Cross section-based hollowing and structural enhancement

Wang, Weiming; Li, Baojun; Qian, Sicheng; Liu, Yong-Jin; Wang, Charlie; Liu, Ligang ; Yin, Baocai ; Liu, Xiuping

DOI
10.1007/s00371-017-1386-5

Publication date
2017

Document Version
Accepted author manuscript

Published in
The Visual Computer: international journal of computer graphics

Citation (APA)
Wang, W., Li, B., Qian, S., Liu, Y.-J., Wang, C., Liu, L., Yin, B., & Liu, X. (2017). Cross section-based hollowing and structural enhancement. *The Visual Computer: international journal of computer graphics*, *33*(6-8), 949-960. https://doi.org/10.1007/s00371-017-1386-5

Noname manuscript No.
(will be inserted by the editor)

Cross Section based Hollowing and Structural Enhancement

Weiming Wang · Baojun Li · Sicheng Qian · Yongjin Liu · Charlie C.L. Wang · Ligang Liu · Baocai Yin · Xiuping Liu

Received: date / Accepted: date

Abstract Recently, 3D printing has become a powerful tool for personal fabrication. However, the price of some materials are still high which limits its applications in home users. To optimize the volume of the model, while not largely affecting the strength of the objects, researchers propose algorithms to divide the model with different kind of light weight structures, such as frame structure, honeycomb-cell structure, truss structure, medial axis tree, etc. However, these algorithms are not suitable for the model whose internal space needs to be reused. In addition, the structural strength and static stability of the models, obtained with modern 3D model acquirement methods, are not guaranteed. In consequence, some models are too fragile to print, and cannot be survived in daily usage, handling, and transportation, or cannot stand in a stable.

To handle the mentioned problems, an algorithm system is proposed based on cross sections in this work. The structural weak cross sections are enhanced and structural strong cross sections are adaptively hollowed to meet a given structural strength, static stability, printability, etc., while the material usage is minimized. The proposed algorithm system has been tested on several typical 3D models. The experimental results demonstrate the effectiveness and practicability of our system.

Keywords 3D Printing · Cross Section · Adaptive Hollowing · Structural Enhancement

Weiming Wang
Dalian University of Technology (wwmdlut@dlut.edu.cn)

Baojun Li
Dalian University of Technology (bjli@dlut.edu.cn)

Yongjin Liu
Tsinghua University (liuyongjin@tsinghua.edu.cn)

Charlie C.L. Wang
Delft University of Technology (c.c.wang@tudelft.nl)

Ligang Liu
University of Science and Technology of China (lgliu@ustc.edu.cn)

Baocai Yin
Dalian University of Technology (ybc@dlut.edu.cn)

Xiuping Liu
Dalian University of Technology (xpliu@dlut.edu.cn)

1 Introduction

3D printers are able to manufacture arbitrary complex 3D objects, and many types of material can be used in 3D printing. Although the cost of 3D printers have come down significantly over the past few years, the price of some materials are still high. As a result, saving material while maintaining the strength of the model is one of the major consideration of most users, especially in industry. Therefore, it is an important task to minimize the usage of material while not largely affecting the strength of the model. 3D printing softwares [16,6] are able to save material usage through uniformly filling or hollowing. But users need to heuristically choice an appropriate density or thickness, to balance among structural strength, static stability, and material usage. For inexperienced users, it is often a very difficult task.

Each 3D model has an upright orientation, and we hope the printed objects are able to stand stable on the plane determined by their upright orientations. However, we find many models cannot meet our expectation. Since the static stability is not considered during the design or generation of the model. Furthermore, the structural soundness of the models are also not considered during obtainment. In consequence, some parts of the models are difficult to print, and the manufactured objects are commonly fragile that they cannot be survived in daily usage, handling, and transportation, and some of them are even collapsed under their own weight.

Therefore, the stability and structural strength of the 3D models should be analyzed and enhanced before printing.

To simultaneously deal with the above problems, we propose an algorithm system based on cross sections. Inspired by the structural analysis method presented in [29], we use bending momentum equilibrium to quickly detect weak and strong cross sections. The proposed system automatically enhances all structural weak cross sections, and adaptively hollows the structural strong cross sections, to achieve the minimal volume of the models, while constrained by a given structural strength, stability, printability, etc. Our system has been tested on a large number of 3D models. Experimental results demonstrate that our system is able to save 75% printing material for the model with huge bodies on average (see Table 1), and enhance all weak parts of the fragile models (see Figure 7 and Figure 8).

2 Related Work

3D Printing Currently, 3D printing has become a hot topic in Computer Graphics. A large number of algorithms have been proposed to deal with complex geometric problems in 3D printing. Some of them attempt to design mechanical toy and automata [45,4], articulated models [1,28], external supporting structures [25,7,30]. The other algorithms are proposed to improve printing efficiency [31,34] and quality [10,37], shape details enhancement [21], and dynamic stability [2,18,19,33]. In contrast, our work is proposed to reduce the material cost, fix the structural weak regions, and optimize static stability of the object.

Material Cost Reduction In computer graphics community, many efforts have been spent on saving printing material, while maintaining a certain strength of the printed objects.

Wang et al. [36] use frame structures to fill the interior of the model with the constraints of stiffness and strength, static stability, printability, etc. Similarly, the work by Zhang et al. [42] uses medial axis tree instead of frame structures, which can naturally transfer the external loads from different directions to the inner core structure. Different from these methods, Lu et al. [14] and e S$\acute{a}$ et al. [24] divide the interior of the model with honeycomb-cell structure which is known to be of minimal material cost, while providing strength in tension.

The algorithms described above have a common point: they divide the interior of the model with light weight structures. However, for some models, their interior spaces have special usage, such as money-box and music box, some hardware should be installed inside. Therefore these algorithms are not suitable for this kind of models any more. In contrast, we adaptively hollow the model through cross section based optimization, while achieve a certain strength of the object, and ensuring its static stability and printability.

Structural Analysis Since structural soundness of 3D models are not considered during the acquirement, many models are difficult to print and transportation, etc. A large number of algorithms have been proposed to analyze the weak regions of the model and even fix them.

Stava et al. [27] propose a novel algorithm to detect the structural problematic regions with the given external loads, and fix them through heuristically hollowing, thickening, and strut insertion. Zhou et al. [44] propose an optimization framework to analyze all weak regions without specific external loads. However, the finite element method (FEM) used in their algorithms is time-consuming when the input models are complex. Xie et al. [38] propose an accelerated analysis algorithm through locally updating stiffness matrix of FEM. Furthermore, Umetani et al. [29] and Xu et al. [39] analyze the key stress of the model based on bending momentum equilibrium which enables interactive analysis speed. In this work, the method proposed by Umetani et al. [29] is used to detect structural weak and strong regions. The weak regions are enhanced, and the strong regions are adaptively hollowed which is not considered in [29] and [39].

