Computers & Graphics 66 (2017) 154–161

Contents lists available at ScienceDirect

Computers & Graphics

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

Special Issue on SMI 2017

Support-free frame structures

Weiming Wang[a], Sicheng Qian[a], Liping Lin[a], Baojun Li[a], Baocai Yin[a], Ligang Liu[b], Xiuping Liu[a,*]

[a] *Dalian University of Technology, Dalian, China*
[b] *University of Science and Technology of China, Hefei, China*

ARTICLE INFO

Article history:
Received 3 April 2017
Revised 17 May 2017
Accepted 25 May 2017
Available online 7 June 2017

Keywords:
Support-free
Lightweight structure
Frame structure
Overhanging angle

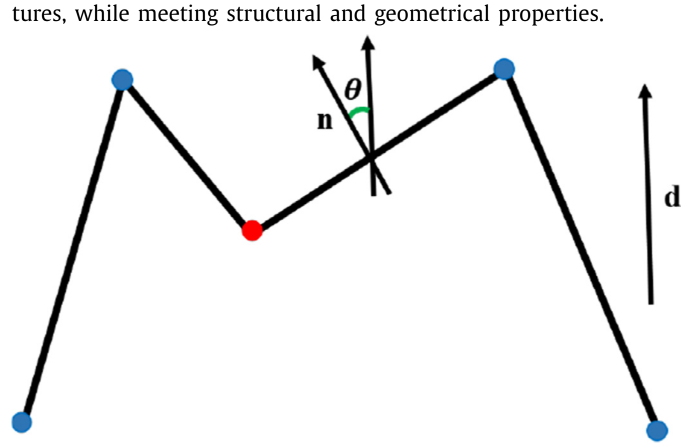
ABSTRACT

Recently, different complex lightweight structures are designed to save printing material, while satisfying specific demands on physical and geometrical properties. However, most structures need additional interior support structures to help print, which not only damage the physical properties of the objects but also consume additional material, time, and energy.

To handle the above problem, a sparsity optimization framework with support-free constraints is proposed in this work, to ensure the generated frame structures are support-free when manufactured with FDM printers. A struts elimination strategy is presented to avoid both inner and outer impending nodes. In addition, angles between adjacent inner struts and radii of struts are further optimized to reduce printing material while achieving a given structural strength. Extensive experimental results demonstrate the effectiveness and practicability of our algorithm.

1. Introduction

Additive manufacturing (3D printing) enables the physical realization of objects with unprecedented complexity from digital models. However, some mechanical properties are often overlooked during the design or generation process. In consequence, some models are too fragile to survive in printing, daily usage, and transportation. In addition, the cost of some materials are still high. Therefore, it is an important task to design lightweight structures, while meeting structural and geometrical properties.



To achieve the above goals, different kinds of light-weight structures have been proposed, such as skin-frame structure [1], honeycomb-cell structure [2], truss structure [3], medial axis tree [4], and bone-like porous structures [5], etc. In addition, some algorithms have been proposed to strengthen the structural weak regions by inserting additional struts [6]. Nevertheless, most structures generated with these algorithms cannot be directly printed without additional interior support structures, because of the impending surfaces, struts, or nodes. Therefore, they often suffer from the problems of adding and removing the supporting structures. A

* Corresponding author.
E-mail address: xpliu@dlut.edu.cn (X. Liu).

http://dx.doi.org/10.1016/j.cag.2017.05.022

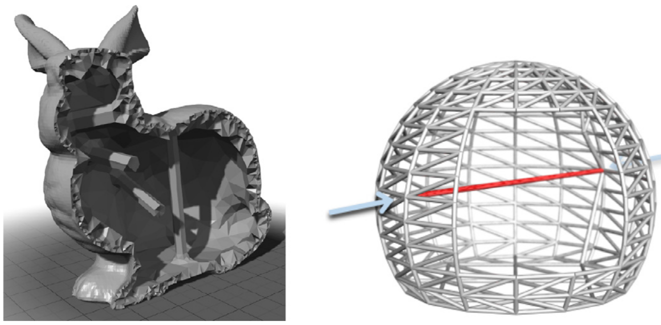


Fig. 1. Non-support-free inner struts. The left one is generated by Stava et al. [6] and right one is generated by Wang et al. [1] with two opposite forces.

surface or strut is impending if the angle θ (overhanging angle) between its outer normal $\mathbf{n}$ (a vector which is the outer normal of the surface containing the strut and perpendicular to this strut) and printing direction $\mathbf{d}$ is smaller than a threshold (the commonly used value is 45°), such as the green angle in the illustration, and a node is said to be impending if its z-coordinate is lower than the z-coordinates of the other end points of the edges that connect with it, such as the red node in the illustration.

A structure satisfies the support-free printing property if and only if it does not have impending surfaces, edges, nodes. As seen in Fig. 1, the left one and right one are generated by the algorithm in [6] and [1], respectively. They do not take into account the support-free printing property of the generated structures, so some impending struts might exist and many interior support structures will be needed during printing. In addition, the printing of additional support structures consumes not only printing time, but also material and energy. Furthermore, it is tough and sometimes even impossible to remove the additional support material inside the objects since most printed objects are closed volumes. Moreover, the existence of the additional interior support structures may violate the physical or geometrical properties. For example, if we plan to optimize the static stability of an object, although the optimized digital model can achieve our goal, the printed additional interior support structures will break the desired static stability.

As analysed above, it is necessary to take into account the support-free printing property of the model during the generation of lightweight structures, while meeting specific structural and geometrical constraints. Frame structure is one of the light-weight structures that used in [1]. They proposed an iterative optimization algorithm to generate skin-frame structures to reduce material cost, while achieving a given structural strength, printability, stability, and so on. However, they do not consider the support-free printing property of the generated inner struts. In this work, we mainly focus on the generation of support-free inner frame structures. After initialization, a sparsity optimization with support-free constraint is proposed to eliminate redundant and non-support-free inner struts, and angles between neighboring inner struts and radii of the rest of inner struts are further optimized to reduce printing material and meet a given structural strength. As a result, the proposed algorithm can guarantee that the generated inner struts not only are printed without interior support structure, but also meet a given structural strength.

