



## Surface Attributes Driven Volume Segmentation for 3D-Printing

Xin Liu<sup>a</sup>, Chuhua Xian<sup>a,\*</sup>, Shuo Jin<sup>b</sup>, Guiqing Li<sup>a</sup>

<sup>a</sup>*School of Computer Science and Engineering, South China University of Technology, Guangzhou, 510006, China*

<sup>b</sup>*BlueFire AI, Hong Kong SAR, China*

### ARTICLE INFO

#### Article history:

Received August 16, 2021

**Keywords:** Multi-material Fabrication, Volume Segmentation, 3D-Printing

### ABSTRACT

Volume segmentation based on surface attributes is an essential problem in multi-material fabrication and model packing. In practice, current mainstream fabrication techniques have difficulties in yielding models with diverse surface attributes in one pass owing to their craft limitations, making model segmentation a sensible choice for model realization. Partitioning 3D objects into single-attribute volumetric parts prevents fabricating models with different material in a single printing procedure, whereas the arisen challenge is to determine a reliable segmentation solution that is able to handle complicated models in various use scenarios. To achieve this goal, we propose a novel volume partition algorithm generating feasible volumetric parts, each of which is affiliated with one single surface attribute. Our technique enables model segmentation with least conflict and constrained wall thickness so that each volumetric segment can be realized independently by 3D-printing. Generally, it starts with computing a partition proposal guided by radial-based-function iso-surface, then optimizes segmentation interface with a prescribed minimal printing thickness to produce high-quality surface for every volumetric part, and finally splits unextractable volumetric parts into smaller sub-volumes to ensure assemblability of the whole model. As previous methods do not work well in optimizing segment interface for printing, we propose a differential evolution based smoothing algorithm to generate smooth and continuous interface, declining the risk of collision between adjacent volumetric parts. Extensive experimental results are provided in this paper to demonstrate the effectiveness and quality of our proposed technique, showing its advantages on model manufacture over prior methods.

© 2021 Elsevier B.V. All rights reserved.

### 1. Introduction

With the recent development of additive manufacturing techniques, digital fabrication has been widely used in various areas such as industrial manufacturing and biomedical engineering, which takes in digital model files and prints out physical objects with plastic, powdery metal, resin or other materials. It is often demanded to satisfy various requirements on the fabricated models based on specific applications. For example, a

typical use case is that different materials or colors are desired on different parts of a given model to achieve certain physical feature or colored appearance (Fig. 1(a)). However, it is technically challenging for the current commonly used 3D-printers to fabricate multi-attribute object as a single solid while retaining its prescribed attributes of each model part. The alternative path to this problem is to decompose the input model into smaller volumetric segments which can be manufactured separately and assembled easily. Specifically, given a surface model with required attributes defined on its different surface regions, model decomposition is expected to split the whole model into assemblable volumetric segments, each of which is associated

\*Corresponding author.

*e-mail:* [chhxian@scut.edu.cn](mailto:chhxian@scut.edu.cn) (Chuhua Xian)



Fig. 1. An example of volume segmentation based on surface attributes. (a) The surface of the model with color attributes, and its segmented volumetric parts using our method. (b) The fabricated result and its assemblable parts.

with a single attribute and can be manufactured easily using today's mainstream 3D-printing techniques.

Previous work [1, 2] on volume segmentation directly divides a volume by well-computed dividing planes in regardless of the contours of surface patches, which is friendly for model assembling. However, it is not intuitive for these methods to handle non-planar patch contour, resulting in unsatisfaction of required attributes in 3D-printing. To process input model with predefined attributes, Yao et al. [3] proposes a contour sweeping algorithm dividing colored model into volumetric parts accurately coincided with surface region boundary, while it can only be applied to furniture models. Surface2Volume [4] partitions furniture and free-form models with graph cut by minimizing a specific energy function designed to enforce assemblability of its volumetric segments. A major defect for Surface2Volume [4] algorithm is that the lack of thickness guarantee applied on segmented parts increases the manufacture risk in 3D-printing. Moreover, its resulting parts with rough interface increases the printing time cost as shown in Fig. 2.

Our proposed technique defines a three-step workflow for generating smooth model partition based on pre-defined attributes on different surface regions, which can be assembled sequentially to realize a complete model fabrication. Rather than operating on all the elements inside a volume object, our approach computes boundary interface to divide volume into valid volumetric parts strictly conforming to the exposed surface attributes. First of all, we discretize the input surface from a 2D-manifold surface to 3D volumetric model filled with tetrahedrons. Second, our method predicts demarcation interfaces matching each group of connected boundary vertices between two adjacent patches as we regard the partition problem as a hole-filling problem. Then we apply a smoothing algorithm on these interfaces with the constraint of guaranteeing minimal printing thickness. Finally, unextractable volumetric parts are identified and further sub-divided to ensure part extractability. Fig. 1 shows an example of our volume segmentation based on its surface attributes. The main contributions of our work are summarized as follows:

- An efficient radial-basis-function (RBF) guided segmentation method is proposed to generate smooth and contiguous model partition based on surface attributes, which serves as the initial segmentation result.
- A differential evolution based smoothing algorithm is introduced to optimize the cut interfaces of segmented parts

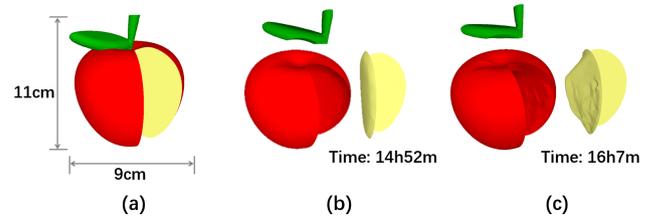


Fig. 2. Comparison of model printing time using our approach and Surface2Volume [4] by FDM printer. (a) The input model with color attributes. (b) Segmentation result using our method and its corresponding printing time. (c) Segmentation result using Surface2Volume [4] and its corresponding printing time.

constrained by the prescribed minimal printing thickness.

- The proposed technique is applied on a variety of models, and the experimental results shows its advantages over prior methods. Additionally, printed physical objects are provided as a proof of its reliability on current mainstream printers.

## 2. Related work

In literature, there are a large amount of techniques tackling the problem of model segmentation. This review does not aim for completeness, but provides an overview of the scope of techniques that our work relates to the most.

**Shape Segmentation.** Shape segmentation approaches target at partitioning input model guided by certain objectives, which can be categorized into shape-oriented methods and surface-oriented methods. Traditional shape-oriented segmentation in digital fabrication processes input object with the exploitation of geometry structure and part interrelation, and the optimization objective mainly concentrates on two aspects: box packing and interlock generation. Previous work [1, 5, 6, 7, 8] aim at packing volumetric parts with least space usage after dividing the model. Chen et al. [1] seek for a globally optimal solution to saving package space based on the well-designed objective function and top-down and iterative searching. Ho et al. [5] decompose a polygonal object into meaningful parts using Minimum Slice Perimeter (MSP) function. Vanek et al. [6] divide model guided by the size of the connected areas or volume of each packed segment to minimize required support material and the bounding box volume of the segments. Attene [7] proposes a method splitting input model in parts that can be efficiently packed within a box, which assists in creating a hierarchy of possible parts and the objective criterion of re-assembling. Strodthoff [8] divides the model by generating volumetric spline models for iso-geometric analysis. Unluckily, all these methods do not account for preserving surface attributes in their segmentation computation.

Another work thread [2, 9, 10] focuses on generating interlocking decomposition for part assemblability. Luo et al. [2] propose a framework to decompose an object into parts with customized connectors on the interfaces abiding by assemblability and other criteria. Fu et al. [9] present a computational