Static Stability With the advent of 3D printing technologies, it becomes very easy to detect whether a model is static stability or not. Traditionally, users often glue the printed objects onto a heavy pedestal which is very tedious. To automatically implement the static stability of an object, Pr$\acute{e}$vost et al. [23] propose a formulation to modify the volume of the object, while preserving its surface details. For some cases, they need to interactively edit the shape of the model during the optimization. However, their method do not consider the material cost and structural strength. Yamanaka et al. [40] control the mass center of the object by optimize the density distribution of the truss structure. Although the volume of the model is optimized in their method, they also do not consider the structural strength of the model. Musialski et al. [18,19] propose shape optimization frameworks, which can achieve both static and dynamic stability. Although [19] takes into account structural strength in shape optimization, it is difficult for them to handle structural weak regions. In this work, we propose an algorithm system to simultaneously optimize stability, material usage, and enhance weak parts.

Offset Surface Offsetting is a fundamental and important geometric operation in rapid prototyping. Although the offsetting operation is mathematically well defined, offsetting a solid model exactly has proven to be difficult [11]. To handle this problem, researchers have paid many attentions to generate offsetting surfaces for triangular mesh [12, 5,13,32]. Kim et al. [12] present an algorithm by moving all vertices along the multiple normal vectors of a vertex

to generate offsetting surface. Liu et al. [13] propose a fast offset surface generation algorithm to construct intersection-free offset surfaces, while preserving sharp features. Chen et al. [5] use layered depth-normal images to uniformly offset the surface. An accelerated version of [5] is proposed by Wang et al. [32] to organize the samples into structured points, and performs parallel computations using multiple cores. Although these algorithms are suitable for 3D printing, they are not cost-effective, and structural strength and static stability are not guaranteed.

Hollowing To design the deformation behavior of 3D printed object, Zhang et al. [41] propose an interactive tool. Given a soft material to be used in 3D printing, the thickness of the model is optimized by considering bending behavior of this material. To print a boundary surface, we need to generate a solid volume by some operator. However, it dose not guarantee the structural strength of the generated volume. To solve this problem, Zhao et al. [43] propose an algorithm to optimize the thickness parameters according to stress constraints to extrude the surface for a given boundary surface and user-specified external loads. However, both algorithms do not handle structural weak regions.

3 Structural Analysis

In this work, we first use cross section based structural analysis method which has been studied in [29], to detect structural problematic regions that should be enhanced, and strong regions that can be hollowed. Although this method can only give us an approximate detection results, but it is fast and the detected results is good enough for the proposed system. In the following, we will briefly introduce some details of cross-sectional structural analysis method that closely related to our algorithms.

For a given 3D manifold triangular mesh M, we slice it into N cross sections in a direction randomly sampled on Gaussian Shape (which will be further discussed in Sec. (4.3.4)), with slicing thickness $\{h_i\}_{i=1}^{i=N}$. As we know, structural weakness of a cross section is defined considering two factors: the magnitude of force needed to break a cross section, and the area over which the force is distributed. Thus, let $f_{min}(\Gamma, p, \phi)$ be the minimal force at position p on the surface, needed to produce the critical stress on cross section Γ in bending direction ϕ. With bending momentum equilibrium, f_{min} can be calculated as follows:

$$f_{min}(\Gamma, p, \phi) = |(I - \xi \otimes \xi)(p - g)|\hat{\sigma}_L Z_\xi \eta, \tag{1}$$

where, ξ and η are the axis of local coordinate system of Γ, $Z_\xi = I_\xi / |\eta|_{max}$, I_ξ is the second moment of area, $\hat{\sigma}_L$ is maximum stress which is given as 60MPa, and g is the mass center of the model. The smaller the force f_{min}, the more fragile the cross section Γ.

For a cross-section, we choose the minimal force from all possible force points and bending directions to evaluate its fragility, that is

$$f_{min}(\Gamma) = \min_{p,\phi} f_{min}(\Gamma, p, \phi), \tag{2}$$

If $f_{min}(\Gamma) < \hat{f}_{max}$, it is considered as fragile and should be enhanced, and if $f_{min}(\Gamma) > \alpha \hat{f}_{max}, \alpha \in [1, +\infty]$, it is regarded as strong cross section and can be hollowed. Where, $\hat{f}_{max}$ is the critical force to distinguish between structural weak and strong cross sections, which is set as 0.2 kgf in [29], and α is a parameter that can be given by users to define structural strong cross sections. Eq.(1) will be used as mechanical constraint in our algorithm system (Section 4.3), to guarantee structural strength of the model. The interested readers are recommended to article [29] for more details.

4 Algorithm

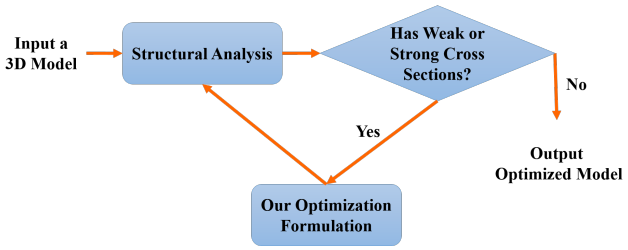


Fig. 1 Overview of the proposed system.

In this section, we propose an algorithm system to automatically enhance the structural problematic cross sections and adaptively hollow the strong cross sections of the model. The overview of our system is shown in Figure 1.

A cross section Γ is a polygon, composed of n ordered points $\Gamma = \{\mathbf{p}_i\}_{i=1}^{i=n}$ (see the green points in Figure 2). In general, a model can be divided into two parts by each cross section. The part with smaller area is donated as S^+ and the other part is donated as S^-. To calculate f_{min}, force points and bending directions should be given. As described in [29], the points on S^+ are selected as force points. For each force point, we uniformly sample m directions as its bending directions, that is force directions. Our system moves cross section points along their shifting directions which will be introduced in Sec. 4.1. The points after optimized are indicated as $\mathbf{p}'_i = \mathbf{p}_i + t_i.\mathbf{d}_i, i = 1, 2, ..., n$, where t_i is the movement of $\mathbf{p}_i$ along direction $\mathbf{d}_i$, and the optimized version of Γ_i is denoted as Γ'_i. Since cross sections are optimized in 2D plane, $\mathbf{p}_i$ and $\mathbf{d}_i$ are 2D points and just have x-y coordinates, that is $\mathbf{p}_i = (p_x^i, p_y^i)$ and $\mathbf{d}_i = (d_x^i, d_y^i)$.

4.1 Shifting direction

Before presenting our system, we should first calculate shifting directions for cross section points. We can simply use point normal $\mathbf{n}_p$ or inverse point normal $-\mathbf{n}_p$ as shifting direction of point $\mathbf{p}$. But with these directions, the optimized cross section may self-intersect or intersect with the other cross sections. Therefore, the major factor to be considered in determining the shifting directions is intersection which will happen in concave regions. For every given cross-sectional polygon Γ, the shifting direction $\mathbf{d}_i$ for each vertex $\mathbf{p}_i$ is expected to:

- let Γ' be homeomorphic to Γ and have similar shape,
- avoid self-intersection on Γ',
- and provide relative large space for shifting.