The contributions of this paper are two folds:

- A support-free optimization framework is proposed to generate support-free inner frame structures.
- A struts elimination strategy is proposed to avoid impending nodes both on the inner and outer frame structures.

The remainder of this paper is organized as follows. In Section 2, we will briefly review articles related to our algorithm. Problem and constraints optimized in our formulation will be introduced in Section 3. The overall methodology of our support-free optimization framework will be introduced in Section 4. To demonstrate the practicability and effectiveness of the proposed algorithm, some experimental results will be shown in Section 5. Finally, conclusions, limitations and future work will be given in Section 6.

2. Related work

3D Printing. Recently, 3D printing has become a popular topic in Computer Graphics Community. Extensive algorithms have been presented to handle complex geometric problems in this field. Some of them attempt to design mechanical toy and automata [7,8], and articulated models [9,10]. The other algorithms are proposed to improve printing efficiency [11,12] and quality [13,14], and dynamic stability [15–18]. In contrast, our work is proposed to generated support-free frame structures with minimal material usage, while meeting a given structural strength, printability, and so on.

Interior structures. In computer graphics community, many efforts have been spent on designing light-weight structures to reduce material cost and maintain certain mechanical properties, such as structural soundness and stability, etc. The idea of cost-effective 3D printing has been introduced in Wang et al. [1]. To reduce printing material, skin-frame structure is designed inside of 3D model with proposed topology optimization and geometry optimization. At the same time, structural strength, static stability, and printability are optimized in their framework. Similarly, the work by Zhang et al. [4] 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. [2] and Sá et al. [19] divide the interior of the model with honeycomb-cell structure which is known to be of minimal material cost, while providing strength in tension. Wu et al. [5] present a method to generate bone-like porous structures as lightweight infill for additive manufacturing with topology optimization.

Since the structural strength is not considered during the obtainment of 3D models, so many models cannot be successfully printed or maybe damaged during daily usage and transportation. To enhance structural weak models, Stava et al. [6] propose three strategies to enhance the structural weak regions: insert additional struts, thicken weak parts, or hollow strong regions.

However, the above algorithms do not take into account the support-free printing property of the generated structures. As a result, many impending struts, nodes, or sub-structures maybe existed resulting in a large number additional internal supporting structures during printing, which not only damage the mechanics properties but also consume additional printing material, printing time, and energy. In contrast, we propose a support-free optimization framework to maker sure the generated frame structures are printed without interior support structures.

External support structure. To guarantee the integrity of the printed objects and stability of printing, additional outer supporting structures should be designed and added on the model. This procedure not only wastes printing material, but also consumes printing time and energy. To effectively solve this issue, many researches have been devoted to reducing the supporting structures with different strategies.

Vanek et al. [20] present a novel, geometry-based approach to minimize the supporting material while providing sufficient support. In their method, they need to orient the model into a position with minimal impending area and the points need to be supported are detected. Finally, a tree-like structure is used to design outer supporting structures. Under a given printing direction, Hu et al. [21] propose an algorithm to partition 3D models into approximate pyramidal parts to largely reduce outer supporting structures, as pyramidal shape does not need supporting structures at all. Hu et al. [22] deform the input shape to reduce impending area to effectively reduce outer supporting structures. Scaffoldings structures [3] and branching structures [23] are used to design outer supporting structures, respectively. Their algorithms are able to reduce a large number outer supporting structures.

However, all algorithms described above do not consider the interior support structures.

Interior support structure. Many 3D printing researches need to generate holes inside the model, such as printing material reduction [2], stability optimization [24], etc. However, these algorithms do not guarantee the generated holes can be printed without inner supporting structures, resulting in the damage of geometrical and structural properties. To handel this problem, Wei et al. [25] partition the model into several shell parts so that each part can be printed without inner supporting structures. Although the parts generated with their algorithm do not need inner supporting structures, the parts should be glued together after printed which will generate clear gaps between neighboring parts. To overcome this issue, Wu et al. [26] propose an algorithm to fill the inside of a model with self-supporting rhombic structures. Similarly to [26], we propose an algorithm to generate support-free inner frame structures to avoid interior support structures while meeting a given structural strength and geometrical constraints.

Topology optimization. Nowadays, topology optimization algorithms have be widely used in 3D printing to generate structural integrity and soundness structures. Specifically, some researchers propose topology optimization frameworks by taking into account the supporting structure. Amir et al. [27] propose a sensitivity-based topology optimization framework which assumes a continuous dependence of support volume on boundary or topology perturbation. Although the proposed algorithm can significantly reduce support structures, they cannot totally avoid supporting structures. To achieve the above goal, Langelaar [28] proposes a new filter for density-based topology optimization targeting fundamental geometrical printability aspects.

Self-supporting structures. Nowadays, many researcher propose algorithms to design self-supporting surfaces or structures. Vouga et al. [29] use the thrust network method of analysis and present an iterative nonlinear optimization algorithm for efficiently approximating freeform shapes by self-supporting masonry. Panozzo et al. [30] propose an algorithm to find self-supported masonry structure that is as close as possible to the given target surface by optimizing the force layouts both geometrically and topologically. Goes et al. [31] present a novel approach for the analysis and design of self-supporting simplicial masonry structures. Liu et al. [32] propose an algorithm to design masonry structures by regular triangulation. Tang et al. [33] propose a too to design self-supporting meshes which can be extended to treat the static equilibrium of the shapes with overhanging parts. Deuss et al. [34] present a method to gradually construct the masonry model in stable sections and drastically reduces the material requirements and construction costs.

However, most of them are not related to the fabrication with 3D printing technologies. Therefore, the self-supporting property considered in their algorithms is not the same as support-free printing property described in our paper.

3. Problem and formulation

3.1. Problem and notations

Given an input mesh $\mathcal{M}$, our goal is to generate a frame structure $\mathcal{H}$ to represent $\mathcal{M}$ so that the inner struts of $\mathcal{H}$ can be printed without any additional interior support at the cost of the minimal material, while considering the constraints of mechanical properties and the geometric similarity as well.