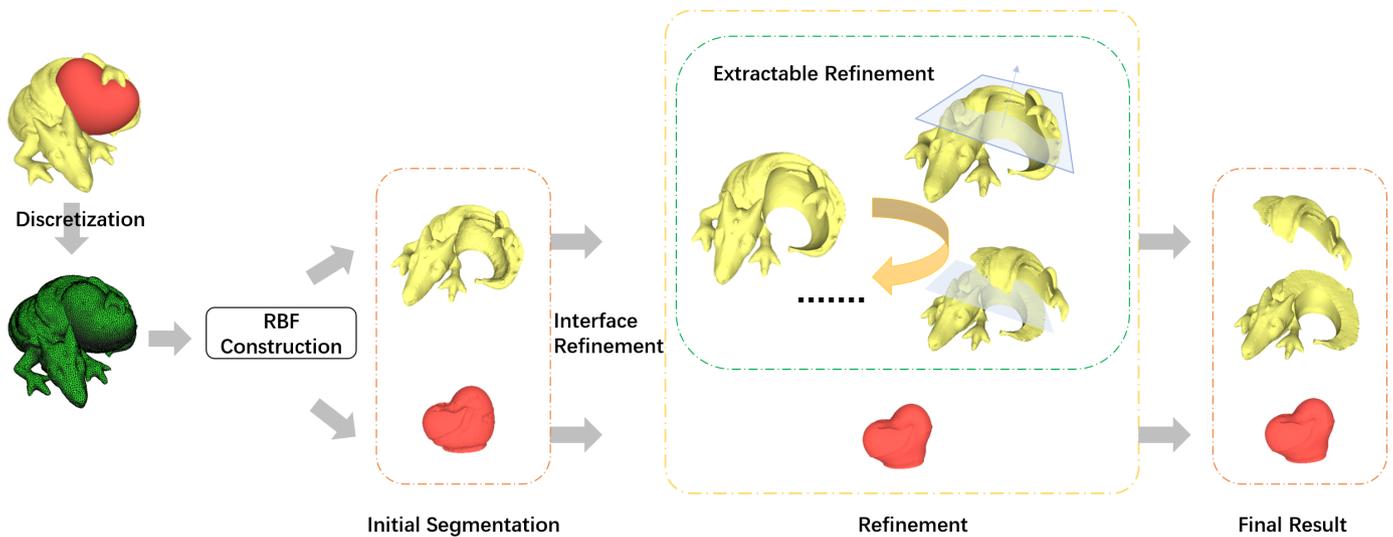


Fig. 3. Flowchart of our volumetric segmentation method driven by prescribed attributes on surface.

1 solution of forming a globally-interlocking furniture assembly by determining the interlocking relationship of different  
 2 parts. Song et al. [10] develop a voxelization-based approach  
 3 to partition input model into interlocking parts, which guaran-  
 4 tees the structural soundness and connection with high qual-  
 5 ity. These shape-oriented methods decomposing original input  
 6 model without considering surface information are not capable  
 7 of generating interface with given boundary contour, thus be-  
 8 come difficult to be directly migrated to handle input models  
 9 with prescribed attributes.

11 To handle the large objects, Song et al. [11] proposes a  
 12 coarse-to-fine fabrication solution, which combines 3D print-  
 13 ing and 2D laser cutting for cost-effective fabrication of large  
 14 objects at lower cost and higher speed. Jadoon et al. [12]  
 15 presents an interactive tool for partitioning a 3D model into  
 16 printable parts if the model is larger than 3D-printer’s work-  
 17 ing volume. In order to reduce the build material and the support  
 18 structure, Gao et al. [13] explore a multi-directional 3D printing  
 19 process. Their method can not only reduce the consumption of  
 20 print and support material, but also to enable a new breed of  
 21 custom products with embedded functionalities. In [14], Wu  
 22 et al. develop a general volume decomposition algorithm for  
 23 effectively reducing the area that needs supporting structures.  
 24 This method fabricates general models with multi-directional  
 25 3D printing systems by printing different model regions along  
 26 different directions, which can speed up the process of 3D print-  
 27 ing by saving time in producing and removing supports. How-  
 28 ever, the decomposition of these methods do not aim at process-  
 29 ing the model with multiple surface attributes.

31 Previous work on surface-oriented volume segmentation  
 32 aims at extracting assemblable parts from attributed surface  
 33 model. Yao et al. [3] proposes a sweeping algorithm that as-  
 34 signs elements lying inside the scope of surface contour sweep-  
 35 ing along the least conflict direction to the corresponding sur-  
 36 face region. The method is not suitable for surface with com-  
 37 plicated boundary, as it specifically focuses on generating joints  
 in furniture models. For free-form objects, Surface2Volume [4]

38 designs an effective energy function of graph cut dividing the  
 39 object volume into a subset of assemblable parts necessitat-  
 40 ing sequential assembly. They succeed in dealing with irregu-  
 41 lar surface segmentation meanwhile remaining extractability  
 42 on each part. However, this method do not consider the quality  
 43 of segmentation interface. Moreover, it costs significant amount  
 44 of time on large-scale models in the multi-cut computing pro-  
 45 cess.

46 **Implicit Surface Reconstruction.** Implicit surface reconstruc-  
 47 tion is the process of retrieving a manifold surface with im-  
 48 plicit function from unstructured input data [15], which can  
 49 be grouped into classic method [16, 17, 18] and variational  
 50 method [19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29]. Among the  
 51 methods mentioned above, implicit surface based on radial ba-  
 52 sis functions (RBF) [22, 30, 31, 32, 33, 34, 35] has been widely  
 53 used for surface reconstruction with contour that approximates  
 54 the target surface by constructing the signed distance function  
 55 with boundary points and interpolating new points in regions  
 56 with holes. Poisson surface reconstruction [18, 36] shows its  
 57 robustness and effectiveness for reconstruction with scattered  
 58 points by representing surface estimation as a Poisson problem.

59 In terms of efficiency and model features, RBF fits our seg-  
 60 mentation process the best. As most of our inputs are in free-  
 61 form shape, RBF learns characteristics of the given surface  
 62 without considering the complicated shape of exposed surface,  
 63 reducing the amount of elements to be considered and speeding  
 64 up the whole computational process. Since the shape of fixed  
 65 hole can be controlled by kernel function and control points, we  
 66 adjust the boundary condition in RBF to obtain required output  
 67 mesh in our paper.

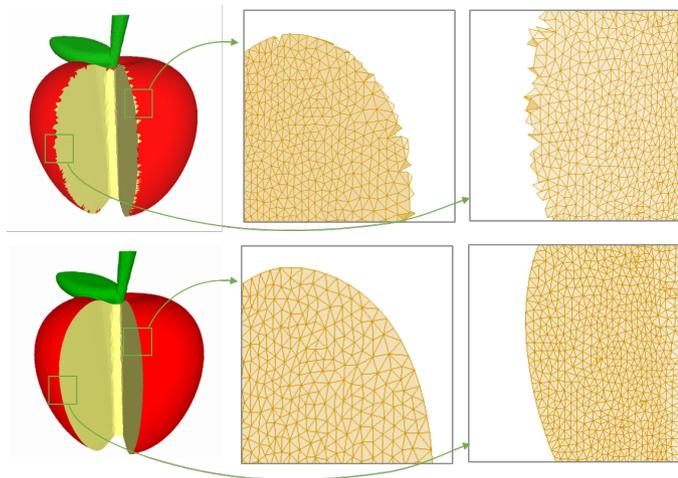
### 3. Methodology

68  
 69 As shown by the flowchart in Fig. 3, we develop a model par-  
 70 tition algorithm to divide input surface models with three main  
 71 steps, producing a set of segmented volumetric parts that can be  
 72 easily assembled for 3D-printing. Our method first discretizes

the closed input mesh surface with tetrahedrons, and then computes RBF iso-surface of each attributed part simultaneously to obtain an initial segmentation. Then, we employ a differential evolution based smoothing step to optimize the initial interfaces between adjacent parts. This smoothing process is constrained by a surface generated from outer surface offset inwards by a certain length to avoid fragile area, where the length relies on the minimal printing thickness in 3D printing. Finally, each volumetric part is embedded in a directed graph reflecting the interlocking relationship and checked whether it can be disassembled along with extracting direction. If it is unextractable, we iteratively divide it into smaller pieces until all parts become extractable by computing proper division planes. We will detail each step in the following sub-sections.

### 3.1. Space Discretization

Given a closed 3D surface model  $S$ , we first yield a uniformly sized tetrahedral mesh by applying classic Delaunay tetrahedralization on it. The original attribute on each surface triangle facet will be assigned to its corresponding surface tetrahedron, while the attributes of the inner tetrahedrons remain undetermined. In this work, we use the open-sourced C++ lib-tetgen [37] to conduct tetrahedralization on surface mesh model. Generally, the non-boundary surface tetrahedrons have unique attribute. However, the boundary surface tetrahedrons may have multiple attributes if it contains two faces corresponding to different attributes. Assigning this kind of tetrahedron to one of the concerning regions may cause sawtooth on the boundary area, as shown in the top row in Fig. 4. To eliminate such ambiguity, we detect and split these boundary tetrahedrons into smaller one at the edges across two surface regions, and each element is labeled with only one attribute. Bottom row in Fig. 4 shows the post-processed result of input model shown in the top row in Fig. 4 to address this issue on the region boundary.