In fact, the second and the third expectations are coupled – the harder self-intersections occurs, the larger range of displacement t_i on each $\mathbf{p}'_i$. To achieve these goals, the heuristic of letting neighboring $\mathbf{d}_i$s be nearly parallel is conducted. Similar heuristic method was used in [22] to generate 3D homeomorphic surfaces. Specifically, we use the 2D shape skeleton [3] to establish shifting direction for each vertex $\mathbf{p}_i$ on a polygon Γ.

Each point $\mathbf{q}$ in the skeleton $sk(\Gamma)$ of Γ is the center of a maximal disk $D(\mathbf{q})$ contained in Γ. We call a point $\mathbf{q} \in sk(\Gamma)$ the *mating skeleton point* of $\mathbf{p}$ (denoted by $\mathbf{q}(\mathbf{p})$) if $\mathbf{p}$ is on the boundary of $D(\mathbf{q})$. For each vertex $\mathbf{p}_i \in \Gamma$, a *proxy point* $\mathbf{v}(\mathbf{p}_i) \in sk(P)$ is defined for finding a shifting direction. An iterative algorithm is developed:

- First of all, let $\mathbf{v}(\mathbf{p}_i) = \mathbf{q}(\mathbf{p}_i)$;
- For every vertex $\mathbf{p}_i$, if $\|\mathbf{v}(\mathbf{p}_i) - \mathbf{p}_i\| < r_{\max}$, we search for a neighboring point $\mathbf{p}^*$ of $\mathbf{v}(\mathbf{p}_i)$ on $sk(\Gamma)$ that

$$\|\mathbf{p}^* - \mathbf{p}_i\| > \|\mathbf{v}(\mathbf{p}_i) - \mathbf{p}_i\|,$$

  and then $\mathbf{p}^*$ is assigned as the new proxy point of $\mathbf{p}_i$;
- The update of proxy point is repeatedly applied until $\|\mathbf{v}(\mathbf{p}_i) - \mathbf{p}_i\| \geq r_{\max}$ is satisfied on all vertices.
- For weak cross sections, the shifting directions of the points are outer vectors, that is $\mathbf{d}^i_w = \frac{\mathbf{p}_i - \mathbf{v}(\mathbf{p}_i)}{||\mathbf{p}_i - \mathbf{v}(\mathbf{p}_i)||}$. While they are inner vectors for strong cross sections, that is $\mathbf{d}^i_s = -\mathbf{d}^i_w$.

Note that $r_{\max}$ is a maximally allowed hollowing thickness which can be determined experimentally or by the manufacturability based analysis taken in the following section. An illustration of our algorithm is shown in Fig.2. It can be found that the shifting vectors between neighboring vertices become more and more conformal to each other and the shitting will have less chance to generate self-intersection on Γ'.

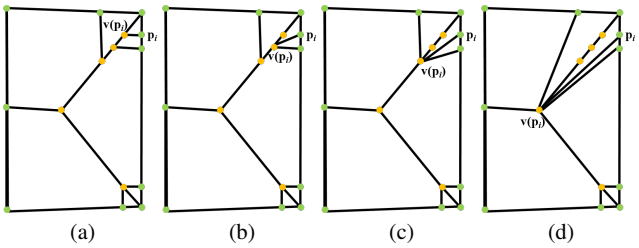


Fig. 2 Some points may have poor shifting directions. We search for new proxy points for them. Yellow points are skeleton points and green points are cross sections points.

4.2 Objective and Constraints

Moving Bounds For structural enhancement, the movement of the lower bound is zero $L^p_w = 0$. To avoid intersection, we should calculate a safe upper bound for each cross section point. For point $\mathbf{p}$, we generate a ray $r(\mathbf{p})$whose end point is $\mathbf{p}$ and the direction is $\mathbf{d}^p_w$. The first intersection point $\mathbf{v}$ between $r(\mathbf{p})$ and the other rays is found. The distance between $\mathbf{p}$ and $\mathbf{v}$ is set as its movement upper bound of $\mathbf{p}$, that is $U^p_w = ||\mathbf{p} - \mathbf{v}||$. If $r(\mathbf{p})$ intersects nothing, its upper bound is set to infinity, $U^p_w = +\infty$.

For adaptive hollowing, the length between $\mathbf{p}$ and $\mathbf{v}(\mathbf{p})$ is set as the moving upper bound of $\mathbf{p}$, that is $U^p_s = ||\mathbf{p} - \mathbf{v}(\mathbf{p})||$. To guarantee the printability of the model, the movement lower bounds should be larger than the printable lower bound ξ_{min} of the printer. However, moving distance along moving direction cannot directly reflect the thickness of the surface, because most moving directions are not perpendicular to the surface. To obtain a valid moving lower bound for point $\mathbf{p}$, we project the desired minimal printing thickness value in its normal direction $\mathbf{n}_p$ onto its moving direction $\mathbf{d}^p_s$ to get a valid moving lower bound:

$$L^p_s = \frac{\xi_{min}}{(\mathbf{d}^p_s)^T \mathbf{n}_p}, \tag{3}$$

Cross-Section Area The goal of our system is to minimize the volume of the model, and it can be approximately calculated with the area of the cross sections. For a cross section Γ, its area is calculated as follows:

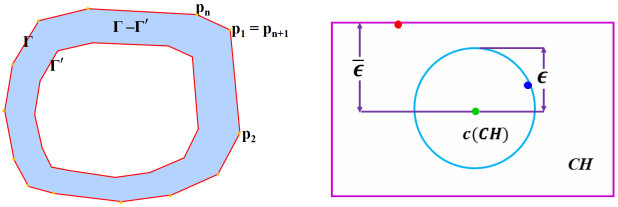
$$S(\Gamma) = \sum_{i=1}^{i=n} \frac{1}{2} \begin{vmatrix} p^i_x & p^{(i+1)}_x \\ p^i_y & p^{(i+1)}_y \end{vmatrix}, \tag{4}$$

where, $|.|$ indicates the determinant of matrix, and $\mathbf{p}_1 = \mathbf{p}_{n+1}$ (see in Fig. 3(a)).

Structural Strength To guarantee the optimized model has a certain structural strength, a structural strength constraint should be added in our algorithm system. According to Eq.(1), the structural strength constraint is defined as follows:

$$f_{min}(\Gamma, \mathbf{q}_k, \phi_j) > \gamma, \quad \mathbf{q}_k \in S^+, j = 1, 2, ..., m \tag{5}$$

where, γ is a parameter to control the strength of the optimized model which must bigger than $\hat{f}_{max}$, that is $\gamma > \hat{f}_{max}$, $\mathbf{q}_k$ is a force point, and ϕ_j is one of its bending directions.



(a) (b)

Fig. 3 2D polygons before and after optimization (a). Notations used in static stability constraint (b).