The frame structure $\mathcal{H}$, used in [1] as lightweight structure, is composed of an outer structure $\mathcal{O}_H$ and an inner structure $\mathcal{I}_{\mathcal{H}}$, that is $\mathcal{H} = \mathcal{O}_{\mathcal{H}} \cup \mathcal{I}_{\mathcal{H}}$. The outer structure $\mathcal{O}_{\mathcal{H}}$ is generated on the offset mesh $\mathcal{M}'$ with a distance h_S to the input mesh $\mathcal{M}$, where $\mathcal{M}'$ is obtained with the algorithm presented in [35]. Each structure is composed of a spherical nodes set $\mathbf{V} = \mathbf{V}_{int} \cup \mathbf{V}_{out} = \{\mathbf{v}_i, i = 1, 2, \ldots, |V|\}$ which are located on or inside the offset mesh $\mathcal{M}'$, and a cylindrical struts set $E = E_{int} \cup E_{out} = \{e_j, j = 1, 2, \ldots, |E|\}$ with radius r_j and length l_j for each strut $e_j \in E$. Fig. 3 illustrates the notations described above.

As the generated frame structures maybe covered by a skin structure, so the sets of $\mathbf{V}_{out}$ and E_{out} are fixed during our optimization, and only the radii of E_{out} are optimized.

3.2. Overview

The pipeline of our algorithm is shown in Fig. 2. For a given triangular mesh $\mathcal{M}$ (Fig. 2(a)), we generate its offset version $\mathcal{M}'$ [35] with distance h_S which is the printable thickness of the current 3D printer (Fig. 2(b)). We randomly sample a large number of nodes inside $\mathcal{M}'$ as inner nodes, and connect k-nearest nodes to generate inner struts whose radii are initialized to provide a prior for the following optimization (Fig. 2(c)). Thirdly, sparsity optimization framework with support-free constraint is proposed to eliminate non-support-free struts and structural redundant struts (Fig. 2(d)). Finally, overhanging angle optimization is proposed to further reduce material usage with overhanging angle constraint (Fig. 2(e)). The details of each step will be introduced in the following.

3.3. Constraints

Before presenting our optimization formulation, we will first introduce several geometrical and structural constraints integrated in our formulation.

3.3.1. Structural constraints

In the following, we will only briefly introduce several structural constraints optimized in our formulation, and the interested readers are recommended to [1] for more details.

Stiffness. In this paper, beam based finite element method (FEM) is used to calculate the stiffness of the frame structure:

$$\mathbf{K}(\mathbf{V}, \mathbf{r})\mathbf{D} = \mathbf{F}(\mathbf{r}), \tag{1}$$

where, $\mathbf{K}(\mathbf{V}, \mathbf{r})$ is a stiffness matrix depending on nodal position $\mathbf{V}$ and strut radii $\mathbf{r}$, $\mathbf{F}(\mathbf{r}) = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_{|V|}\}$ are the (internal and external) forces acting on the nodes, and $\mathbf{D} = \{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_{|V|}\}$ are the deformations of the nodes.

Elastic property. For a strut, its shear stress and axis stress are defined as follows:

$$\frac{|\mathbf{e}^T\mathbf{d}_\mathbf{e}|}{\|\mathbf{e}\|^2}\gamma \le \sigma, \quad \mathbf{e} \in E, \tag{2}$$

$$\frac{\|\mathbf{e}\|^2\mathbf{d}_\mathbf{e} - (\mathbf{e}^T\mathbf{d}_\mathbf{e})\mathbf{e}}{\|\mathbf{e}\|}\mu \le \tau, \quad \mathbf{e} \in E, \tag{3}$$

where, $\mathbf{d}_\mathbf{e} = \mathbf{d}_{i_2} - \mathbf{d}_{i_1}$, $\mathbf{e} = \mathbf{v}_{i_2}\mathbf{v}_{i_1}$, $\mathbf{v}_{i_1}$ and $\mathbf{v}_{i_2}$ are two end points of $\mathbf{e}$, σ, γ, μ, and τ represent tensile (or compression) strength, tensile modulus, shear modulus, and shear strength of the printing material, respectively.

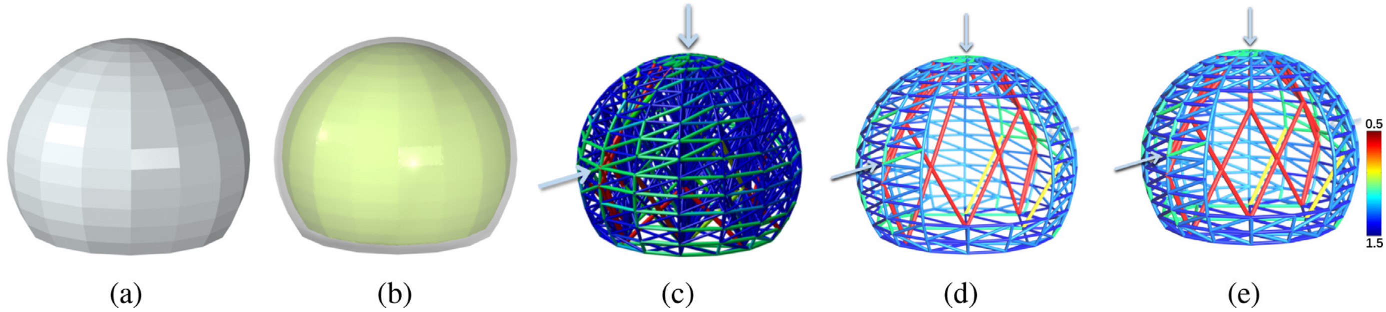


Fig. 2. Pipeline of the proposed algorithm. Input a triangular mesh. (a) Support-free inner frame structure is generated in its offset version. (b) We randomly generate a large number of inner struts and nodes and initialize the radii of all struts. (c) All redundant struts and non-support-free struts are eliminated. (d) The inner struts and nodes are further optimized to reduce material usage. (e) This model is generated with three forces: one on the top (15 N) and the other two are opposite. (15 N) Color in these subfigures are used to visualize the radii of all struts. Some front outer struts are removed to clearly see the inner struts.