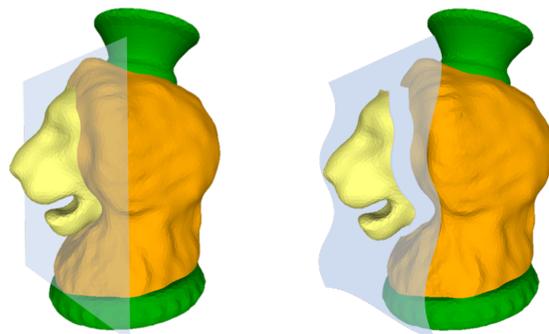


**Fig. 4.** Elimination of attribute ambiguity for boundary tetrahedrons. **Top:** original discretization result using tetgen software. **Bottom:** post-processed result by sub-dividing illegal tetrahedrons with multiple attributed faces. The left sub-figures are colorized triangle mesh model while the right sub-figures are the detailed view of local regions on the tetrahedral model.

### 3.2. Initial Volume Partition

The goal of volume partition is to segment the discretized volume model into feasible parts with respect to specified attributes on surface as well as extracting direction. Segmentation problem can be intuitively formulated as a graph partitioning problem with certain targets, where tetrahedrons are nodes and edges exist if two adjacent tetrahedrons sharing the same face. In our work, the goal of model partition is to force the interface between two connecting surface regions to be not only least conflicted for extraction but as planar as possible to facilitate subsequent processes like joint installation.

It assumes that a deformable 3D plane is to be produced to split two adjacent regions. Since the region boundary points do not usually lie on a plane perfectly, we need to deform the plane to coincide with the boundary points as shown in Fig. 5. The division surface generated with minimal distortion is our desired interface with least mean curvature and as-flat-as-possible shape, which naturally turns into the problem of computing a minimal surface with fixed boundary [38, 39].



**Fig. 5.** **Left:** division plane is not able to respect the non-planar boundary. **Right:** deformed division surface cuts the model along the region boundary.

According to the definition of minimal surface, one surface is minimal if and only if its mean curvature is equal to zero everywhere on it. Due to the uncertainty of boundary shape, it is not straightforward to calculate the minimal surface directly. Thus, we first generate an approximate surface with low curvature, which is used to iteratively approach least curvature. In terms of efficiency and model characteristics, radial-basis-function (RBF) [22] fits our segmentation target best. Firstly, implicit surface from RBF can coincide with all control points, namely the boundary points in our setting. Secondly, RBF outputs a continuous and fairing interpolation surface by minimizing the energy function

$$E = \int_{\mathbb{R}^3} f_{xx}^2(\mathbf{x}) + f_{yy}^2(\mathbf{x}) + f_{zz}^2(\mathbf{x}) + 2f_{xy}^2(\mathbf{x}) + 2f_{yz}^2(\mathbf{x}) + 2f_{zx}^2(\mathbf{x}), \quad (1)$$

which reflects the smoothness on generated surface. The resulting RBF interpolant of the implicit function can be written as

$$f(\mathbf{p}) = g(\mathbf{p}) + \sum_{i=1}^N \lambda_i \phi(\|\mathbf{p} - \mathbf{p}_i\|), \quad (2)$$

where  $g(\mathbf{p})$  is a polynomial of point  $\mathbf{p}_i$ ,  $\lambda_i$  are the coefficients of kernel function, and  $\|\cdot\|$  represents the Euclidean norm.

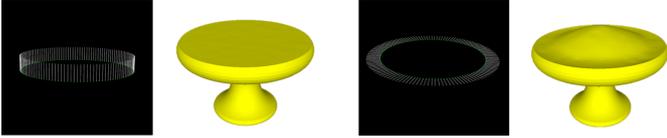


Fig. 6. Different reconstructed results with different normal sets and the same control points.

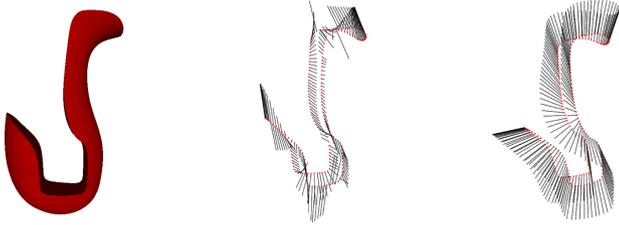


Fig. 7. Different normals for given boundary vertices. Left: original boundary vertices. Middle: computed normal using plane fitting. Right: computed normal using our linear system solution.

To reconstruct RBF surface from region boundary on the input model, boundary vertices and their corresponding normals are collected at the beginning. Given  $n$  distinct points  $P = \{(x_i, y_i, z_i)\}_{i=1}^n$  on a region boundary  $B$ , it is possible to fit a continuous implicit surface  $S'$  which is reasonably approximate to  $B$  coinciding at the boundary points  $\mathbf{p}_i \in P$  with proper radial basis function. Our approach is to model the approximate continuous surface implicitly with selected kernel function, by forcing all the points  $\mathbf{p}_i \in P$  lying on boundary  $B$  to satisfy the following equation:

$$f(\mathbf{p}_i) = 0. \quad (3)$$

To solve the RBF linear system, we need to add non-zero off-surface points to determine the vertex direction of implicit surface point. Regarded as non-zero off-surface points, geometric control points  $P' = \{\mathbf{p}'_i\}$  that are not on the boundary should be constrained by:

$$f(\mathbf{p}'_i) = d_i \neq 0. \quad (4)$$

Generally, the flatter the segmentation interfaces, the easier it is to extract the volumetric parts. Since the linear kernel function can generate a flat surface, we set  $\phi(r) = \|r\|$ . The key to RBF hole filling method is to determine the geometric control point set near the hole. In practice, geometric points are computed by translating boundary points by a certain small distance along their point normals. It can be seen in Fig. 6 that normals vertical to region boundary lead to more planar interface, thus we propose a method generating such relatively vertical and consistent normals for boundary points.

Since boundary contour is a closed curve, simply using plane fitting on boundary vertex and its neighbors within a small range often fails in the collineation area [19]. Moreover, it remains a unsolved problem in plane fitting that the consistency of normal orientation is not guaranteed on all points as shown in Fig. 7. In order to generate continuous normals for the boundary points, we compute a fixed point  $\mathbf{c}(x_c, y_c, z_c)$  to control their

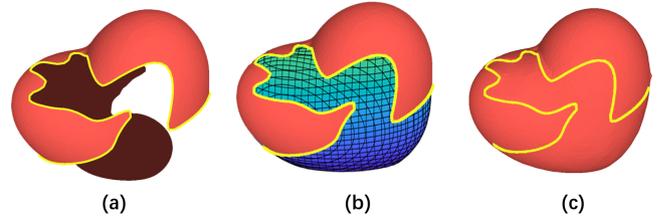


Fig. 8. Example of interface reconstruction on boundary vertices with RBF. (a) The original surface region with complex boundary. (b) The iso-surface generated by RBF. (c) The final reconstruction result.

normal, with which the normal of a control point  $\mathbf{p}_i$  is set as  $\mathbf{p}_i - \mathbf{c}$ . To force the normal vertical to a small area of plane through  $\mathbf{p}_i$ , the dot product between normal  $\mathbf{p}_i - \mathbf{c}$  and vector  $\mathbf{p}_i - \mathbf{p}_j$  should be close to 0, where  $\mathbf{p}_j$  is the neighboring point of  $\mathbf{p}_i$ . To minimize the energy function:

$$E = \frac{1}{NM} \sum_{i=1}^N \sum_{k=1}^M \left[ (\mathbf{p}_{ik} - \mathbf{p}_i) (\mathbf{p}_i - \mathbf{c})^T \right]^2, \quad (5)$$

we taking the derivative of  $x_c, y_c, z_c$ , we can obtain a system of linear equations in three variables, whose solution is our desired fixed point  $\mathbf{c}(x_c, y_c, z_c)$  if it exists. Otherwise the centroid of outer surface points will be assigned to the fixed point. With this fixed point, the geometric point  $\mathbf{p}'_i$  is computed by:

$$\mathbf{p}'_i = \mathbf{p}_i + \frac{\mathbf{p}_i - \mathbf{c}}{\|\mathbf{p}_i - \mathbf{c}\|}. \quad (6)$$

Fig. 8 shows an example of interface reconstruction on boundary vertices with RBF implicit surface. It can be seen that the RBF based method can generate the high-quality interface with complex boundary.

Then the remaining unlabeled tetrahedrons should be classified referring to the built RBF implicit surface. Using implicit function generated from RBF, unassigned tetrahedrons are grouped into inner and outer parts by computing the function value with vertex positions. A tetrahedron will be assigned to inner part if the implicit function value of one of its vertices is less than zero, and vice versa. Subsequently, we need to determine whether the inner or outer side belongs to current region. As our approach aims at generating minimal surface between each pair of adjacent regions, we separately compute the interface area of both sides, which is the sum of all triangular facets on the proposed interface. Inner or outer part with smaller interface area will be selected as the resultant part to be allocated to current region.