Static Stability The upright orientation, which can be obtained by the state-of-art-methods [8,35], of the model defines its supporting plane. The points of the model on the supporting plane are named as supporting points. As we know, to achieve static stability, the projected mass center of an object, along its gravity direction, must be inside of the convex hull CH (see magenta polygon in Fig. 3(b)) of the supporting points. Thus, we have the following static stability constraint:

$$c(M)^{\perp g} \in CH \tag{6}$$

where, $c(M)$ is the mass center of the model M, g is the gravity direction($\|g\| = 1$), and $^{\perp g}$ denotes the perpendicular projection onto the supporting plane along g. Constraint Eq(6) is able to pull $c(M)^{\perp g}$ inside of CH, but it will just on the boundary of H (see the red point in Fig. 3(b)). To improve it (see the blue point in Fig. 3(b)), an optional constraint is given in the following:

$$||c(M)^{\perp g} - c(CH)|| < \epsilon, \tag{7}$$

where, $c(CH)$ is the mass center of CH (see the green point in Fig. 3(b)), ϵ is a parameter to control the distance between $c(M)^{\perp g}$ and $c(CH)$. In Fig. 3(b), the upper bound $\bar{\epsilon}$ of ϵ is the inscribed circle radius of CH.

Given a cross section Γ, we represent its hollowed version as Γ' (see the polygons in Fig. 3(a)). Based on cross sections, we can approximately and easily calculate the mass center $\hat{c}$ of a hollowed model:

$$\hat{c}(M) = \frac{Vol(M)c(M) - \sum_{i=1}^{i=N} S(\Gamma_i')h_i c(\Gamma_i')}{Vol(M) - \sum_{i=1}^{i=N} S(\Gamma_i')h_i} \tag{8}$$

where, $S(\Gamma')$ is the area of cross section Γ', which is calculated with Eq.(4), $Vol(M)$ is the volume of the model M, h_i is the thickness of the i-th cross section, and $c(\Gamma')$ is the mass center of cross section Γ'.

4.3 Algorithm System

4.3.1 Formulation

After structural analysis with Eq.(2) under a given slicing direction, we collect the structural weak cross sections $\mathbf{\Gamma}_w = \{\Gamma | f_{min}(\Gamma) < \hat{f}_{max}\}$ and structural strong cross sections $\mathbf{\Gamma}_s = \{\Gamma | f_{min}(\Gamma) > \alpha \hat{f}_{max}, \alpha \in [1, +\infty]\}$. A cross section Γ_i has n points, so it has n unknown $\mathbf{T}_i = [t_1^i, t_2^i, ..., t_n^i]$. Then, a cross section based optimization framework is proposed to optimize them, while enhancing the structural weak cross sections and hollowing the structural strong cross sections:

$$\begin{gathered}
\min_{\{\mathbf{T}_i\}} \textstyle\sum_{\Gamma_i \in \mathbf{\Gamma}_w \cup \mathbf{\Gamma}_s} S(\Gamma_i, \Gamma_i') h_i + \\
\lambda \textstyle\sum_{\Gamma_i \in \mathbf{\Gamma}_w} (var(\mathbf{T}_i) + \sum_{j \in Nei(i)} ||var(\mathbf{T}_i) - var(\mathbf{T}_j)||^2) \\
\text{s.t.} \\
L_w^{ik} \le t_k^i \le U_w^{ik}, \quad \Gamma_i \in \mathbf{\Gamma}_w, k = 1, 2, ..., n_1 \\
L_s^{ik} \le t_k^i \le U_s^{ik}, \quad \Gamma_i \in \mathbf{\Gamma}_s, k = 1, 2, .., n_2 \\
(5), (6), \ or \ (7)
\end{gathered} \tag{9}$$

where, n_1 and n_2 are the number of points on the corresponding cross sections, ik means the k^{th} point on the i^{th} cross section, t_k^i represents the k^{th} element in $\mathbf{T}_i$. $var(\mathbf{T}_i)$ is the variance of the movement vector $\mathbf{T}_i$ of Γ_i, $Nei(i)$ is the adjacent cross sections of i^{th} cross section, and λ is a parameter to balance material cost and shape of the cross sections, which is set to a small value in our implementation. The second item in objective function of Eq.(9) is used to control the movement of the points on a cross section and its neighbor cross sections in $\mathbf{\Gamma}_w$. By optimizing this objective function, the shape of the weak cross regions and smoothness between neighboring cross sections can be preserved as much as possible. For the cross sections in $\mathbf{\Gamma}_s$, their details are not important, so we do not need to control the movement of their points. $S(\Gamma_i, \Gamma_i')$ represents the area difference between the initial cross section Γ_i and cross section Γ_i' after optimizing, which is defined as:

$$S(\Gamma_i, \Gamma_i') = \begin{cases} S(\Gamma_i') - S(\Gamma_i) & \Gamma_i \in \mathbf{\Gamma}_w \\ S(\Gamma_i) - S(\Gamma_i') & \Gamma_i \in \mathbf{\Gamma}_s \end{cases}, \tag{10}$$

where, $S(\Gamma)$ is the area of cross section Γ, which is defined in Eq.(4).

In formulation (9), both constraints (6) and (7) are able to pull the projected mass center inside of CH. But constraint (7) can better control the distance between $c(CH)$ and $c(M)^{\perp g}$, that is the stability. Therefore, constraint (7) is optimized in our experiments. And users can select an appropriate constraint according to their demands. The material usage and stability caused by constraint (7) will be discussed in Sec.(5.2).

Formulation(9) is a non-linear optimization problem, which can be effectively solved with interior-point algorithm provided in [20].

4.3.2 Deformation

Given the optimized weak cross sections $\mathbf{\Gamma}'_w$, the whole surface should be deformed to fit the shape of these cross sections, so that the weak regions are enhanced. We first fix the points on cross section in $\mathbf{\Gamma}'_w$, and then as-rigid-as possible surface modeling method [26] is used to deform the other regions of the model. Since the smoothness is considered in our optimization objective function, the finial shape will be smooth.

4.3.3 Hole Generation

Given the optimized (hollowed) cross sections $\mathbf{\Gamma}'_s$ on different layers, the interior surface M_0 is constructed by the strip-triangulation method [17] with a variation to avoid intersection between M and M_0. The efficient intersection detection is realized by the OBB-tree based proximity query [9].

4.3.4 Discussion

Our optimization framework (9) assumes a given slicing direction, therefore generates a direction dependent optimum. As seen in Fig. 1, our system includes an iteration in slicing direction, because cross section based structural analysis method is sensitive to slicing direction. Fig. 4 shows the structural analysis results of a Human model that is sliced in four different directions. From this figure we can see that different slicing direction will result in different weak and strong cross sections. For a given slicing direction, optimization formulation (9) can give us a solution at that direction. Then, slicing direction is changed and formulation (9) is implemented again. The whole process can be implemented iteratively, until the model has no weak regions. For each iteration, we are able to obtain holes and their total volumes. Finally, the holes in some iteration with the maximal total volume is selected as the final holes. However, to totally enhance the whole model, we sometimes need to sample a large number of slicing directions, resulting in a large number of iterations of our system. In consequence, the effectiveness of our system will be largely affected.