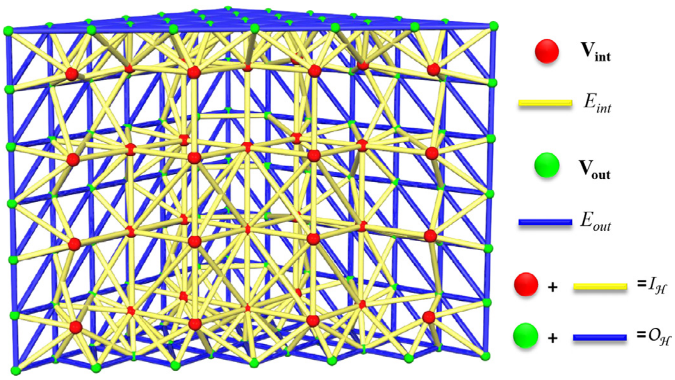


Fig. 3. The illustration of some notations used in this work.

Buckling. All struts are subject to buckling constraints according to

$$r_j \geq \frac{l_j}{\alpha}, \quad \mathbf{e}_j \in E_{out} \cup E_{int} \tag{4}$$

where, α is the slenderness radio.

3.3.2. Support-free constraints

To achieve the support-free printing property of struts, we design the following constraint:

$$(\cos\theta_j - \cos\Theta).r_j < \varepsilon, \quad \mathbf{e}_j \in E_{int} \tag{5}$$

Once all inner struts satisfy support-free printing property, we can further optimize the positions of inner nodes to reduce material usage under the following overhanging angle constraint:

$$\cos\theta_j \leq \cos\Theta, \quad \mathbf{e}_j \in E_{int} \tag{6}$$

where, θ_j in Eqs. (5) and (6) represents the overhanging angle of the jth inner strut, Θ in Eqs. (5) and (6) is a parameter to control support-free printing property and the shape of the final struts, which is set to 45° in our paper.

In sparsity optimization (Section 4.2.1), we only optimize the radii of all struts, that is overhanging angle θ_js are fixed. For Eq. (5), if $\theta_j < \Theta$ (impending), that is $\cos\theta_j - \cos\Theta > 0$, to satisfy constraint (5), r_j must approach ε. In contrast, constraint (5) is naturally satisfied when $\theta_j \geq \Theta$. In this way, the radii of non-support-free struts will become very small after sparsity optimization. In overhanging angle optimization (Section 4.2.2), the node positions are optimized. Therefore, we should add overhanging angle constraint (Eq. (6)) for inner struts during optimization to guarantee their support-free property, that is the overhanging angle must be larger than Θ.

3.3.3. Geometrical similarity constraint

To maintain the geometry details and the appearance of the model, the frame structure should not have large deformations, that is:

$$\|\mathbf{d}_i\| \leq \varepsilon, \quad i = 1, 2, \ldots, |V| \tag{7}$$

3.3.4. Printability

To ensure the generated struts are manufacturable, the lower bounds of strut radii are set as $\underline{\eta}$ for a particular 3D printer. In addition, the upper bounds for outer struts are set as h_S to avoid lying outside $\mathcal{M}$ if it is covered by $\mathcal{M}$, i.e.,

$$\underline{\eta} \leq r_j \leq h_S, \quad e_j \in E_{out} \tag{8}$$

and the radii of the inner struts can be set with a larger upper bound $\bar{\eta}$, i.e.,

$$\underline{\eta} \leq r_j \leq \bar{\eta}, \quad e_j \in E_{int} \tag{9}$$

4. Our algorithm

As shown in Fig. 2, our algorithm is composed of two major steps: Initialization and support-free optimization which contains sparsity optimization and overhanging angle optimization. The details of these steps will be introduced in the following sections.

4.1. Initialization

For outer frame structure, its nodes are the points of $\mathcal{M}'$ with radius c, and its struts generated by replacing the edges of $\mathcal{M}'$ with cylinders whose radius are r. For inner nodes, a user-specified number is used to evenly sample inside the solid volume enclosed by $\mathcal{M}'$. The inner frame struts of inner frame structure $\mathcal{T}_{\mathcal{I}}$ is generated by connecting k-nearest neighboring sampling nodes via the ANN algorithm [36], where k is set to 15 in our experiments.

The outer struts initialized in the above way may not be totally support-free. To handle this issue, strut based supports design method described in [1] is used to design supports for outer frame structure. The supports located inside of inner frame structure are seen as the struts of inner frame structure, and optimized with the proposed support-free optimization framework. In this way, we just need to optimize the support-free printing property of inner frame structure.

To provide a reasonable solution for support-free optimization, we use the size optimization described in [1] to obtain the initial radii and volume which are denoted as $\tilde{\mathbf{r}}$ and $Vol(\tilde{\mathbf{r}}, \mathbf{V}, E)$, respectively.

4.2. Support-free optimization

In this section, a support-free optimization framework is proposed to generate support-free inner frame structure, which contains two steps: sparsity optimization and overhanging angle optimization. Sparsity optimization is used to eliminate the redundant inner struts and the struts which do not satisfy support-free

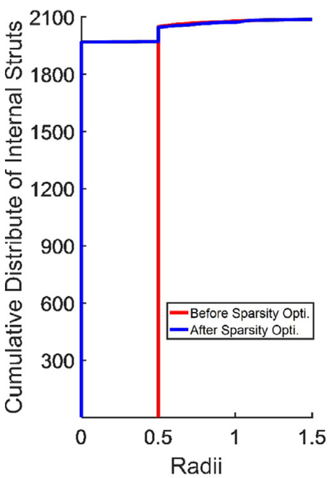


Fig. 4. Distributions of strut radii and angle between neighboring inner struts before and after sparsity optimization on the Semi-Sphere model shown in Fig. 2. Left: the cumulative distributions of strut radii before and after sparsity optimization. Right: the cumulative distributions of overhanging angles of internal struts before and after sparsity optimization. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

constraint. In addition, a struts' elimination strategy is proposed to design supports with inner struts for all of the outer and inner impending nodes. Overhanging angle optimization is applied to further optimize the shape of inner frame structures to reduce material usage.

