

# Cost-effective Printing of 3D Objects with Skin-Frame Structures

Weiming Wang<sup>†,‡</sup> Tuanfeng Y. Wang<sup>†</sup> Zhouwang Yang<sup>†\*</sup> Ligang Liu<sup>†</sup> Xin Tong<sup>§</sup>  
Weihua Tong<sup>†</sup> Jiansong Deng<sup>†</sup> Falai Chen<sup>†</sup> Xiuping Liu<sup>‡</sup>

<sup>†</sup>University of Science and Technology of China <sup>‡</sup>Dalian University of Technology <sup>§</sup>Microsoft Research Asia



**Figure 1:** Given an input Horse model (a), our method generates a skin-frame structure (b), which is approximate to the model, to minimize the cost of material used in printing it. The frame structure is designed to meet various constraints by an optimization scheme. In (b) we remove the front part of the skin in order to show the internal structure of frame. (c) is the photo of an printed model by removing part of its skin to see the internal struts. (d) is the photo of the printed model generated by our method. A small red drawing pin is put under the object as a size reference in (c) and (d) respectively. The material usage in (d) is only 15.0% of that of a solid object.

## Abstract

3D printers have become popular in recent years and enable fabrication of custom objects for home users. However, the cost of the material used in printing remains high. In this paper, we present an automatic solution to design a skin-frame structure for the purpose of reducing the material cost in printing a given 3D object. The frame structure is designed by an optimization scheme which significantly reduces material volume and is guaranteed to be physically stable, geometrically approximate, and printable. Furthermore, the number of struts is minimized by solving an  $\ell_0$  sparsity optimization. We formulate it as a multi-objective programming problem and an iterative extension of the preemptive algorithm is developed to find a compromise solution. We demonstrate the applicability and practicability of our solution by printing various objects using both powder-type and extrusion-type 3D printers. Our method is shown to be more cost-effective than previous works.

**CR Categories:** I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling; I.3.8 [Computer Graphics]: Applications;

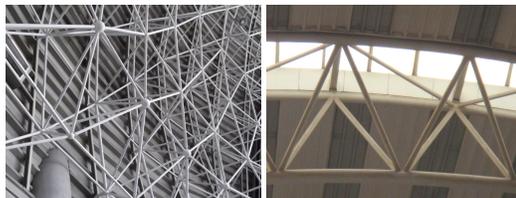
**Keywords:** 3D printing, fabrication, frame structure, sparsity optimization

\*Corresponding author: yangzw@ustc.edu.cn (Zhouwang Yang)

## 1 Introduction

Additive manufacturing (3D printing) enables fabrication of physical objects from digital models where the printed objects are created by laying down successive layers of material [3DSystems 2012; Shapeways 2012]. During the last few years, research on computational techniques of 3D printing has received considerable attention for assisting users to generate desired manufacturable objects [S-tava et al. 2012; Luo et al. 2012; Chen et al. 2013; Prévost et al. 2013].

However, reducing the material used in printing, which is an important problem due to its high cost, has not been well studied. The straightforward approach used in commercial printer packages [Shapeways 2012] is to uniformly hollow the 3D object by extruding the outer surface and creating a scaled-down version on its inside. The user has to choose a scaling factor (thickness of object) based on experience. A large factor may lead to material waste while a small factor could cause structural stability problem. Thus it is technically nontrivial for the hollowing method to simultaneously match the goals of saving material and maintaining physical stability in 3D printing.



**Figure 2:** Common frame structures used in architecture.

In this paper, we present an automatic method to minimize material cost of the object in 3D fabrication. The key idea is to ‘hollow’ the object by creating a lightweight frame structure (see Figure 2), made of a mesh of nodes and thin cylindrical struts with large voids among them inside the object (see Figure 1 (b)). Frame structures benefit 3D printing in two aspects. First, the mass of object could be significantly reduced through the use of frame structures while maintaining its strength and stiffness [Gibson and Ashby 1999].

Second, frame structures provide sufficient flexibility and variability, which make them possible to meet a variety of constraints in 3D printing. We develop an optimization scheme to minimize the frame volume subject to various constraints such as stiffness, stability, geometrical approximation, self-balance, and printability.

