history record can guarantee optimistic results within a short time.

6 Conclusion, Limitations and Future Work

In this paper, we present a skeleton-based approach for partitioning a shell model into parts which are free of support structure when fabricated. We prove the NP-hardness of the problem. We formulate the model partition problem as a constrained graph partition problem which is particularly tailored toward fabrication. To tackle the NP-hardness of the problem, we exploit a stochastic method which adaptively searches for better partition results while avoiding local minima. We also propose a polynomial time algorithm for the skeleton partition problem. 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.

Models	Skeleton generation time	Print material (original)	Print time (original)	Number of parts (skeleton partition)	Number of parts (mesh partition)	Print material (partition)	Print time (partition)	Material save (%)	Time save(%)
Tree	15.37s	8.07m (19g)	6h 9min	6	8	5.19m(12g)	4h 2min	35.6877	34.4173
Armadillo	82.54s	7.7m (18g)	5h 15min	9	10	4.65m(11g)	3h 30min	39.6104	33.3333
Octopus	6.72s	14.31m (34g)	10h 56min	5	9	8.75m(21g)	6h 51min	38.8539	37.3476
Deer	11.42s	10.52m (25g)	8h 54min	4	6	8.89m(21g)	5h 25min	15.4943	39.1386
Airplane	25.23s	6.58m (16g)	5h 42min	5	7	5.24m(12g)	3h 10min	20.3647	44.4444
Knots	3.46s	21.45m (51g)	16h 41min	10	15	12.38m(29g)	11h 20min	42.2844	32.0679
Sculpture	6.28s	21.85m (52g)	18h 41min	4	10	16.35m(39g)	11h 23min	25.1716	39.0723
Gargoyle	55.34s	6.75m (16g)	5h 18min	4	5	5.16m(12g)	3h 48min	23.5556	28.3019
Bearing	55.34s	8.52m20g	6h 33min	10	10	7.36m(18g)	4h 42min	13.62	28.24

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

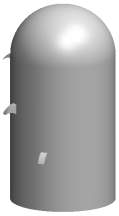
Models	Tree	Armadillo	Octopus	Deer	Airplane	Knots	Sculpture	Gargoyle	Bearing
number of vertices	11443	34594	1343	8917	15485	2904	5979	25002	44945
number of skeleton arcs	130	143	81	77	60	77	56	50	44
time	187.2576s	346.374s	102.1403s	93.6414s	91.8654s	95.8742s	76.8461s	202.7632s	173.4283

TABLE 2
The running time of processing the models with our stochastic algorithm.

Models	Tree	Armadillo	Octopus	Deer	Airplane	Knots	Sculpture	Gargoyle	Bearing
optimal number	4	NA	5	4	5	NA	4	4	10
number of CPU cores	128	144	80	48	48	80	16	32	32
time(hours)	0.49	36.27	1.36	0.05263	0.06824	36.01	0.24	0.0793	4.2746

TABLE 3
The result of our polynomial-time algorithm for skeleton partition when the number of partitions and the degree for each node are bounded by a small constant. The algorithm is run on a super computer.

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



Limitations and Future Work. Although our approach is devoted to shell models, it can also be applied to cutting solid models without any problem. As we assembled the parts, we realized that a pair of parts may not be matched correctly of their interfaces are perfect circles since the two parts can be rotated while keeping the interfaces touching perfectly. We remark that this issue can be addressed by a small trick as follows: on each interface, protruding a planar tinny spike from the inner boundary of the shell such that its pinpoint pieces into the interior of the void zone. In this way, the relative position of a pair of interfaces can be determined by matching the spikes. 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. In this work, we restrict our focus to a uniform setting of the shell thickness whose value is determined by an error-and-trial process. A future research would be to determine the minimum shell thickness in different parts of a given model, this process requires an efficient detection of self-intersection and an experimental study of the printability according to the curvature changes of the shell model.
- 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 cut number as well as the minimal cut length. A potential future research is to take care of salience region during graph partition; particularly, as a trade-off between support material and salience preservation, a non-planar cut might be a consideration to alleviate the salience problem.
- For a model with tiny spikes that result in local minimum points, a skeleton may not be able to capture the geometric feature of these spikes, which could beg for a support below each local minimum point. Refer to the inset figure for an illustration.
- Laplacian skeleton is not the unique curve skeleton suited for describing the topology of the models, any curve skeleton that captures the geometric features of the model can be used as well. A comparison of the performance of various curve skeletons is worthy of future research. Furthermore, a strict measurement of how much detail the skeleton needs to capture in order to find the optimal global partition solution is challenging in our current setting, as it amounts to solve another NP-hard searching problem.
- The end of a skeleton branches (node of the skeleton of valence 1) is not handled properly when its corresponding geometry feature (spherical cap for instance) is used as base. For example, the octopus or the gargoyle head is not handled

properly.