To effectively reduce the number of iterations, we first sample a large number of slicing directions on Gaussian sphere. Then cross section based structural analysis method (Sec.(3)) is used to detect weak cross sections in each direction. The number of weak cross sections for each slicing direction are counted and sorted in descending order. Finally, the first K directions is selected and used in our optimization system. In our experiments, K is set to 15.

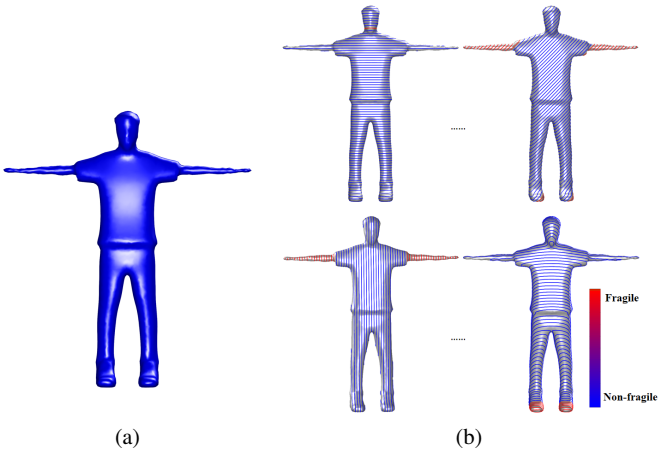


Fig. 4 The input 3D model (a) is sliced and analyzed in a large number of directions sampled on a Gaussian Sphere. Here we only show four analysis results of them (b). In (b), red cross sections are fragile and blue cross sections are non-fragile.

5 Experimental Results

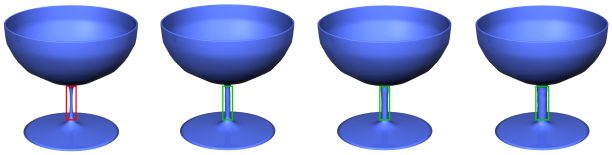
5.1 Implementation Details

Our algorithm system was implemented in mixed C++ and Matlab, and it was run on a PC with Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz and 32GB memory. The results are fabricated by [15] with printing layers of 0.2 mm. Its printing size is $285mm \times 153mm \times 155mm$. The printing material is white PLA plastic with a yielding strength of 60MPa and Young's modulus of 2300MPa. On average, our system needs 25 minutes to finish optimizing.

5.2 Parameters

In this work, several parameters should be set. m in Sec.(4) is set to 16, ε_{min} in Eq.(3) is set to 0.5 mm, λ in Eq.(9) is set to 0.1. For structural analysis, all models are sliced into cross sections with interval $h = 1.0\ mm$ in 100 directions with the form of $(x, y, z)^T$, where each of x, y, z takes the values (-1, 0, +1). Most models are uniformly scaled to have the height of $80mm$. In our experiments, these parameters are suitable for all models.

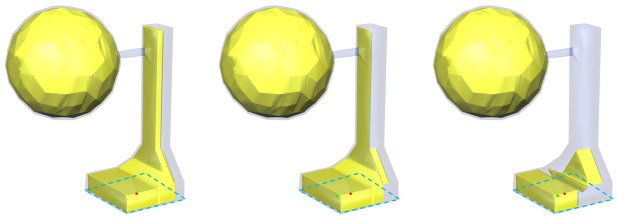
In addition, γ in Eq.(5) can be given by users to control the strength of the optimized model. In Figure 5, a cup model has a fragile leg (red box in Figure 5(a)). Our system optimizes it with three different γ: (b) $\gamma = 0.5kgf$, (c) $\gamma = 1.0kgf$, and (d) $\gamma = 2.0kgf$. As we can see from the green boxes, Figure 5(d) is generated with the largest γ, so it has the biggest structural strength. But its printing material is the most. Therefore, users are able to manually change γ to balance between material usage and structural strength according to their applications. In our experiments, its default value is set to 1.0 kgf.



(a) (b) (c) (d)

Fig. 5 The Cup model (a) is optimized with different γ: (b) 0.5 kgf, (c) 1.0 kgf, and (d) 2.0 kgf. (d) has the largest γ, so its leg is the strongest. However, it needs the most printing material. The volumes increased related to (a) are: 173.72 mm^3 (b), 453.2 mm^3 (c), and 1017.4 mm^3 (d). The red box in (a) shows the weak regions, and the boxes in (b), (c), and (d) are the enhanced regions.

ϵ in Eq.(7) is used to control the stability of the model. The Hangingball model shown in Figure 6 is optimized with three different ϵ. The model in Figure 6(a) is optimized with $\epsilon = 10mm$, which has smaller volume, while the model Figure 6(c) is optimized with $\epsilon = 4mm$, and it has much better stability. Therefore, users can manually adjust ϵ to balance between material usage and static stability of the model according to their requirements. In our experiments, its default value of ϵ is set as $\bar{\epsilon}/2$.



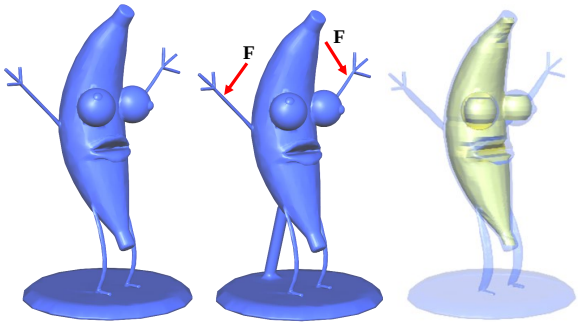
(a) (b) (c)

Fig. 6 The Hangingball model is optimized with different ϵ: 10mm (a), 6mm (b), and 4mm (c). The sky blue polygon is CH, red point is $c(CH)$, and green point is $c(M)^{\perp g}$. In (c), the model has the best stability, but it needs the most printing material to print. The volumes of these models are : 1.501 mm^4 , 1.808 mm^4, and 2.164 mm^4.

5.3 Structural Enhancement Comparisons

To evaluate the effectiveness of the proposed algorithm system for structural enhancement, we first compare our system with the algorithm proposed in [27]. In their work, FEM is used to analyze the stress distribution of the model, and the weak regions are fixed through thickening, hollowing, and adding external struts. For the Bananaman model (see Fig. 7 (a)), its legs are detected as weak regions, and fixed by adding external struts (see Fig. 7(b)). Although their method can guarantee the printability of these legs, they may be broken during cleaning, processing, and transportation. Moreover, the added strut will affect the visual quality of the model. Furthermore, they do not consider arms as structural weak regions. But, in fact, the arms of the Bananaman model are too fragile to sustain some external loads (see the red arrows in Fig. 7 (b)).

In this work, bending momentum equilibrium [29] is used to detect all possible weak regions of the model. In our experiment, the arms and legs of the Bananaman model are detected as weak regions, and enhanced with our system (9) (see Figure 7 (c)). The printed Bananaman model optimized with our system is shown in Fig. 12. In addition, our algorithm can minimize the added volume: the volume generated by their method and our system are 5.172×10^4, and 5.136×10^4 (without hollowing) and 2.283×10^4 (with hollowing), respectively.