In order to eliminate redundant struts with negligible influence on the whole frame, the frame is also expected to have simple topology structure with minimal number of struts. The basic combinatorial nature of topology design, i.e., finding the optimal set of frame struts, which remains in structural optimization problems, has been proved to be NP-hard. Heuristic approaches such as genetic algorithms [Rajeev and Krishnamoorthy 1997] have been applied for structural optimization problems with stress and buckling constraints. However, these fairly general approaches are restricted to fairly small scale problems and are not suitable for 3D printing.

Different from previous approaches, we convert the minimization of the number of struts (structure redundancy) into an  $\ell_0$  sparsity optimization of the radius vector of struts, which can be approximately solved by a reweighted  $\ell_1$ -norm optimization. Also we cannot expect to find the globally optimal solution of achieving minimal frame volume and minimal strut number simultaneously. To this end, we formulate the problem as a multi-objective programming problem subject to a variety of constraints, and develop an iterative extension of the preemptive algorithm to find a compromise solution. The algorithm starts by minimizing frame volume and then optimizes structure redundancy without increasing the frame volume. The above two optimizations are iterated until the volume of frame does not decrease.

The frame structure generated by our algorithm is guaranteed to be physically stable, geometrically approximate, and printable (see Figure 1 (d)), which is cost-effective for printing. We have applied our algorithm on a variety of different models and successfully printed them using both powder-type printer and extrusion-type printer. The results show that our algorithm reduces material around 70% (from 66.3% to 92.1% in our experiments) with respect to the solid printing and achieves much better cost-effectiveness ratios than previous works.

**Contributions** Our contributions are summarized as follows.

- We propose a novel approach for automatically generating a skin-frame structure to minimize material usage in 3D printing, taking into account structural stability, geometrical approximation, self-balance, and printability of the object.
- An  $\ell_0$  sparsity optimization is adopted to automatically removing redundant struts with negligible influence on the whole frame.
- We formulate the problem of minimizing the frame volume and the structural redundancy as a multi-objective programming problem and develop an iterative extension of the preemptive algorithm to find a compromise solution.

## 2 Related Work

**Geometry processing for 3D fabrication** With 3D printers being more accessible to everyday users due to their rapid drop in price, more and more attention has been paid to fabrication techniques and applications. Quite a few topics in 3D fabrication, including structural analysis [Stava et al. 2012; Zhou et al. 2013], segmentation [Luo et al. 2012], articulated model generation [Bächer et al. 2012; Cali et al. 2012], and balancing shape [Prévost et al. 2013], have been researched in computer graphics recently. Based on finite element analysis of stress, the approach of [Stava et al.

2012] can strengthen printed objects by hollowing, local thickening, and adding extra struts. Different from the goal of our work, it does not always reduce the volume of printing material.

**Patterned structure synthesis** Hollowing the objects is the most straightforward scheme to reduce material usage, and has been adopted in commercial printing packages [Shapeways 2012]. The hollowing method is simply to hollow an object and possibly to fill its interior with pre-defined lattices to enhance strength. The size and density of the lattices and the thickness of skin surface are heuristically chosen by users and is far from optimal on saving material in printing. Using lightweight structures for improving the strength and stiffness of objects has been studied in the field of rapid manufacturing [Wang et al. 2005; Rosen et al. 2006], where the particle swarm optimization or generic algorithms were selected to search for design solutions. Patterned structures can also be mapped or synthesized into the interior of hollowed objects [Wang and Rosen 2002; Chen 2007]. However, the synthesized structures might not well adapt to high stress and there is no guarantee to obtain physically stable structures. So far these methods cannot simultaneously achieve the goals of saving material and maintaining physical stability and shape retention in 3D printing.

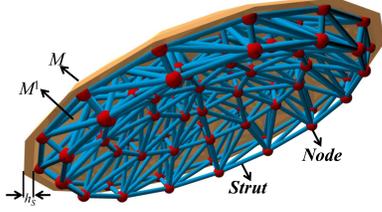
**Truss structure design and optimization** In the field of structural engineering, truss structures are flexible and broadly used for structural optimization. Detailed reviews on various aspects of structural optimization can be found in the literature [Kirsch 1989; Bendsøe and Sigmund 2003]. Structural size optimization, geometry optimization, and topology optimization are generally viewed as three different stages of structural optimization research. Using duality principles and non-smooth analysis, structural optimization problems with fixed nodal positions can be reformulated as min-max (saddle point) problems. Then the multilevel approach was constructed to solve the min-max problems by treating the topology optimization as the inner problem [Bendsøe et al. 1994]. Due to difference in the types of objectives and constraints, the approach cannot be applied to our optimization problem in 3D printing.