The tetrahedron allocation scheme mentioned above may not cover all tetrahedrons in the volume, leaving the attribute labels of a small amount of tetrahedrons to be further decided. Therefore, we further build a probability for attribute assignment based on the observation that a region with attribute label  $l_i$  closer to a unlabeled tetrahedron is more likely to be allocated to it. Minimal distance from each unlabeled tetrahedron to the outer surface is taken as the score for each tetrahedron, which is defined as:

$$Score(l_i, l_h) = \frac{d(l_i, l_h)^2}{A'(l_i)}, \quad (7)$$

where  $d(l_i, l_h)$  is the shortest distance of a unlabeled tetrahedron  $l_h$  to the closest surface vertex of region with label  $l_i$  and  $A'(l_i)$  is the total exposed surface area of label  $l_i$ . A higher score reflects a higher probability for label assignment, guiding the assignment of attribute labels for the remaining unlabeled tetrahedrons.

### 3.3. Interface Smoothing

We obtain the initial segmentation interfaces in Sec. 3.2 treated as rough interfaces for separating volumetric parts, whose geometric shape are coarse and irregular. It is challenging to directly realize such micro irregularity with printers, thus smoothing the interfaces becomes a sensible step to further refine their shape. The goal of surface smoothing is to reduce mean curvature of interfaces to produce easy-to-manufacture model parts. However, for concave regions, there may occur intersection between interface vertex and outer surface during smoothing iterations. Therefore, smoothing should be restricted with a certain threshold indicating a surface offset [40] inwards a certain distance from the outer surface, where the offset distance is a hyper-parameter that is often set to a prescribed minimal printing thickness. In this work, we develop a differential evolution (abbr. DE) based smoothing method to facilitate surface convergence and acquire better approximation heading for minimal surface. The basic DE algorithm [41, 42, 43] is a method for searching an optimal solution to minimize a fitness function by iteratively improving candidate solutions based on an evolutionary process. It starts with a group of candidate solutions with random initialization value and moving them around in the search space. These candidate solutions are considered as individuals, in the form of a real vector in which each real number in the vector called gene is updated with certain rules during iterations. In each iteration, DE generates children individual for each existing individual by using linear combination operation [41] on three individuals picked randomly from the population, that is, the group of candidate solutions. The newly generated individual will be accepted and replaced the old one if it has improved or equal fitness, otherwise simply discarded. The process is repeated until the termination criterion is satisfied and finally return the individual with the best fitness as the optimal solution. In order to make DE smoothing effective for our scenario, we should address the design of genes for individuals, as well as the fitness function.

The input to DE smoothing algorithm is a surface mesh model  $S_M = (V, E)$  extracted from the interface between a pair of adjacent parts, which has the vertices  $V = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{n-1}, \mathbf{v}_n\}$  and edges  $E$ . To transform DE algorithm into surface refinement, our approach simulates smoothing process by iteratively moving vertices on interface along with the direction of Laplacian vector, obtaining the optimal solution with least curvature. Our approach regards the smoothing coefficient  $\alpha_i$  of each vertex  $\mathbf{v}_i \in V$  on interface as the gene in individuals. Let  $\mathbf{x} \in \mathbb{R}^n$  designate a candidate solution in the population. Then  $\mathbf{x} = [\alpha_1, \alpha_2, \dots, \alpha_n]$  is called individual and each element  $\alpha_i \in [0, 2]$  is a gene. The coefficient reflects the offset of a vertex from its original position to Laplacian center. Laplacian based iteration is selected as the update rule for fitness conver-

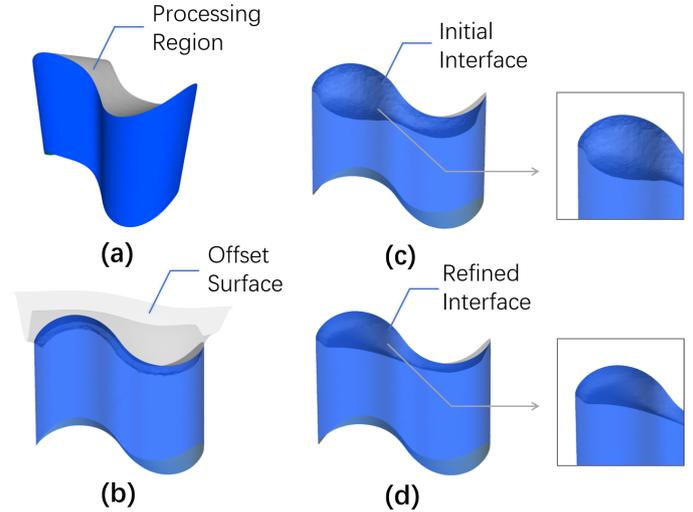


Fig. 9. DE smoothing workflow. (a) Original input model. (b) Offset surface of the processing region. (c) Initial segmentation result of the processing region. (d) Refinement result by DE smoothing.

gence, which is

$$\mathbf{v}'_i = \mathbf{v}_i + \alpha_i \cdot \mathbf{d}_i. \quad (8)$$

$\mathbf{v}'_i$  represents the updated vertex, and

$$\mathbf{d}_i = \sum_{(i,j) \in E} w_{ij} \mathbf{v}_j - \mathbf{v}_i, \quad (9)$$

$$w_{ij} = \frac{\omega_{ij}}{\sum_{\mathbf{v}_k \in N(\mathbf{v}_i)} \omega_{ik}}, \quad (10)$$

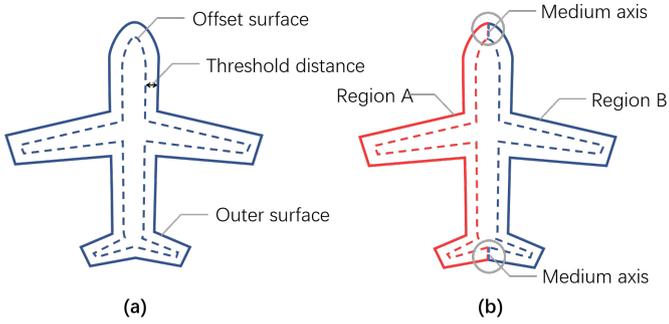
where  $N(\mathbf{v}_i)$  is the set of neighbors of  $\mathbf{v}_i$ . In our implementation, we set  $\omega_{ij} = 1$ . The local area of a vertex becomes smoother as it moves along the Laplacian vector  $\mathbf{d}_i$  both on convex and concave surface. At the end of each iteration, vertices position on interface will be updated with gene value of each individual, which means each individual preserves their independent vertices position information to form a complete interface. The individual corresponding to the interface with least fitness will be selected as the optimal solution.

Having determined the gene value and updating rule, the objective fitness function  $f^*$  of DE smoothing is naturally designed as:

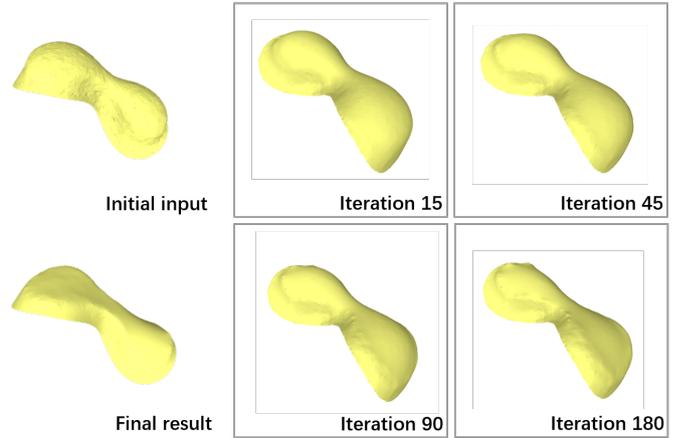
$$f^* = A(l_i) + \gamma \sum_{\mathbf{v}_i \in V} \|\mathbf{d}_i\| = A(l_i) + \gamma \sum_{\mathbf{v}_i \in V} \left\| \sum_{(i,j) \in E} \omega_{ij} \mathbf{v}_j - \mathbf{v}_i \right\|. \quad (11)$$

The first term  $A(l_i)$  measures the interface area of current region with label  $l_i$ , targeting at forcing the interface to converge to a surface with least area, while the second term  $\sum_{\mathbf{v}_i \in V} \|\mathbf{d}_i\|$  guarantees smoothness on interface globally. Here, we set  $\gamma$  to 0.1 in our experimental setup.

Ideally, we could obtain satisfactory segmentation interface via applying the illustrated DE smoothing algorithm. However, for non-convex region on the input surface, unconstrained smoothing may cause the vertices on interface intersecting the region surface. Moreover, fragile areas and thin structures may



**Fig. 10. Left: Offset surface for given outer surface. Right: offset surface for a pair of adjacent region.**



**Fig. 11. Iterative interface optimization through the smoothing process.**

interface as shown in Fig. 11.