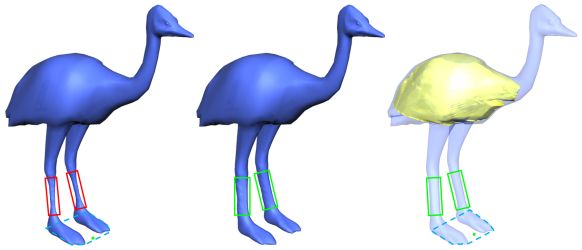
(a) (b) (c)

Fig. 7 Bananaman model (a) is structurally enhanced by Stava et al. [Stava et al. 2012] (b) and our algorithm system (c), respectively. Stava et al. add an external strut to increase its strength, which affects the visual quality of the model. However, the model generated by our method do not need external strut, and it has smaller volume than that in (b). Furthermore, the arms of the model in (c) are fixed which are not handled in (b). The volume of the models generated by these methods are: 5.172×10^4 (b) and 2.283×10^4 (c).

Similar to our method, Xu et al. [39] propose a structural enhancement algorithm based on cross sectional structural analysis method. With bending momentum equilibrium, they detect the weakest cross section, and heuristically calculate a scaling factor for the domain including this cross section. Although their method is able to enhance the weak regions, the domain includes the weakest cross section dose not need to have the same scaling factor, which may waste printing material. In addition, their heuristical strategy does not consider the smoothness of the cross sections. In contrast, we propose an optimization algorithm to optimize the shape of the weakest cross section, while minimizing the volume to be increased. Fig. 8 shows the Ostrich model optimized by both two methods. To be fair, our model is optimized with $\gamma = 2.0kgf$ which is used in [39] for this model. The model generated by our system (Fig. 8 (c)) has smaller volume than that in Fig. 8 (b).

In addition, the algorithms proposed in [27] and [39] do not optimize the strong regions to minimize the material usage and obtain static stability of the object. Although Stava et al. propose a heuristic hollowing strategy for stress relief, their method cannot minimize the volume of the model. In contrast, our system can minimize the volume of the structural strong regions (see the Fig. 7(c) and Fig. 8(c)).

Although $c^{\perp g}$ of the model (green point) in Fig. 8(a) is inside of CH (sky blue dotted polygon), it very closes to the



(a) (b) (c)

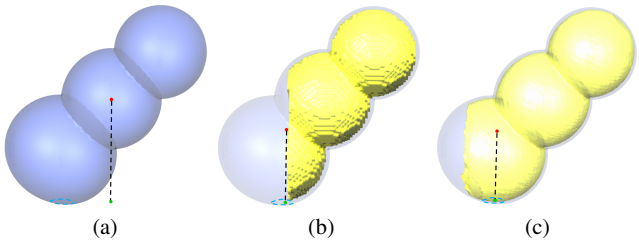
Fig. 8 Ostrich model (a) is structurally enhanced by Xu et al. [Xu et al.2016] (b) and our algorithm system (c). For this model, the volume of the model generated by their method is 4.524×10^4, while the volume of the model generated by our system is 4.485×10^4 (without hollowing) and 1.363×10^4 (with hollowing). In addition, our system is able to obtain a better static stability (see green points in (a) and (c).

boundary. Therefore, this model has a weak stability. Our system is able to obtain a better stability (see green point in Fig. 8(c)).

5.4 Static Stability Comparisons

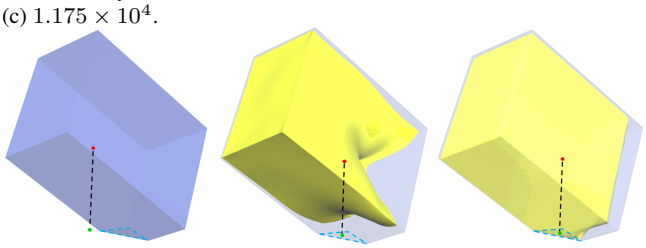
To achieve static stability, Prévost et al. [23] propose an algorithm to change the mass center of the model through heuristically hollowing or deformation. Their method can guarantee the static stability of the model under a given posture. However, it is difficult for them to control the volume and maintain the strength of the model. Musialski et al. [18] propose a shape optimization framework to meet different goals. For static stability, their goal is to minimize the distance between $c^{\perp g}$ and $c(CH)$. Since their method is a global optimization, so it needs less volume to achieve the same static stability as [23]. But their algorithm cannot minimize the printing material cost. Moreover, [23] and [18] do not consider the structural strength of the model. In addition, they sometimes may even hollow the structural weak regions, because their methods cannot distinguish between structural weak and strong regions. In contrast, our algorithm system is proposed to optimize the mass center of the model with minimal printing material cost, while maintaining a given structural strength, and printability.

Fig. 9 and 10 show the comparison results with [23] and [18], respectively. To be fair, the projected mass centers are optimized to the same positions as that in [23] and [18]. In Figure 9(b), the model generated by [23] is just heuristically hollowed without deformation, and the volume is 2.842×10^4 , while the volume of the model optimized by our system is 1.175×10^4 (Figure 9(c)). In Fig. 10, the volume of the model generated by [18] is 5.712×10^4 (Fig. 10(b)), while the volume of the model generated by our system is 1.289×10^4 (Fig. 10(c)).



(a) (b) (c)

Fig. 9 Comparison with the method of [Prévost et al. 2013] without surface deformation. The projected mass center of (a) is outside of CH (sky blue dotted polygon). (b) and (c) are generated by [Prévost et al. 2013] and our system, respectively. Both methods can achieve the same static stability, but the volume of the models are: (b) 2.842×10^4 and (c) 1.175×10^4.



(a) (b) (c)

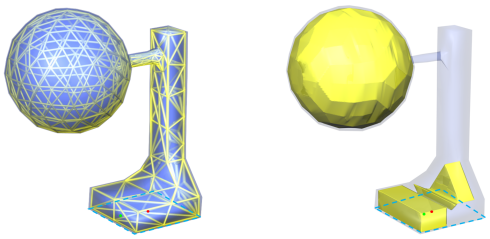
Fig. 10 Comparison with the method of [Musialski et al 2015] for static stability. The projected mass center of (a) is outside of CH (sky blue dotted polygon). (b) and (c) are generated by [Musialski et al 2015] and our system, respectively. Both methods can achieve the same static stability, but the volume of the models are: (b) 5.712×10^4 and (c) 1.289×10^4.

5.5 Material Reduction

Although several algorithms have been proposed to design light-weight structures for the purpose of reducing material usage, here we only compare with the algorithm [36] to evaluate our system. Because this method is the most close to our work. Both methods optimize the volume of the model, while constrained by structural strength, static stability, printability. We compare the material cost with their algorithm under the similar structural strength. In [36], the external force for the Hangingball model is set to 10N, so γ in formulation(9) is set to $1.0kgf$. Figure 11(a) is generated by [36], and its volume is $3.940 \times 10^4 mm^3$. While the volume of the model generated by our system (Fig. 11(b)) is $2.164 \times 10^4 mm^3$. In this figure, red point is $c(CH)$ and green point is $c^{\perp g}$. We can see from this figure, the model generated by our system not only has smaller volume, but also has better static stability.