4.2.1. Sparsity optimization

The left red curve of Fig. 4 shows the result of Semi-Sphere model after initialization. It is obviously seen that many struts of inner frame structure $\mathcal{I}_{\mathcal{H}}$ have negligible influence on the structural strength as their radii approach the printable lower bound of the printer, which will waste many material if it is printed. Furthermore, many struts do not satisfy support-free printing property, see the red curve in the right of Fig. 4. Therefore, the inner frame structure should be optimized to eliminate redundant struts and non-support-free struts to reduce material usage and generate support-free inner frame structure.

We know that if the radius of a strut becomes or approximates to zero, it is vanished. Thus the number of struts is equal to the number of non-zero strut radii, i.e., $|E_{int}| = \|\mathbf{r}_{int}\|_0$. Therefore, we propose the following ℓ_0 sparsity optimization to eliminate redundant and non-support-free struts and achieve simplicity of the inner frame structure:

$$\begin{aligned} \min_{\mathbf{r}} \quad & |E_{int}| = \|\mathbf{r}_{int}\|_0 \\ s.t. \quad & \{(1-4), (6), (8-10)\} \end{aligned} \tag{10}$$

As our major goal is to reduce printing material, so a volume constraint should be added in formulation (10):

$$Vol(\mathbf{r}, \mathbf{V}, E) \le Vol(\widetilde{\mathbf{r}}, \mathbf{V}, E) \tag{11}$$

where, $Vol(\widetilde{\mathbf{r}}, \mathbf{V}, E)$ is obtained from initialization step. Formulation (10) can be approximately solved with a reweighting strategy presented in [1].

As we can see from the blue curve of Fig. 4(left), the radii of a large number of inner struts approach zero after sparsity optimization. Therefore, we are able to simply eliminate these struts with a threshold ζ, that is:

$$\widehat{E}_{int} = \{\mathbf{e}_j \in E_{int} \mid \widehat{r}_j \ge \zeta\} \tag{12}$$

where, ζ is set to 10^{-5} mm in our implementation. After struts elimination, the overhanging angle of all struts are larger than 45° (blue curve of Fig. 4(right)), that is all struts are support-free.

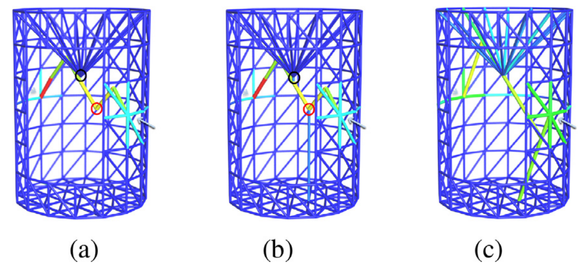


Fig. 5. The node marked with red circle is a impending point exposed after the optimization process and the node marked with black circle is a impending point on the outer frame structure. (a) The new-added struts (b) can support these impending nodes and make the model manufacturable. All inner struts and nodes are further optimized to achieve minimal material usage (c). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

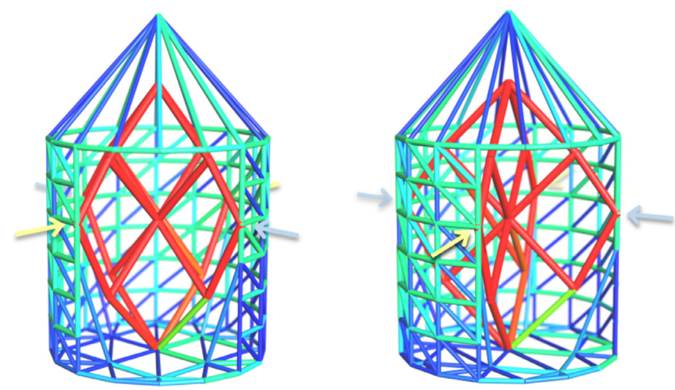


Fig. 6. Two different views of the optimized Bird-Cage model. The Bird-Cage model are generated with two pairs of opposite forces. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

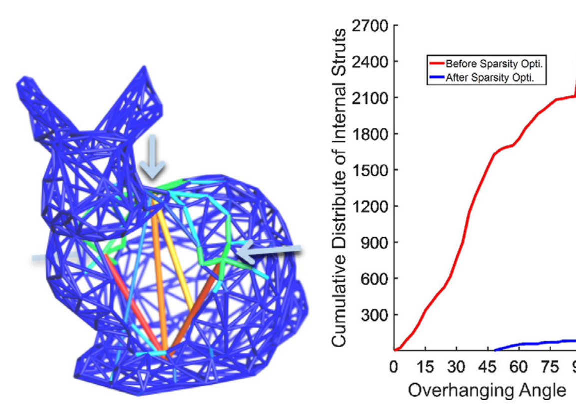


Fig. 7. Bunny model is optimized with three forces in different directions (see the light blue arrows) by the proposed algorithm. As we can see that all inner struts are support-free. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 5(a) shows a frame structure whose non-support-free struts and redundant struts have been eliminated according to Eq. (12). We can see from the red circle that some inner node is impending which will need support structures during printing. In addition, some node of outer frame structure maybe also impending (black circle).

To prevent the occurrence of these kinds of impending nodes, we propose the following strategy.

- We detect the inner struts whose radii are larger than ζ and analyse the nodes connecting with these struts.