- The proposed method works well if each locally tubular shape has constant radius along its corresponding skeleton, which provides guarantee on the absence of supports.
- Although the final segmentation can always be assembled, different assembling sequences can lead to different degrees of shape error; even if the assembling sequence is fixed, how two components are matched at their interfaces each time also influences the shape error. Therefore, a matching strategy which leads to the least gluing effort or finest appearance is worthy of future exploration.

ACKNOWLEDGMENTS

This work was supported in part by Shanghai Sailing Program No.16YF1405500, Shanghai Jiao Tong University Grant No. AF0200163, State Key Laboratory of Mechanical Systems and Vibration Grant No. MSVZD201505, and The National Natural Science Foundation of China No. 51605290, No. 61502306, and the China Young 1000 Talents Program.

REFERENCES

[1] M. Livesu, S. Ellero, J. Martínez, S. Lefebvre, and M. Attene, "From 3d models to 3d prints: an overview of the processing pipeline," in *Computer Graphics Forum*, vol. 36, no. 2. Wiley Online Library, 2017, pp. 537–564.

[2] 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.

[3] 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.

[4] 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.

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

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] A. Tagliasacchi, H. Zhang, and D. Cohen-Or, "Curve skeleton extraction from incomplete point cloud," in *ACM Transactions on Graphics (TOG)*, vol. 28, no. 3. ACM, 2009, p. 71.

[11] 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.

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

[13] 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.

[14] 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.

[15] 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 Transactions on Graphics (TOG)*, vol. 32, no. 6, p. 177, 2013.

[16] 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.

[17] S. Hornus, S. Lefebvre, J. Dumas, and F. Claux, "Tight printable enclosures for additive manufacturing," Ph.D. dissertation, Inria, 2015.

[18] Y. Xie and X. Chen, "Support-free interior carving for 3d printing," *Visual Informatics*, vol. 1, no. 1, pp. 9–15, 2017.

[19] A. Telea and A. Jalba, "Voxel-based assessment of printability of 3d shapes," in *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*. Springer, 2011, pp. 393–404.

[20] W. M. Wang, C. Zanni, and L. Kobbelt, "Improved surface quality in 3d printing by optimizing the printing direction," in *Computer Graphics Forum*, vol. 35, no. 2. Wiley Online Library, 2016, pp. 59–70.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] O. V. 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.

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

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

[31] 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.

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

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

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

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

[36] 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.

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

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

[39] 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.

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

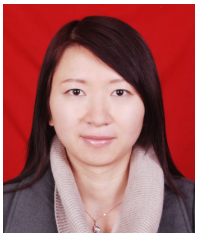
[41] J.-M. Lien, J. Keyser, and N. M. Amato, "Simultaneous shape decomposition and skeletonization," in *Proceedings of the 2006 ACM symposium on Solid and physical modeling*. ACM, 2006, pp. 219–228.

[42] 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.

[43] 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.

[44] 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.
[45] 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.
[46] M. Attene, "Shapes in a box: Disassembling 3d objects for efficient packing and fabrication," in *Computer Graphics Forum*, vol. 34, no. 8. Wiley Online Library, 2015, pp. 64–76.
[47] M. Yao, Z. Chen, L. Luo, R. Wang, and H. Wang, "Level-set-based partitioning and packing optimization of a printable model," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 6, p. 214, 2015.
[48] P. Song, B. Deng, Z. Wang, Z. Dong, W. Li, C.-W. Fu, and L. Liu, "Cofifab: coarse-to-fine fabrication of large 3d objects," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, p. 45, 2016.
[49] 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.
[50] P. Song, C. Fu, and D. Cohen-Or, "Recursive interlocking puzzles," *ACM Trans. Graph.*, vol. 31, no. 6, p. 128, 2012.
[51] 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.
[52] 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.
[53] Y. Shinagawa, T. L. Kunii, and Y. L. Kergosien, "Surface coding based on morse theory," *IEEE Computer Graphics and Applications*, vol. 11, no. 5, pp. 66–78, 1991.
[54] M. Rumpf and A. Telea, "A continuous skeletonization method based on level sets," in *Proceedings of the symposium on Data Visualisation 2002*. Eurographics Association, 2002, pp. 151–ff.
[55] A. Tagliasacchi, T. Delame, M. Spagnuolo, N. Amenta, and A. Telea, "3d skeletons: A state-of-the-art report," in *Computer Graphics Forum*, vol. 35, no. 2. Wiley Online Library, 2016, pp. 573–597.
[56] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of computer computations*. Springer, 1972, pp. 85–103.
[57] M. De Berg, M. Van Kreveld, M. Overmars, and O. C. Schwarzkopf, "Computational geometry," in *Computational geometry*. Springer, 2000, pp. 1–17.
[58] 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.