In addition, the proposed system is able to enhance the structural weak regions, which is not considered in the state-of-arts-methods about structural optimization and shape optimization.

The printing material reduced by our system and running time of our system for all models are listed in Tab.1. It is worth noticing that the running time of our system is mainly related to factors: the number of cross section (the slicing thickness h), and the number of vertices on each cross



(a) (b)

Fig. 11 Comparison with the method of [Wang et al 2013] for material reduction. Both models are optimized with the same structural strength. Red points is $c(CH)$ and green point is $c^{\perp g}$. (a) is generated by [Wang et al 2013] and its volumes is $3.940 \times 10^4 mm^3$. (b) is generated by our system and its volume is $2.164 \times 10^4 mm^3$ which is smaller than that in (a). In addition, the static stability of (b) is better than that in (a).

section. From this table we can see that our system can save 75% printing material on average. Meanwhile, the printing time of the model optimized by our system can be effectively reduced. Fig. 12 shows the printed objects optimized with our system that appeared in our paper.

6 Conclusions and Future Work

In this work, we propose an algorithm system to simultaneously enhance structural problematic regions, and adaptively hollow structural strong regions, while constrained by structural strength, static stability, and printability. A number of experimental results illustrate the practicability and robustness of our algorithm system and demonstrate that our algorithm system can increase structural strength, reduce material cost, and meet static stability of the model.

However, our algorithm system still has several limitations. First, our algorithm system is designed to thicken the thin parts and dose not consider the external struts. If a model needs to add external struts, the algorithm introduced in [27] can be used. Second, our adaptive hollowing algorithm is mainly suitable for the model whose internal space needs to be reused. If users want to generate an object with light weight structures inside, our algorithm is not as good as [36, 14, 42].

In the future, we plan to develop an algorithm to combine adaptive hollowing and light weight structures designing. Because, for a complex 3D model with several huge parts, some parts may need to be filled with light weight structures, while the other parts' spaces may need to be reused.

Acknowledgements

We would like to thank the reviewers for their detailed comments and suggestions which greatly improved the manuscript.

The research leading to these results has received funding from China Postdoctoral Science Foundation (2016M601308), the One Hundred Talent Project of the Chinese Academy of Sciences, Fundamental Research Fund (DUT16RC(3)061), National Natural Science Foundation of China (61370143, 61432003, 61661130156, 61672482, 11626253, 11472073), Hong Kong RGC GRF (14207414), and Royal Society-Newton Advanced Fellowship (NA150431).

References

1. Bächer, M., Bickel, B., James, D.L., Pfister, H.: Fabricating articulated characters from skinned meshes. ACM Transactions on Graphics (TOG) **31**(4), 47 (2012)
2. Bächer, M., Whiting, E., Bickel, B., Sorkine-Hornung, O.: Spin-it: Optimizing moment of inertia for spinnable objects. ACM Transactions on Graphics (TOG) **33**(4), 96 (2014)
3. Cacciola, F.: 2D straight skeleton and polygon offsetting. In: CGAL User and Reference Manual, 4.2 edn. CGAL Editorial Board (2013)
4. Ceylan, D., Li, W., Mitra, N.J., Agrawala, M., Pauly, M.: Designing and fabricating mechanical automata from mocap sequences. ACM Transactions on Graphics (TOG) **32**(6), 186 (2013)
5. Chen, Y., Wang, C.C.: Uniform offsetting of polygonal model based on layered depth-normal images. Computer-aided design **43**(1), 31–46 (2011)
6. Cura: 3d printing software - cura. https://ultimaker.com/en/products/cura-software, year = 2016
7. Dumas, J., Hergel, J., Lefebvre, S.: Bridging the gap: automated steady scaffoldings for 3d printing. ACM Transactions on Graphics (TOG) **33**(4), 98 (2014)
8. Fu, H., Cohen-Or, D., Dror, G., Sheffer, A.: Upright orientation of man-made objects. In: ACM transactions on graphics (TOG), vol. 27, p. 42. ACM (2008)
9. Gottschalk, S., Lin, M.C., Manocha, D.: Obbtree: A hierarchical structure for rapid interference detection. In: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, pp. 171–180. ACM (1996)
10. Hildebrand, K., Bickel, B., Alexa, M.: Orthogonal slicing for additive manufacturing. Computers & Graphics **37**(6), 669–675 (2013)
11. Hoffmann, C.M.: Geometric and solid modeling (1989)
12. Kim, S.J., Lee, D.Y., Yang, M.Y.: Offset triangular mesh using the multiple normal vectors of a vertex. Computer-Aided Design and Applications **1**(1-4), 285–291 (2004)
13. Liu, S., Wang, C.C.: Fast intersection-free offset surface generation from freeform models with triangular meshes. Automation Science and Engineering, IEEE Transactions on **8**(2), 347–360 (2011)
14. Lu, L., Sharf, A., Zhao, H., Wei, Y., Fan, Q., Chen, X., Savoye, Y., Tu, C., Cohen-Or, D., Chen, B.: Build-to-last: Strength to weight 3d printed objects. ACM Transactions on Graphics (TOG) **33**(4), 97 (2014)
15. Makerbot: Rapid prototyping and 3D printing. http://www.makerbot.com/ (2012)
16. Makerware: 3d printing software - makerware. http://www.makerbot.com/blog/2013/12/09/makerbot-makerware-2-4-1-release, year = 2016
17. Meyers, D., Skinner, S., Sloan, K.: Surfaces from contours. ACM Transactions On Graphics (TOG) **11**(3), 228–258 (1992)
18. Musialski, P., Auzinger, T., Birsak, M., Wimmer, M., Kobbelt, L.: Reduced-order shape optimization using offset surfaces. ACM Trans. Graph **34**(4), 102 (2015)
19. Musialski, P., Hafner, C., Rist, F., Birsak, M., Wimmer, M., Kobbelt, L.: Non-linear shape optimization using local subspace projections. ACM Transactions on Graphics **35**(4) (2016)

Model	NV.	NF.	SV. ($10^4\ mm^3$)	OV. ($10^4\ mm^3$)	MR. (%)	MH. (mm)	Running Time (m)
Bananaman(Fig.7 and Fig.12)	9676	19352	5.136	1.265	75.2	120	33
Hangingball(Fig.6 and Fig.12)	266	528	8.442	2.164	74.4	80	16
Ostrich(Fig.8)	7046	14088	4.437	1.363	69.3	120	35
Buddha(Fig.12)	9999	19994	15.019	1.918	87.2	80	24
Frog	12495	24990	2.108	0.6183	69.9	80	28
Sphere(Fig.9 and Fig.12)	6042	12080	5.567	1.194	78.6	80	21
Cube(Fig.10 and Fig.12)	386	768	19.144	2.169	88.7	80	15
Bear(Fig.12)	10075	20146	5.263	1.182	78.5	80	22
Human(Fig.4)	75081	150170	0.881	0.363	58.8	80	30

Table 1 Printing material saved by our system(9). NV., NF., SV., OV., MH., and MR. represent the number of vertices and faces, Solid Volume, Optimized Volume, Model Height, and Material Reduced, respectively. The seventh column lists the percentage of the material reduced by our algorithm which is relative to the solid volume, that is MR = (SV - OV) / SV. The last column lists the running time of all models, and the unit is $minute$.