44

---

#### Algorithm 1: DE smoothing algorithm

---

**Input:** Interfaces -  $I$ ; Offset surface function -  $F_o$ ;  
Iteration -  $T_N$ ; Threshold -  $\delta$

**Output:** The optimized interfaces -  $I'$

```

1  ▶ Initialization
2   $\Delta \leftarrow 0$ 
3   $t \leftarrow 1$ 
4  for  $I_{ij} \in I$  do
5    for  $\mathbf{v}_k \in I_{ij}$  do
6       $\mathbf{x}_t^k = \mathbf{v}_k$ 
7    end
8    while  $t \leq T_N$  or  $\Delta \leq \delta$  do
9      ▶ Select individual with minimal fitness by DE
10      $f_{min}(\mathbf{x}_t) \leftarrow DE(\mathbf{x}_t)$ 
11     ▶ Update vertex position
12     for  $\mathbf{v}_k \in I_{ij}$  do
13       if  $\mathbf{x}_t^k$  not lies inside  $F_o$  then
14          $\mathbf{x}_{t+1}^k = \mathbf{x}_t^k$ 
15       else
16          $\mathbf{x}_{t+1}^k = \mathbf{x}_{t-1}^k$ 
17       end
18     end
19      $\Delta \leftarrow f_{min}(\mathbf{x}_t) - f_{min}(\mathbf{x}_{t-1})$ 
20      $t \leftarrow t + 1$ 
21   end
22 end
23 return Optimized interfaces -  $I'$ 

```

---

### 3.4. Extraction Refinement

45

The RBF implicit surface together with DE smoothing generates a refined volume segmentation respecting the boundaries of the attribute regions on surface. However, the extractability of those volumetric segments is not explicitly guaranteed, resulting in possible failure of assembling the printed parts as a complete whole model. Thus, it demands further refinement on them to meet the requirements of realizable fabrication.

46

47

48

49

50

51

52

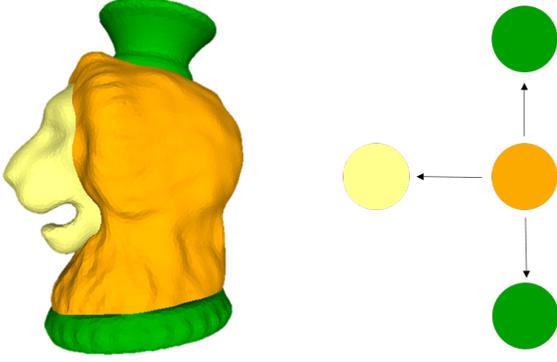


Fig. 12. Topological graph of a model generated by surrounding relationship between each pair of adjacent volumetric parts. Left: the initial segmentation model. Right: its corresponding topological graph. The directed edge pointing from orange node to yellow node indicates that orange part surrounds the adjacent yellow part.

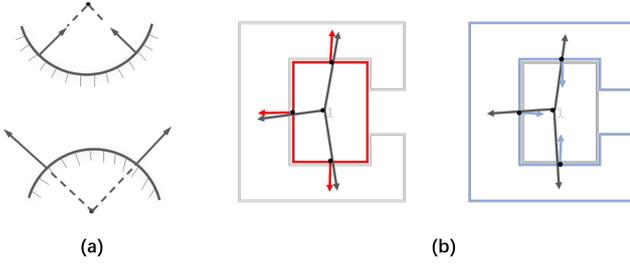


Fig. 13. Example of convexity check result for a surrounding situation. (a) Normal orientation for concave and convex surface. (b) Convexity/concavity check of interfaces based on interface normal and central point. The red box on the left is defined as convex because  $s_c > 0$ , while the blue shape on the right is concave due to  $s_c < 0$ .

We simplify this refinement as an iterative process of bi-splitting a model part into assemblable contiguous sub-volumes with roughly equal size. To ensure least collision, two key factors should be taken into consideration, which are processing sequence and division plane generation. Processing sequence means the order to sub-divide infeasible parts guided by the connectivity graph of the initial segmentation. Division plane is the cut plane dividing a volume into two parts, whose the volume sizes are balanced and extractability is optimized.

**Processing sequence.** To generate an appropriate sequence, we start from the volumetric part with smallest size to establish a directed topological graph  $G = \{N, E\}$ , where nodes  $H$  denotes each volumetric part of current segmented model and the directed edges  $E$  reflect the surrounding relationship between connected nodes as shown in Fig. 12. Firstly, all the vertices on the interface  $I_{ij}$  between a part  $H_i$  and its adjacent part  $H_j$  are used to compute a centroid point  $\mathbf{c}_{ij}$ :

$$\mathbf{c}_{ij} = \frac{1}{n} \sum_{\mathbf{v}_k \in I_{ij}} \mathbf{v}_k. \quad (12)$$

To determine whether  $H_i$  is an outer or inner part, we check for the convexity of the interface  $I_{ij}$  belonging to  $H_i$ . As indicated in Fig. 13, the difference between convexity and concavity

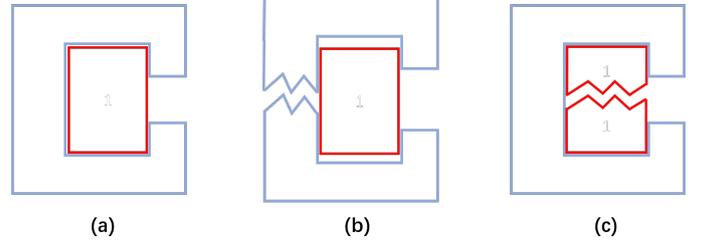


Fig. 14. Binary segmentation result of surrounding situation. (a) shows surrounding relationship of a pair of adjacent region, which red region is inner part and blue region is the outer part surrounding inner volume. (b) shows binary segmentation on the outer part succeeding in extractability and (c) shows binary segmentation on the inner volume failing in disassembly.

can be distinguished by the orientation of surface normal. For convex surface, the surface normal point away from the curvature center, whereas the concave case has the opposite. If we connect center  $\mathbf{c}_{ij}$  to each vertex  $\mathbf{v}_k \in I_{ij}$  on the interface, the directional vector  $\mathbf{v}_k - \mathbf{c}_{ij}$  will invert to the normal of vertex if concave, i.e.  $(\mathbf{v}_k - \mathbf{c}_{ij}) \cdot \mathbf{n}_k$  is less than zero. Therefore, we categorize the type of  $I_{ij}$  by summing up the dot product of each vertex normal on interface and its directional vector regarding  $\mathbf{c}_{ij}$ :

$$s_c = \sum_{\mathbf{v}_k \in I_{ij}} (\mathbf{v}_k - \mathbf{c}_{ij}) \cdot \mathbf{n}_k. \quad (13)$$

If the sum  $s_c$  is larger than zero, we consider the interface to be convex, and vice versa. Since concavity infers a surface or a line that is curved inward, concave interface  $I_{ij}$  reflects that part  $H_i$  surrounds part  $H_j$  in shape. Thus the weight of  $e_{ij} \in E$  will be assigned to one denoting that a directed edge from  $H_i$  to its adjacent part  $H_j$ . This condition can be represented as:

$$\begin{aligned} e(i, j) &= 1, \text{ if } I_{ij} \text{ is concave;} \\ e(j, i) &= 1, \text{ if } I_{ij} \text{ is convex.} \end{aligned} \quad (14)$$

Having built up the topological graph of the model, we determine a reasonable extracting sequence using topological sorting that results in a new sequence of the volumetric parts from the outermost volumetric one to the innermost one. As illustrated in Fig. 14, if the inner part is split before the outer part, it is impossible to achieve extractability. Thus, our approach applies bi-splitting referring to topological ordering, avoiding invalid operation caused by surrounding relationship. A topological ordering is possible if and only if the graph  $G$  has no directed cycles. Otherwise, we temporarily start with the largest volume part to apply further division that will be addressed later. We test extractability of each volumetric part with the validation method proposed in Surface2Volume [4], guaranteeing that each vertex lying on exposed surface of current part does not intersect any triangular facet belonging to other part along defined extraction direction.

**Division plane generation.** To efficiently cut a volumetric part, we consider the option of directly bi-splitting it with well-designed plane, whose decisive parameters to are its anchor point and normal vector. Anchor point can be located at the centroid of all the interface vertices of volumetric part  $H_i$ . We

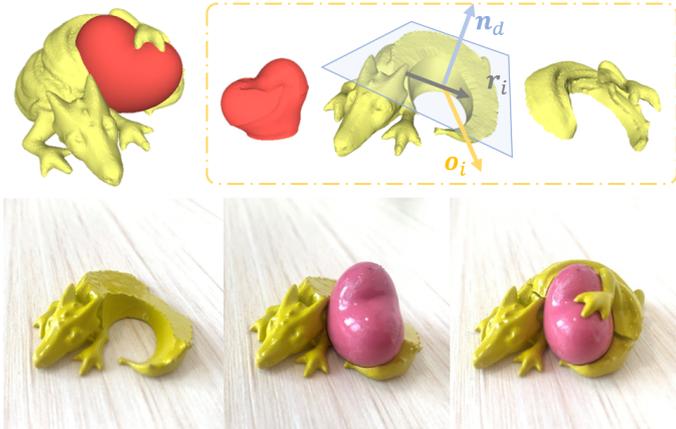


Fig. 15. Binary splitting of a segment with single attribute using computed cut plane for ensuring model assemblability.