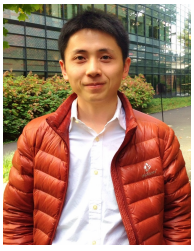
Siqi Qiu is currently an Assistant Professor with the Institute of Intelligent Manufacturing and Information Engineering, School of Mechanical Engineering, Shanghai Jiao Tong University. She received her PhD degree from the Department of Computer Engineering, Compiegne University of Technology, her MSc 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.

Ruiliang Feng is a PhD Candidate with the Institute of Intelligent Manufacturing and Information Engineering, School of Mechanical Engineering, Shanghai Jiao Tong University. He received his ME degree in Mechanical Engineering from from the China Coal Research Institute; and BE degree in Mechanical Engineering from China University of Mining and Technology. His current research interest is Additive Manufacturing.

Yaobin Tian is a Post-doctoral fellow with the Department of Industrial Engineering and Logistics Management of Hong Kong University of Science and Technology. He received his PHD and Master degree in School of Mechanical, Electronic and Control Engineering from Beijing Jiao Tong University. His current research interests include Additive Manufacturing, Robotics and Mechanism Design.

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 a Researcher (PI) at the State Key Lab of CAD&CG, College of Computer Science, Zhejiang 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.

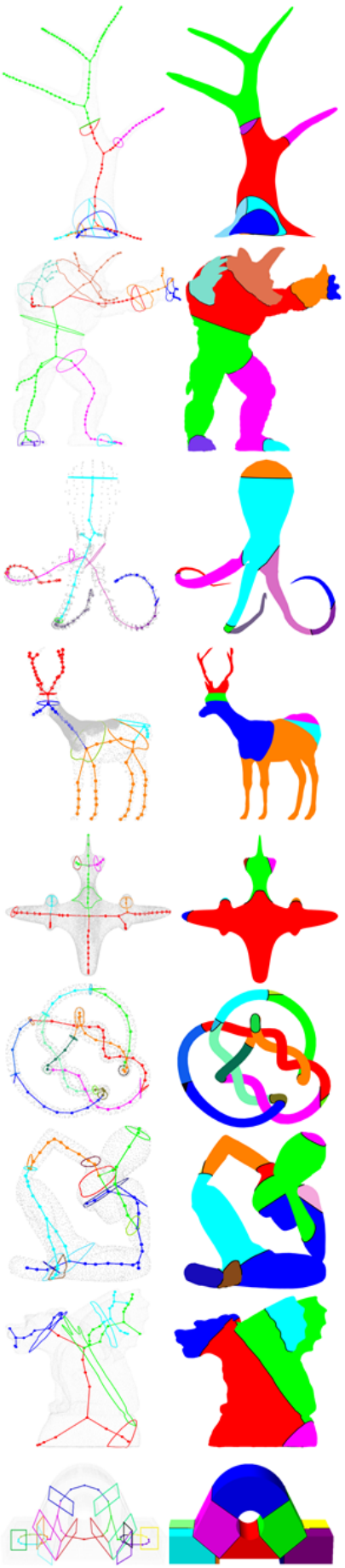


Fig. 12. Partition examples as $\theta = 70°$. The printing direction of each part is orthogonal to its base (shown in the same color as the part)

Fig. 13. A comparison between printing of original shell models and our partitioned models. The orientations of the models in the left column are determined as the optimal printing directions that correspond to the minimum amount of support materials (by using Meshmixer software provided by Autodesk).