For structural optimization problems with complicated constraints, either approximate formulations or heuristic approaches (such as random search-based algorithms and branch-bound techniques) have been applied [Cheng 1995; Rajeev and Krishnamoorthy 1997; Bendsøe and Sigmund 2003]. An optimization method has been presented in [Smith et al. 2002] for designing truss structures of macro-architectures, such as bridges, towers, roof supports and building exoskeletons. Based on a mode of failure (i.e., buckling), their method considers the truss as a set of beams connected by pin joints, where the beams are rotation-free and exert axial forces only. It cannot guarantee geometrical approximation, which is very important for 3D printed objects, because there is no variable to characterize the deformations of beams in their method. On the contrary, our proposed method is based on the elastic property of solid materials and takes into account the deformations of struts and shear stress as well as axial stress.

## 3 Problem and Formulation

**Problem** Given an input mesh  $M$ , our goal is to generate a skin-frame structure  $\mathcal{H}$  to represent  $M$  such that the material cost used in printing  $\mathcal{H}$  is as low as possible. The shape of  $\mathcal{H}$  is expected to approximate  $M$  within a geometric error. Moreover, the design of  $\mathcal{H}$  depends on the physical characteristics of the printing material including its tensile strength  $\sigma$  (or compression strength), shear strength  $\tau$ , tensile modulus  $\gamma$ , shear modulus  $\mu$ , and slenderness ratio  $\alpha$ .

**Skin-frame structure** The skin-frame structure  $\mathcal{H}$  consists of a thin skin  $S$  and a frame structure  $\mathcal{T}$ , as shown in Figure 3.  $S$  is a thin layer between  $M$  and its offset  $M^1$  with a distance of skin thickness  $h_S$ .  $\mathcal{T}$  is created within the volume enclosed by  $M^1$ .  $\mathcal{T}$  is composed of a set of frame nodes  $V = \{\mathbf{v}_i, i = 1, 2, \dots, |V|\}$  which are located on  $M^1$  and in the volume enclosed by  $M^1$ , and a set of frame struts  $E = \{\mathbf{e}_j, j = 1, 2, \dots, |E|\}$  which are the edges connecting the nodes. Each node represents a geometric position and each strut  $\mathbf{e}_j \in E$  is a cylindrical shape with radius  $r_j$  and length  $l_j$ .  $\mathcal{T}$  can be seen as a graph of  $V$  and  $E$  with the geometry defining node positions and strut radii, and the topology defining the connectivity between nodes. To make it manufacturable by the printer, the strut radius should be no less than the minimum manufacturable size  $\eta$ .



**Figure 3:** Illustration of the skin-frame structure. It consists of a thin skin (shown in orange) bounded by  $M$  and its offset  $M^1$  with a distance of skin thickness  $h_S$  and a frame composed of a set of frame nodes (shown in red spheres) and a set of frame struts (shown in blue cylinders).

As we will see in Section 4, the set of skin nodes  $V_{\text{skin}}$  and the set of skin struts  $E_{\text{skin}}$  are fixed in order to sustain the physical properties of the model surface, while the set of internal nodes  $V_{\text{int}} = V - V_{\text{skin}}$  and the set of internal struts  $E_{\text{int}} = E - E_{\text{skin}}$  will be set as variables in the optimizations.

### 3.1 Mechanics analysis of frame structure

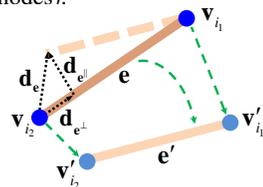
The mechanics of frame structures has been studied based on beam theory [Hughes 1987; Gibson and Ashby 1999] where frame struts are assumed to behave like simple beams under linear deformation.