1 select a pair of vertices on interface with maximal distance to  
 2 get a vector  $r_i$  between these two vertices reflecting the largest  
 3 diameter of the convex hull wrapping the interface of part  $H_i$ .  
 4 Since part  $H_i$  is obstructed while disassembling along with the  
 5 extracting direction  $o_i$ , the plane normal perpendicular to  $o_i$  is  
 6 most likely to become assemblable with least conflict. So the  
 7 plane normal  $n_d$  is computed by the cross product of extraction  
 8 direction  $o_i$  and diameter vector  $r_i$ . For each examined part we  
 9 divide it into sub-volumes and disallow small pieces by merg-  
 10 ing them into closest region. Fig. 15 shows an example of our  
 11 subdivision result.

#### 4. Results

12  
 13 We validate our proposed volume segmentation method on  
 14 various types of input models and demonstrate its performance  
 15 and effectiveness by showing our fabricated results together  
 16 with statistical analysis. To evaluate its robustness, the input  
 17 models in our experiments vary in boundary contour, model  
 18 shape as well as model size. The pre-defined attributes of in-  
 19 put models are represented by different colors on model sur-  
 20 face. The results show our method succeeds in handling sim-  
 21 ple and complicated models, generating high-quality segmen-  
 22 tation outputs that are easy to be printed and assembled using to-  
 23 day's mainstream printers. We provide various results contain-  
 24 ing 300,000 to 1,200,000 tetrahedrons, whose original vertex  
 25 number ranges from 50,000 to 200,000. All reported segmen-  
 26 tation outputs are generated in 5–25 minutes with our C++ im-  
 27 plementation, which is finished on a PC with Intel(R) i5-4590  
 28 CPU, 16GB RAM and Windows 10. We use ZRapid iSLA660  
 29 printer to manufacture all our printed results. Our method con-  
 30 tains a set of hyper parameters controlling the segmentation  
 31 process. Tuning these parameters for a certain input model may  
 32 help improve partition quality but decrease the quality for oth-  
 33 ers. To verify the robustness of our method, we process all the  
 34 examples with a fixed set of parameters to balance the perfor-  
 35 mance of our algorithm on all models. **Segmentation results.**  
 36 Fig. 16 provides the segmentation results together with their  
 37 fabricated objects with desired colors. The first column shows



Fig. 16. Various model segmentation results using our approach and their corresponding printed objects.



Fig. 17. An example of multi-material fabrication. The chair legs are fabricated by flexible material that can be deformed to different shape.

the input models that are closed surface mesh with prescribed  
 color attributes and the second column demonstrates the cor-  
 responding output segments using our implemented technique.  
 Different colored segments are manufactured separately with  
 resin material, leading to final glued objects which are shown  
 in the third and fourth columns respectively. Besides multi-color  
 model production, our approach can easily yield multi-material  
 model satisfying different physical requirements such as hard-  
 ness, luster, etc. For example, the chair model in Fig. 17 has a  
 rigid back and slightly deformable legs due to different print-  
 ing materials.

38  
 39  
 40  
 41  
 42  
 43  
 44  
 45  
 46  
 47  
 48

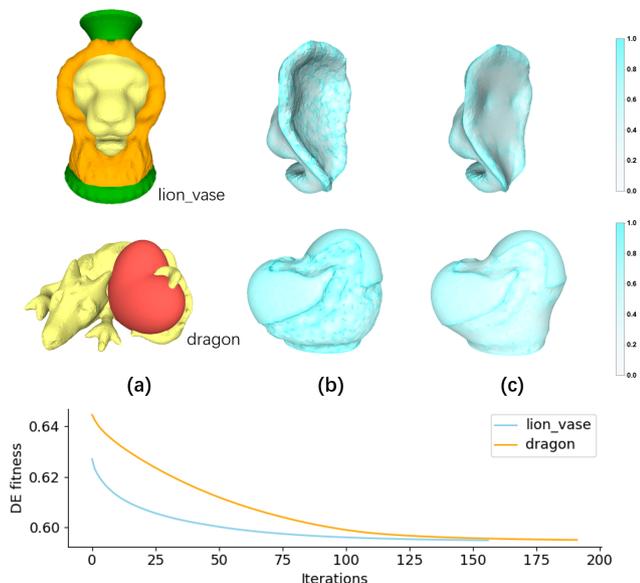


Fig. 18. Example of interface refinement by DE smoothing, where vertices on interface are colored according to the absolute value of mean curvature. (a) The original surface model. (b) Initial interface before DE iteration. (c) Refined interface after DE smoothing.

**Convergence of DE smoothing.** Our method exploits DE algorithm to compute minimal surface with fixed boundary contour, transferring the vertex positions of surface into the optimal solution for minimizing DE fitness function. Fig. 18 shows the interfaces deformation of volume parts before and after DE iteration. Since each vertex on interface is colored by the absolute value of mean curvature, lighter color means smaller mean curvature value of single vertex. It can be witnessed that our DE smoothing algorithm effectively reduces the global mean curvature of vertices as well as the area of interface. During training, the value of DE fitness decreases along with interface shrink as shown in the line chart of Fig. 18 both on lion\_vase and dragon model. Since the fitness value reflects the area of interface as well as smoothness on surface, part interface comes into smaller area and smoother shape as the objective value effectively reduces until the fitness converges to a certain number after iterations.

**Comparison with Surface2Volume.** Although there exist many prior methods addressing the problem of model segmentation for 3D printing, it lacks efforts made to propose model partition driven by attributes. Previous work that utilizes surface sweeping along a computed direction [3] or exploits multi-cut method dividing model into parts [4] is defective in interface quality and wall thickness preservation. As shown in Fig. 20, our method outperforms Surface2Volume [4] on wall thickness preservation and interface smoothness. In Fig. 20, wall thickness on different vertices is visualized by color, where the green and yellow areas reflect that the wall thickness of those vertices is smaller than the minimal printing thickness which are called illegal vertices, and red regions show that the minimum wall thickness is respected. It is easy to notice that our resultant part contains smaller amount of illegal vertices while the output from Surface2Volume [4] possesses larger illegal area near

the sharp edge of this model part. The scatter plot in Fig. 20 provides the distribution of wall thickness of each vertex on contour lines, indicating that our result achieves higher portion of vertices satisfying the wall thickness requirement. Contour lines on outer surface are determined by offsetting certain distance away from surface region boundaries of the input model as shown in Fig. 20. Because too-thin regions of the model increase the risk of manufacturing fragile part, it may cause the 3D printed model easy to break in transportation. Fig. 21 shows a failure example of volumetric parts fabrication using Surface2Volume [4]. In contrast, our result with average thickness succeeds in producing high-quality physical model. The second row in Fig. 21 shows that our method is convenient to adjust the wall thickness by setting the threshold in the DE smoothing step to meet different requirements. For example, it is necessary to consider the diameter of the printer nozzle when determines the minimal wall thickness of the 3D model in FDM printing. Except for product quality, we reduce the fabrication time by segmenting the printed models into parts with smoother interfaces. As shown in Fig. 19, the height of the bar in the chart reflects the total printing time of each model listing in the horizontal axis, while color regions inside the bar indicate the respective time of each volumetric part among a complete model. It can be witnessed that our method obviously reduces the fabrication time.

**Limitations.** Our method generates demarcation surface respecting each boundary contour with the constraint of wall thickness. However, non-optimal interface is obtained in some cases due to our process scheme. Specifically, during the iteration of DE smoothing introduced in Sec. 3.3, we fix the position of an interface vertex if it falls inside the offset surface, which may lead to unsatisfied interface shape of current region as shown in Fig. 22(a), i.e. the interface would not shrink to the optimal minimal surface but terminate at a defective status. This limitation could be solved by increasing the tolerance of the position of vertex lying inside the offset surface. If the dihedral angle between two adjacent facets corresponding to different attribute labels is larger than 180 degrees, the medium axis is not the optimal direction dividing two adjacent region (Fig. 22(b)). Reversed surface normal direction is the best choice to guarantee wall thickness as shown in Fig. 22(c). To improve the result quality of DE smoothing, we need to compute offset surface more precisely by taking local shape of the contact area of two adjacent regions into consideration. With more reasonable smoothing constraint, the positions of interface vertices can be updated more flexibly, allowing the DE smoothing algorithm to achieve minimal surface with better quality.