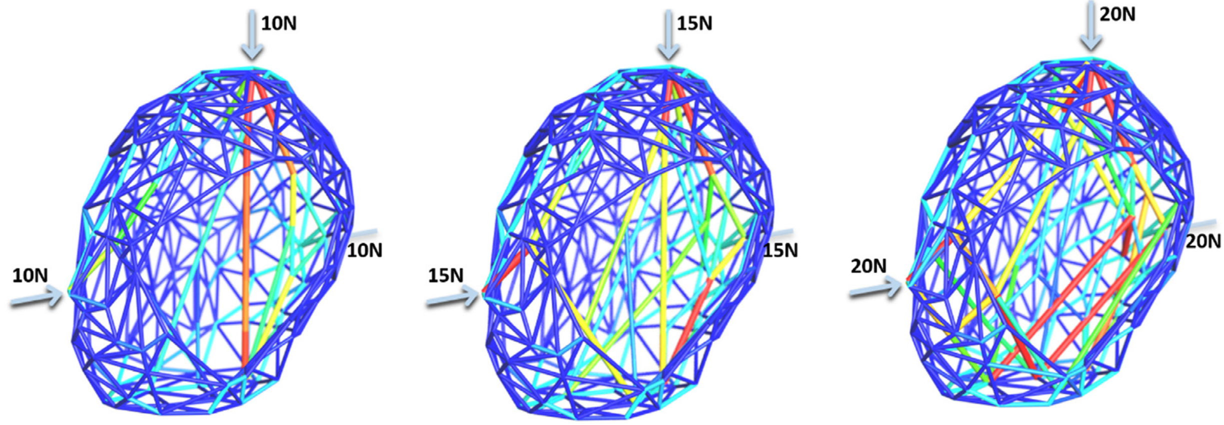


Fig. 8. Buddha model is optimized with three different size of forces on the same regions: 10 N(left), 15 N(middle), and 20 N(right), respectively. Each model is optimized with force in three different directions (arrows). The number of inner struts and their radii are increased with the increase of forces.

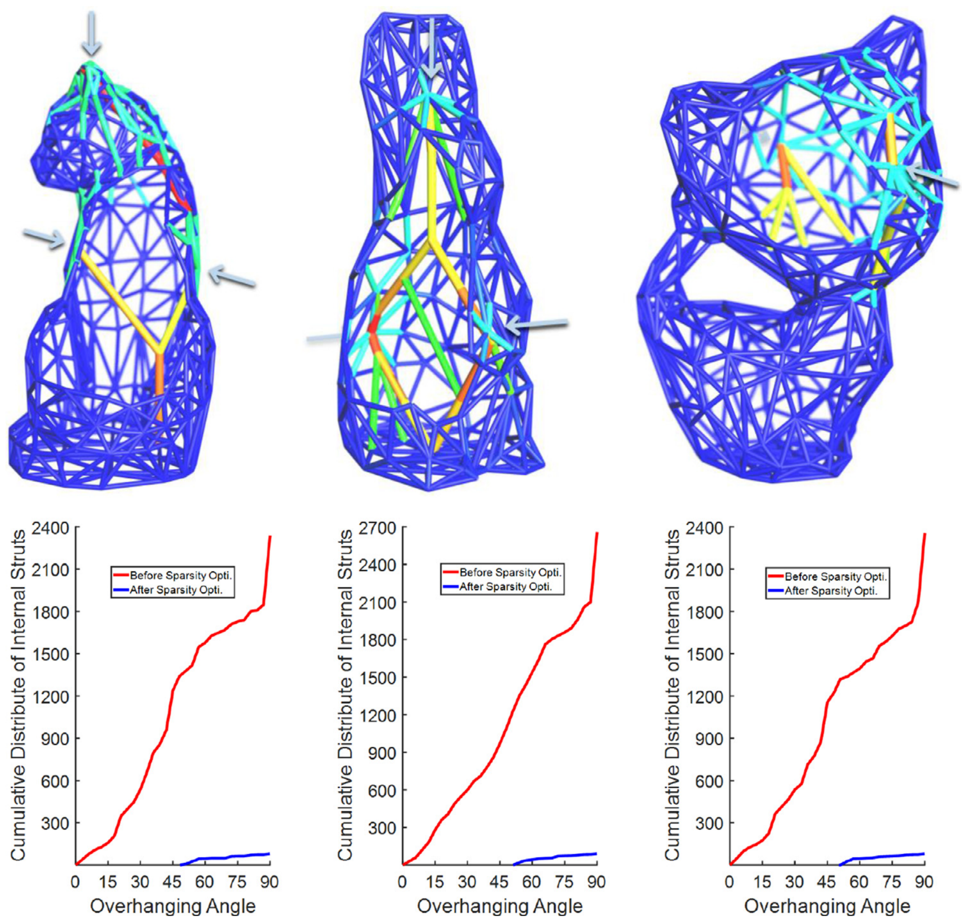


Fig. 9. More frame structures generated with our algorithm. The first row shows the final frame structures. The second row shows the distribution of overhanging angles before and after sparsity optimization.

- If a node p is impending, we start from p to find a strut which is support-free and its the other end node q is below of p.
- The radius of the found strut is set as a large value, and we check whether q is impending.
- If so, we back to the step 2, otherwise the process is stopped.

The above strategy is iteratively implemented until no impending inner nodes. For impending nodes on outer frame structure, we implement the same procedure described above to find their supports among inner struts.

As seen in Fig. 5(b), some inner struts are added for these impending nodes which are further optimized with overhanging angle optimization (Section 4.2.2). The final result is shown in Fig. 5(c).

4.2.2. Overhanging angle optimization

After sparsity optimization and struts' elimination, the radii of many struts are not optimal which will waste some printing material. Furthermore, the angles between inner struts can be optimized to obtain an optimal inner support-free frame structure.