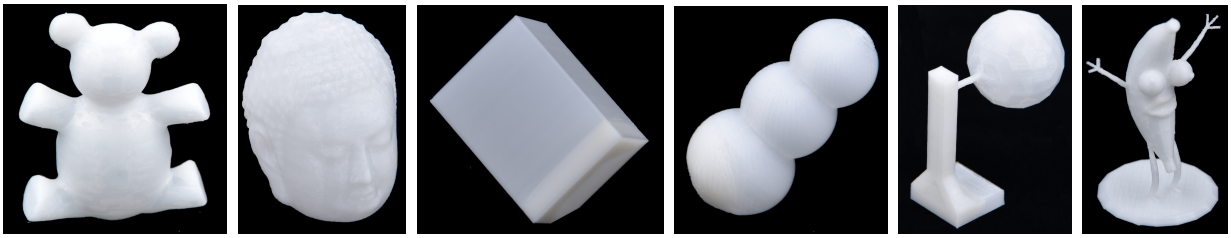


Fig. 12 Printed objects optimized with our system.

20. Nocedal, J., Wright, S.: Numerical optimization. Springer Science & Business Media (2006)
21. Pintus, R., Gobbetti, E., Cignoni, P., Scopigno, R.: Shape enhancement for rapid prototyping. The Visual Computer **26**(6-8), 831–840 (2010)
22. Porumbescu, S.D., Budge, B., Feng, L., Joy, K.I.: Shell maps. ACM Trans. Graph. **24**(3), 626–633 (2005). DOI 10.1145/1073204.1073239. URL http://doi.acm.org/10.1145/1073204.1073239
23. Prévost, R., Whiting, E., Lefebvre, S., Sorkine-Hornung, O.: Make it stand: balancing shapes for 3d fabrication. ACM Transactions on Graphics (TOG) **32**(4), 81 (2013)
24. e Sá, A.M., Mello, V.M., Echavarria, K.R., Covill, D.: Adaptive voids. The Visual Computer **31**(6-8), 799–808 (2015)
25. Schmidt, R., Umetani, N.: Branching support structures for 3d printing. In: ACM SIGGRAPH 2014 Studio, p. 9. ACM (2014)
26. Sorkine, O., Alexa, M.: As-rigid-as-possible surface modeling. In: Proceedings of EUROGRAPHICS/ACM SIGGRAPH Symposium on Geometry Processing, pp. 109–116 (2007)
27. Stava, O., Vanek, J., Benes, B., Carr, N., Měch, R.: Stress relief: improving structural strength of 3D printable objects. ACM Transactions on Graphics (Proc. SIGGRAPH) **31**(4), 48:1–11 (2012)
28. Sun, T., Zheng, C., Zhang, Y., Yin, C., Zheng, C., Zhou, K., Yue, Y., Smith, B., Batty, C., Xu, Z., et al.: Computational design of twisty joints and puzzles. ACM Transactions on Graphics (TOG) **34**(4), 101 (2015)
29. Umetani, N., Schmidt, R.: Cross-sectional structural analysis for 3d printing optimization. SIGGRAPH Asia **5**, 1–4 (2013)
30. Vanek, J., Galicia, J., Benes, B.: Clever support: efficient support structure generation for digital fabrication. In: Computer Graphics Forum, vol. 33, pp. 117–125. Wiley Online Library (2014)
31. Vanek, J., Galicia, J., Benes, B., Měch, R., Carr, N., Stava, O., Miller, G.: Packmerger: A 3d print volume optimizer. Computer Graphics Forum **33**(6), 322–332 (2014)
32. Wang, C.C., Manocha, D.: Gpu-based offset surface computation using point samples. Computer-Aided Design **45**(2), 321–330 (2013)
33. Wang, L., Whiting, E.: Buoyancy optimization for computational fabrication. In: Computer Graphics Forum, vol. 35, pp. 49–58. Wiley Online Library (2016)
34. Wang, W., Chao, H., Tong, J., Yang, Z., Tong, X., Li, H., Liu, X., Liu, L.: Saliency-preserving slicing optimization for effective 3d printing. In: Computer Graphics Forum, vol. 34, pp. 148–160. Wiley Online Library (2015)
35. Wang, W., Liu, X., Liu, L.: Upright orientation of 3d shapes via a tensor rank minimization. Journal of Mechanical Science and Technology **28**(7), 2469–2477 (2014)
36. Wang, W., Wang, T.Y., Yang, Z., Liu, L., Tong, X., Tong, W., Deng, J., Chen, F., Liu, X.: Cost-effective printing of 3d objects with skin-frame structures. ACM Transactions on Graphics (TOG) **32**(6), 177 (2013)
37. Wang, W.M., Zanni, C., Kobbelt, L.: Improved surface quality in 3d printing by optimizing the printing direction. In: Computer Graphics Forum, vol. 35, pp. 59–70. Wiley Online Library (2016)
38. Xie, Y., Xu, W., Yang, Y., Guo, X., Zhou, K.: Agile structural analysis for fabrication-aware shape editing. Computer Aided Geometric Design **35**, 163–179 (2015)
39. Xu, W., Li, W., Liu, L.: Skeleton-sectional structural analysis for 3d printing. Journal of Computer Science and Technology (In press) **31**(3) (2016)
40. Yamanaka, D., Suzuki, H., Ohtake, Y.: Density aware shape modeling to control mass properties of 3d printed objects. In: SIGGRAPH Asia 2014 Technical Briefs, p. 7. ACM (2014)
41. Zhang, X., Le, X., Wu, Z., Whiting, E., Wang, C.C.L.: Data-driven bending elasticity design by shell thickness. Computer Graphics Forum **35**(5), 157–166 (2016)
42. Zhang, X., Xia, Y., Wang, J., Yang, Z., Tu, C., Wang, W.: Medial axis treełan internal supporting structure for 3d printing. Computer Aided Geometric Design **35**, 149–162 (2015)
43. Zhao, H., Xu, W., Zhou, K., Yang, Y., Jin, X., Wu, H.: Stress-constrained thickness optimization for shell object fabrication. In: Computer Graphics Forum. Wiley Online Library (2016)
44. Zhou, Q., Panetta, J., Zorin, D.: Worst-case structural analysis. ACM Transactions on Graphics (TOG) **32**(4), 137 (2013)
45. Zhu, L., Xu, W., Snyder, J., Liu, Y., Wang, G., Guo, B.: Motion-guided mechanical toy modeling. ACM Trans. Graph. **31**(6), 127 (2012)