## 5. Conclusions

In this paper, we introduce a practical technique for generating assemblable volume segmentation of closed surface models guided by pre-defined surface attributes. Our method presents an appropriate workflow producing minimal surface as initial segment interface, whose quality is further improved by applying a DE smoothing algorithm. Model assemblability is promised by identifying and cutting invalid segmented parts

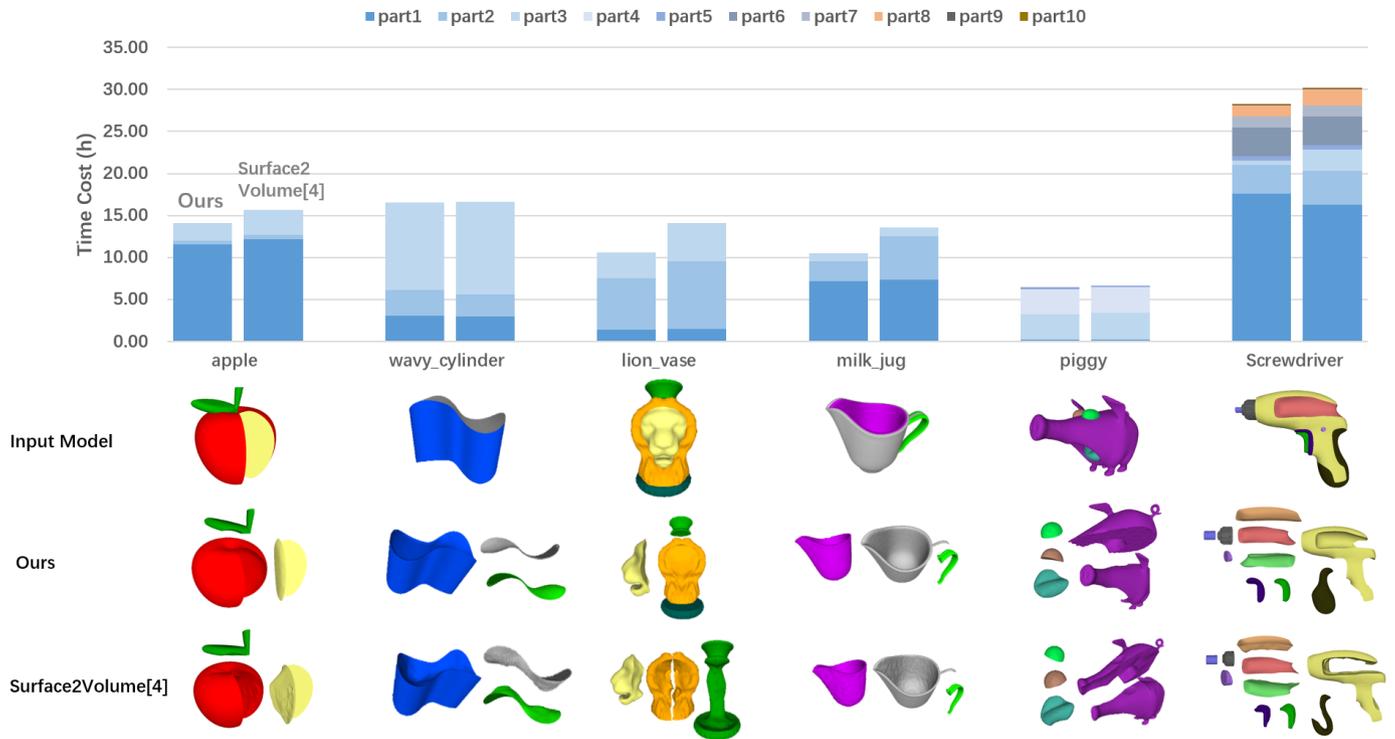


Fig. 19. Comparisons between our method and Surface2Volume [4] in terms of segmentation results and fabrication time in 3D-printing.

with the help of a topological graph. The provided experimental results validate the effectiveness of our approach on various models, and an additional comparison with Surface2Volume [4] shows its advantages in terms of interface smoothness and naturalness.

For future work, we would like to explore the possibility of enhancing the smoothing quality according to the discussion in Sec. 4. It will also be intriguing to research applying our method to more application scenarios, especially engineering applications which our approach may bring unique value to.

## Acknowledgements

This paper is supported by Natural Science Foundation of Guangdong Province (No.2019A1515011793, No.2017A030313347, No.2021A1515011849), the Fundamental Research Funds for the Central Universities(No.2020ZYGXZR042), and NSFC(No.61972160, No.51978271).

## References

- [1] Chen, X, Zhang, H, Lin, J, Hu, R, Lu, L, Huang, QX, et al. Dapper: decompose-and-pack for 3d printing. *ACM Trans Graph* 2015;34(6):213–1.
- [2] Luo, L, Baran, I, Rusinkiewicz, S, Matusik, W. Chopper: Partitioning models into 3d-printable parts. *ACM Transactions on Graphics (TOG)* 2012;31(6):1–9.
- [3] Yao, J, Kaufman, DM, Gingold, Y, Agrawala, M. Interactive design and stability analysis of decorative joinery for furniture. *ACM Transactions on Graphics (TOG)* 2017;36(2):1–16.
- [4] Araújo, C, Cabiddu, D, Attene, M, Livesu, M, Vining, N, Sheffer, A. Surface2volume: surface segmentation conforming assemblable volumetric partition. *ACM Transactions on Graphics (TOG)* 2019;38(4):1–16.
- [5] Ho, TC, Chuang, JH, et al. Volume based mesh segmentation. *Journal of Information Science and Engineering* 2012;28(4).
- [6] Vanek, J, Galicia, JG, Benes, B, Měch, R, Carr, N, Stava, O, et al. Packmerger: A 3d print volume optimizer. In: *Computer Graphics Forum*; vol. 33. Wiley Online Library; 2014, p. 322–332.
- [7] Attene, M. Shapes in a box: Disassembling 3d objects for efficient packing and fabrication. In: *Computer Graphics Forum*; vol. 34. Wiley Online Library; 2015, p. 64–76.
- [8] Strothoff, B, Jüttler, B. Automatic decomposition of 3d solids into contractible pieces using reeb graphs. *Computer-Aided Design* 2017;90:157–167.
- [9] Fu, CW, Song, P, Yan, X, Yang, LW, Jayaraman, PK, Cohen-Or, D. Computational interlocking furniture assembly. *ACM Transactions on Graphics (TOG)* 2015;34(4):1–11.
- [10] Song, P, Fu, Z, Liu, L, Fu, CW. Printing 3d objects with interlocking parts. *Computer Aided Geometric Design* 2015;35:137–148.
- [11] Song, P, Deng, B, Wang, Z, Dong, Z, Li, W, Fu, CW, et al. Cofifab: coarse-to-fine fabrication of large 3d objects. *ACM Transactions on Graphics (TOG)* 2016;35(4):1–11.
- [12] Jadoon, AK, Wu, C, Liu, YJ, He, Y, Wang, CC. Interactive partitioning of 3d models into printable parts. *IEEE computer graphics and applications* 2018;38(4):38–53.
- [13] Gao, W, Zhang, Y, Nazzetta, DC, Ramani, K, Cipra, RJ. Revomaker: Enabling multi-directional and functionally-embedded 3d printing using a rotational cuboidal platform. In: *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. 2015, p. 437–446.
- [14] Wu, C, Dai, C, Fang, G, Liu, YJ, Wang, CC. General support-effective decomposition for multi-directional 3-d printing. *IEEE Transactions on Automation Science and Engineering* 2019;17(2):599–610.
- [15] Khatamian, A, Arabnia, HR. Survey on 3d surface reconstruction. *Journal of Information Processing Systems* 2016;12(3).
- [16] Blinn, JF. A generalization of algebraic surface drawing. *ACM transactions on graphics (TOG)* 1982;1(3):235–256.
- [17] Muraki, S. Volumetric shape description of range data using “blobby model”. In: *Proceedings of the 18th annual conference on Computer*