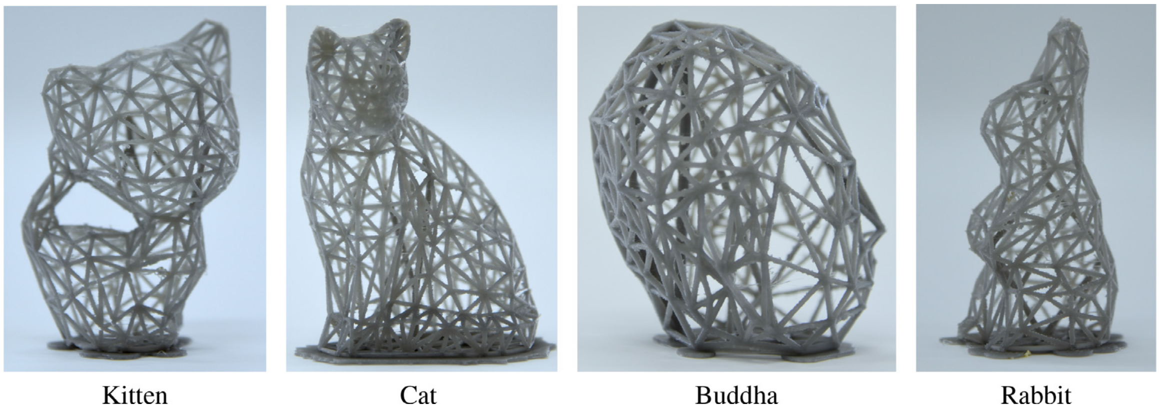


Fig. 10. Printed objects generated with our algorithm.

Table 1
Statistics of Our approach. The third column shows the number of struts (outer and inner) after initialization, the fourth column shows the number of final struts (outer and inner), the fifth column shows the solid volume, the sixth column shows the volume of final struts (outer and inner), and the last column lists the running time of models whose unit is minute.

Model	Figure	#Struts (Before opt.)	#Struts (After opt.)	Volume solid ($10^4 mm^3$)	Struts volume opt. ($10^3 mm^3$)	Timing (min.)
Semi-Sphere	Fig. 2	2087	762	92.43	18.29	28.8
Bird-Cage	Fig. 6	2494	623	41.89	13.57	39.2
Bunny	Fig. 7	2489	806	30.58	5.54	37.4
Buddha	Fig. 8(left)	2313	641	34.61	7.09	31.4
Cat	Fig. 9(left)	2338	1032	26.43	6.24	34.3
Rabbit	Fig. 9(middle)	2660	488	8.11	3.52	43.9
Kitten	Fig. 9(right)	2355	795	12.17	4.54	36.2

Therefore, we proposed the following overhanging angle optimization framework to optimize the geometric positions of the inner nodes and the radii of struts with overhanging angle constraint defined in Eq. (6):

$$\begin{aligned} &\min_{\mathbf{r},\mathbf{V}_{int}} \quad Vol(\mathbf{r},\mathbf{V},\widehat{E}) \\ &s.t. \quad \{(1-4),(7-10)\} \end{aligned} \tag{13}$$

Finally, we get an optimized frame structure $\mathcal{T}^* = \{\mathbf{V}^*, E^*\}$. The final result of semi-sphere is shown in Fig. 2(d).

5. Implementation and results

There are several parameters in our formulation. η and $\bar{\eta}$ are set as 0.5 mm and 1.5 mm, respectively. ε in Eq. (7) is set to 0.05 mm. h_S in Eq. (8) is set to 0.8 mm. The height of all models optimized in our paper is scaled to 100 mm.

The whole optimization formulation was implemented with mixed MATLAB and C++, and it was run on a PC with Intel(R) Core(TM) i7-3770 CPU @ 3.40 GHz and 32 GB memory. For the constrained nonlinear optimization problem, Matlab solver fmincon is used to solve it which is based on interior point theory.

To evaluate the effectiveness and robustness of the proposed algorithm, we test our method on several 3D models. All frame structures generated with our algorithm can withstand a certain external forces and can be printed without any interior support structures. To clearly see the generated inner frame structure, we cut some front struts of outer frame structure.

In Fig. 6, Bird-Cage model is optimized with four forces (yellow arrows and light blue arrows) and the result is shown in two different views: front (Fig. 6(left)) and one side (Fig. 6(right)). From this figure we can see that the generated inner frame structure with the proposed optimization framework can withstand a given structural strength and meet support-free printing property, that is all overhanging angles of inner struts are larger than 45°.

As shown in the left of Fig. 1, Stave et al. insert additional struts to increase the structural strength of the Bunny model. But some of the added struts are impending which need additional support structures to help print. In addition, the frame structure generated with [1] under a finger grasping force for Semi-sphere model has an impending inner strut which also cannot be printed without support structures. However, the frame structure optimized with our algorithm for these models are support-free, see Figs. 2 and 7.

In Fig. 8, we show the frame structures generated with three different forces: 10N, 15N, and 20N, respectively. From this figure we can see that the number of inner struts and their radii are increased with the increase of forces. However, all inner struts are support-free.

Fig. 9 shows more frame structures generated with our optimization framework. The first row shows the final frame structures. The second row shows the distribution of overhanging angles before and after sparsity optimization. This figure clearly shows that all inner struts generated with our algorithm are support-free.

Fig. 10 shows some of the printed objects that generated with our method. All objects are printed with Makerbot 2. The printing material is grey PLA and the slicing thickness is 0.2 mm. Table 1 lists the number of struts before and after optimization, volume of the initial model and optimized frame structure, and the running time of all models.

6. Conclusions and future work

In this work, we propose an automatic and practical method to generate support-free frame structure for a given 3D model. Support-free constraints are designed according to the angle between printing direction and strut normals. A sparsity optimization is proposed to eliminate redundant and non-support-free struts and overhanging angle optimization is proposed further optimize the shape of struts for the purpose of material reduction. In addition, an elimination strategy is designed to avoid impending inner and outer nodes.

However, the proposed algorithm still has several limitations. First of all, we do not optimize the frame structures with arbitrary external forces which is often insufficient to expose the true structure weakness. Secondly, we only take into account the internal support structure, since outer support structures are always easy to be cleaned. Finally, our algorithm is mainly suitable for fused deposition printing technology, such as FDM.

Acknowledgments

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, 61562062, 61432003, 61672482, 11626253, 11472073).

References

[1] Wang W, Wang TY, Yang Z, Liu L, Tong X, Tong W, et al. Cost-effective printing of 3d objects with skin-frame structures. ACM Trans Gr (TOG) 2013;32(6):177.
[2] Lu L, Sharf A, Zhao H, Wei Y, Fan Q, Chen X, et al. Build-to-last: strength to weight 3d printed objects. ACM Trans Gr (TOG) 2014;33(4):97.
[3] Schmidt R, Umetani N. Branching support structures for 3d printing. In: Proceedings of the ACM SIGGRAPH 2014 Studio. ACM; 2014. p. 9.
[4] Zhang X, Xia Y, Wang J, Yang Z, Tu C, Wang W. Medial axis tree an internal supporting structure for 3d printing. Comput Aided Geom Des 2015;35:149–62.
[5] Wu J, Aage N, Westermann R, Sigmund O. Infill Optimization for Additive Manufacturing – Approaching Bone-like Porous Structures. IEEE Trans Visual Comput Graphics 2017;PP(99):1–14.
[6] Stava O, Vanek J, Benes B, Carr N, Měch R. Stress relief: improving structural strength of 3d printable objects. ACM Trans Gr (TOG) 2012;31(4):48.
[7] Zhu L, Xu W, Snyder J, Liu Y, Wang G, Guo B. Motion-guided mechanical toy modeling. ACM Trans Gr 2012;31(6):127.
[8] Ceylan D, Li W, Mitra NJ, Agrawala M, Pauly M. Designing and fabricating mechanical automata from mocap sequences. ACM Trans Gr (TOG) 2013;32(6):186.
[9] Bächer M, Bickel B, James DL, Pfister H. Fabricating articulated characters from skinned meshes. ACM Trans Gr (TOG) 2012;31(4):47.
[10] Sun T, Zheng C, Zhang Y, Yin C, Zheng C, Zhou K, et al. Computational design of twisty joints and puzzles. ACM Trans Gr (TOG) 2015;34(4):101.
[11] Vanek J, Galicia J, 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. 322–32.
[12] Wang W, Chao H, Tong J, Yang Z, Tong X, Li H, et al. Saliency-preserving slicing optimization for effective 3d printing. In: Computer Graphics Forum, vol. 34. Wiley Online Library; 2015. p. 148–60.
[13] Hildebrand K, Bickel B, Alexa M. Orthogonal slicing for additive manufacturing. Comput Gr 2013;37(6):669–75.
[14] Wang WM, Zanni C, Kobbelt L. Improved surface quality in 3d printing by optimizing the printing direction. In: Computer Graphics Forum, vol. 35. Wiley Online Library; 2016. p. 59–70.
[15] Bächer M, Whiting E, Bickel B, Sorkine-Hornung O. Spin-it: optimizing moment of inertia for spinnable objects. ACM Trans Gr (TOG) 2014;33(4):96.
[16] Musialski P, Auzinger T, Birsak M, Wimmer M, Kobbelt L. Reduced-order shape optimization using offset surfaces. ACM Trans Gr 2015;34(4):102.
[17] Musialski P, Hafner C, Rist F, Birsak M, Wimmer M, Kobbelt L. Non-linear shape optimization using local subspace projections. ACM Trans Gr 2016;35(4).
[18] Wang L, Whiting E. Buoyancy optimization for computational fabrication. In: Computer Graphics Forum, vol. 35. Wiley Online Library; 2016. p. 49–58.
[19] Sá ME, Mello VM, Rodriguez Echavarria K, Covill D, et al. Adaptive voids. Vis Comput Int J Comput Gr 2015;31(6–8):799–808.
[20] Vanek J, Galicia JA, Benes B. Clever support: efficient support structure generation for digital fabrication. In: Computer graphics forum, vol. 33. Wiley Online Library; 2014. p. 117–25.
[21] Hu K, Jin S, Wang CC. Support slimming for single material based additive manufacturing. Comput Aided Des 2015;65:1–10.
[22] Hu K, Zhang X, Wang CC. Direct computation of minimal rotation for support slimming. In: Proceedings of the IEEE International Conference on Automation Science and Engineering (CASE). IEEE; 2015. p. 936–41.
[23] Dumas J, Hergel J, Lefebvre S. Bridging the gap: automated steady scaffoldings for 3d printing. ACM Trans Gr (TOG) 2014;33(4):98.
[24] Musialski P, Hafner C, Rist F, Birsak M, Wimmer M, Kobbelt L. Non-linear shape optimization using local subspace projections. ACM Trans Gr (TOG) 2016;35(4):87.
[25] Wei X-R, Zhang Y-H, Geng G-H. No-infill 3d printing. 3D Res 2016;7(3):24.
[26] Wu J, Wang CC, Zhang X, Westermann R. Self-supporting rhombic infill structures for additive manufacturing. Comput Aided Des 2016;80:32–42.
[27] Mirzendehdel AM, Suresh K. Support structure constrained topology optimization for additive manufacturing. Comput Aided Des 2016;81:1–13.
[28] Langelaar M. An additive manufacturing filter for topology optimization of print-ready designs. Struct Multidiscip Optim 2017;55(3):871–83.
[29] Vouga E, Höbinger M, Wallner J, Pottmann H. Design of self-supporting surfaces. ACM Trans Gr (TOG) 2012;31(4):87.
[30] Panozzo D, Block P, Sorkine-Hornung O. Designing unreinforced masonry models. ACM Trans Gr (TOG) 2013;32(4):91.
[31] De Goes F, Alliez P, Owhadi H, Desbrun M. On the equilibrium of simplicial masonry structures. ACM Trans Gr (TOG) 2013;32(4):93.
[32] Liu Y, Pan H, Snyder J, Wang W, Guo B. Computing self-supporting surfaces by regular triangulation. ACM Trans Gr (TOG) 2013;32(4):92.
[33] Tang C, Sun X, Gomes A, Wallner J, Pottmann H. Form-finding with polyhedral meshes made simple. ACM Trans Gr 2014;33(4):70–1.
[34] Deuss M, Panozzo D, Whiting E, Liu Y, Block P, Sorkine-Hornung O, et al. Assembling self-supporting structures. ACM Trans Gr (TOG) 2014;33(6):214.
[35] CL Wang C, Chen Y. Thickening freeform surfaces for solid fabrication. Rapid Prototyp J 2013;19(6):395–406.
[36] Arya S, Mount DM, Netanyahu NS, Silverman R, Wu AY. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. J ACM (JACM) 1998;45(6):891–923.