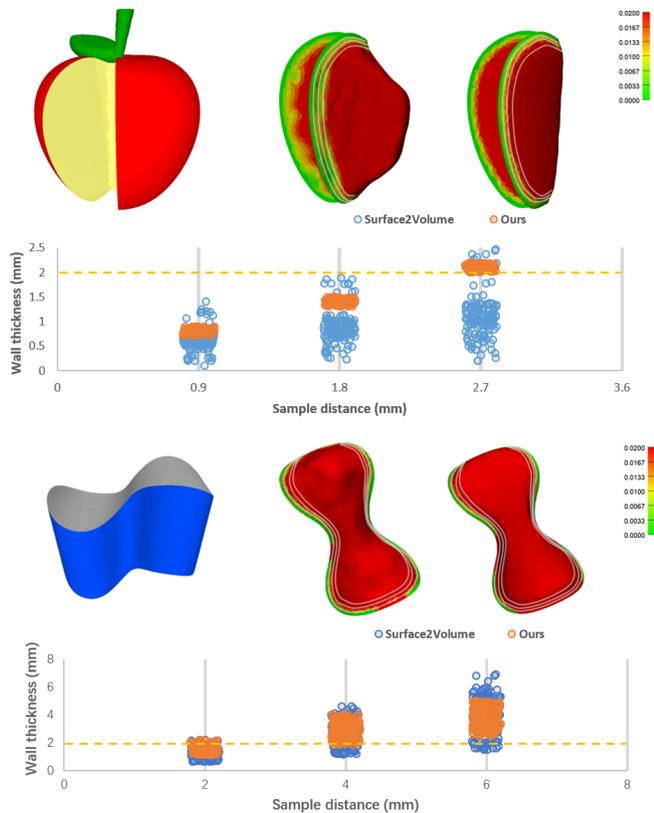


Fig. 20. Comparison of results between our method and Surface2Volume [4]. This figure shows the scatter plot of wall thickness distributions and visualized wall thickness of obtained segmented parts. The point data in the scatter plot are sampled from the contour lines on the surface of the volumetric part.

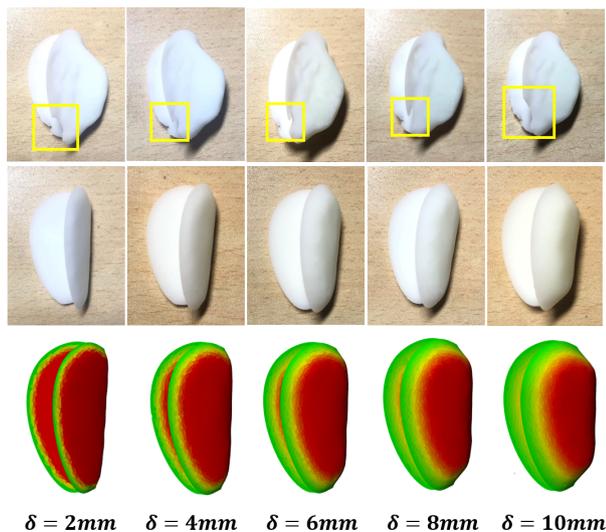


Fig. 21. Segmentation results of different thickness threshold and comparison results of our method and Surface2Volume [4]. The first row are the real printed results of the apple model using Surface2Volume [4], and the second row shows the fabrication results using our segmentation method. The third row are the corresponding digital models generated by our method with different thickness.

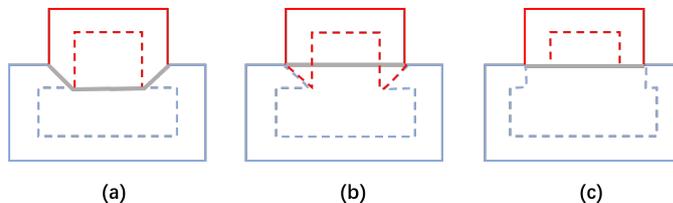


Fig. 22. Interface reconstruction restricted by wall thickness. (a) Interface of current region generated by our method. (b) Optimal interface with least area of current region. (c) Optimal interface generated from refined offset surface.

[18] Kazhdan, M, Bolitho, M, Hoppe, H. Poisson surface reconstruction. In: Proceedings of the fourth Eurographics symposium on Geometry processing; vol. 7. 2006.

[19] Hoppe, H, DeRose, T, Duchamp, T, McDonald, J, Stuetzle, W. Surface reconstruction from unorganized points. In: Proceedings of the 19th annual conference on computer graphics and interactive techniques. 1992, p. 71–78.

[20] Zhao, HK, Osher, S, Fedkiw, R. Fast surface reconstruction using the level set method. In: Proceedings IEEE Workshop on Variational and Level Set Methods in Computer Vision. IEEE; 2001, p. 194–201.

[21] Dinh, HQ, Turk, G, Slabaugh, G. Reconstructing surfaces using anisotropic basis functions. In: Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001; vol. 2. IEEE; 2001, p. 606–613.

[22] Carr, JC, Beatson, RK, Cherrrie, JB, Mitchell, TJ, Fright, WR, McCallum, BC, et al. Reconstruction and representation of 3d objects with radial basis functions. In: Proceedings of the 28th annual conference on Computer graphics and interactive techniques. 2001, p. 67–76.

[23] Alexa, M, Behr, J, Cohen-Or, D, Fleishman, S, Levin, D, Silva, CT. Computing and rendering point set surfaces. IEEE Transactions on visualization and computer graphics 2003;9(1):3–15.

[24] Duan, Y, Yang, L, Qin, H, Samarasinghe, D. Shape reconstruction from 3d and 2d data using pde-based deformable surfaces. In: European conference on computer vision. Springer; 2004, p. 238–251.

[25] Hornung, A, Kobbelt, L. Robust reconstruction of watertight 3d models from non-uniformly sampled point clouds without normal information. In: Symposium on geometry processing. 2006, p. 41–50.

[26] Alliez, P, Cohen-Steiner, D, Tong, Y, Desbrun, M. Voronoi-based variational reconstruction of unoriented point sets. In: Symposium on Geometry processing; vol. 7. 2007, p. 39–48.

[27] Huang, H, Li, D, Zhang, H, Ascher, U, Cohen-Or, D. Consolidation of unorganized point clouds for surface reconstruction. ACM transactions on graphics (TOG) 2009;28(5):1–7.

[28] Pan, R, Meng, X, Whangbo, T. Hermite variational implicit surface reconstruction. Science in China Series F: Information Sciences 2009;52(2):308–315.

[29] Liu, S, Brunnert, G, Wang, J. Multi-level hermite variational interpolation and quasi-interpolation. The Visual Computer 2013;29(6):627–637.

[30] Carr, JC, Beatson, RK, McCallum, BC, Fright, WR, McLennan, TJ, Mitchell, TJ. Smooth surface reconstruction from noisy range data. In: Proceedings of the 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia. 2003, p. 119–ff.

[31] Ohtake, Y, Belyaev, A, Seidel, HP. 3d scattered data interpolation and approximation with multilevel compactly supported rbfs. Graphical Models 2005;67(3):150–165.

[32] Branch, J, Prieto, F, Boulanger, P. Automatic hole-filling of triangular meshes using local radial basis function. In: Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06). IEEE; 2006, p. 727–734.

[33] Macedo, I, Gois, JP, Velho, L. Hermite radial basis functions implicit. In: Computer Graphics Forum; vol. 30. Wiley Online Library; 2011, p. 27–42.

[34] Liu, S, Wang, CC. Quasi-interpolation for surface reconstruction from scattered data with radial basis function. Computer Aided Geometric Design 2012;29(7):435–447.

[35] Liu, S, Wang, CC, Brunnert, G, Wang, J. A closed-form formulation of hrbf-based surface reconstruction by approximate solution. Computer-

- 1 Aided Design 2016;78:147–157.
- 2 [36] Kazhdan, M, Hoppe, H. Screened poisson surface reconstruction. ACM  
3 Transactions on Graphics (ToG) 2013;32(3):1–13.
- 4 [37] Si, H. Tetgen, a delaunay-based quality tetrahedral mesh generator. ACM  
5 Transactions on Mathematical Software (TOMS) 2015;41(2):1–36.
- 6 [38] Lagrange, JL. Essai d’une nouvelle methode pour de’terminer les max-  
7 ima, et les minima des formules integrales indefinies. 1761.
- 8 [39] Meeks III, W, Pérez, J. The classical theory of minimal surfaces. Bulletin  
9 of the American Mathematical Society 2011;48(3):325–407.
- 10 [40] Liu, S, Wang, CC. Duplex fitting of zero-level and offset surfaces.  
11 Computer-Aided Design 2009;41(4):268–281.
- 12 [41] Storn, R, Price, K. Differential evolution—a simple and efficient heuristic  
13 for global optimization over continuous spaces. Journal of global opti-  
14 mization 1997;11(4):341–359.
- 15 [42] Das, S, Suganthan, PN. Differential evolution: A survey of the state-of-  
16 the-art. IEEE transactions on evolutionary computation 2010;15(1):4–31.
- 17 [43] Das, S, Mullick, SS, Suganthan, PN. Recent advances in differen-  
18 tial evolution—an updated survey. Swarm and Evolutionary Computation  
19 2016;27:1–30.