**Stiffness** The finite element method (FEM) is used to derive the stiffness of the frame. The stiffness equation is given as:

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

where  $\mathbf{V}$  denotes geometric positions of the nodes  $V$ ,  $\mathbf{K}(\mathbf{V}, \mathbf{r})$  is the stiffness matrix depending on nodal positions  $\mathbf{V}$  and strut radii  $\mathbf{r}$ ,  $\mathbf{F}(\mathbf{r}) = \{\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_{|V|}\}$  are the (internal and external) forces acting on the nodes, and  $\mathbf{D} = \{\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_{|V|}\}$  are the deformation quantities of the nodes caused by  $\mathbf{F}(\mathbf{r})$ . We consider the mass of the skin layer and the struts as the internal load in  $\mathbf{F}$  by distributing the mass uniformly to their neighborhood nodes. External loads acting on the frame with any directions can be handled by adding the forces in  $\mathbf{F}$ . For example, we can add forces in  $\mathbf{F}$  to withstand external loads like a pinch grip. After the force  $\mathbf{F}$  is specified under a given load scenario, the deformations of nodes  $\mathbf{D}$  can be solved from the linear system with some boundary conditions (e.g., fixing one node and the  $z$ -axes of three other nodes).

**Elastic property** Because of elasticity, all struts are deformable when forces act on them. Considering a strut  $\mathbf{e} = \mathbf{v}_{i_1}\mathbf{v}_{i_2}$  (see the figure on the right), it is deformed to  $\mathbf{e}' = \mathbf{v}'_{i_1}\mathbf{v}'_{i_2}$ . The displacements of two end nodes are  $\mathbf{d}_{i_1} = \mathbf{v}'_{i_1} - \mathbf{v}_{i_1}$  and



$\mathbf{d}_{i_2} = \mathbf{v}'_{i_2} - \mathbf{v}_{i_2}$  respectively, and the

difference between them is  $\mathbf{d}_e = \mathbf{d}_{i_2} - \mathbf{d}_{i_1}$ . Denote  $\mathbf{d}_{e\parallel} = \frac{\mathbf{e}^T \mathbf{d}_e}{\|\mathbf{e}\|^2} \mathbf{e}$  and  $\mathbf{d}_{e\perp} = \mathbf{d}_e - \mathbf{d}_{e\parallel}$  as decomposition components of  $\mathbf{d}_e$  along  $\mathbf{e}$  and its orthogonal direction, respectively. Then the deformation of  $\mathbf{e}$  should be within the ranges of axial stress and shear stress as:

$$\frac{\|\mathbf{d}_{e\parallel}\|}{\|\mathbf{e}\|} \gamma \leq \sigma, \quad \mathbf{e} \in E, \quad (2)$$

$$\frac{\|\mathbf{d}_{e\perp}\|}{\|\mathbf{e}\|} \mu \leq \tau, \quad \mathbf{e} \in E. \quad (3)$$

Note that rotation and bending stress is weak and can be negligible. The deformations of nodes  $\mathbf{D} = \{\mathbf{d}_i\}$  are auxiliary variables in our optimization as they are coupled with the primary variables  $\mathbf{r}$ .

**Buckling** Compressive forces can cause a beam to bend out of shape and ultimately fail, which is known as Euler buckling (see the figure in the right). The struts are subject to buckling constraints according to

$$r_j \geq l_j / \alpha, \quad \mathbf{e}_j \in E_{\text{skin}}, \quad (4)$$

and

$$r_j \geq l_j / \alpha, \quad \mathbf{e}_j \in E_{\text{int}}, \quad (5)$$

where  $\alpha$  is the slenderness ratio.

### 3.2 Other constraints

Other than the mechanism constraints on the frame, we have to consider other constraints to make the frame structure physically stable, geometrically approximate, and printable.

**Geometrical approximation** The frame used in 3D printing is required to geometrically approximate the shape of input object. Thus the frame structure should not be deformed excessively to avoid apparent visual alteration, that is

$$\|\mathbf{d}_i\| \leq \epsilon, \quad i = 1, 2, \dots, |V| \quad (6)$$

where  $\epsilon$  is the geometric error. Note that the geometrical approximation is an extremely necessary constraint for 3D printing.

**Shape barrier** Another geometric constraint function is 'barrier' constraint, which keeps internal struts inside of the volume enclosed by  $M^1$ . For each internal strut  $\mathbf{e}_j = \mathbf{v}_{i_1}\mathbf{v}_{i_2} \in E_{\text{int}}$ , we let

$$\mathbf{v}_{i_1}, \mathbf{v}_{i_2} \in C_j \quad (7)$$

where  $C_j$  is the maximum convex region that contains  $\mathbf{e}_j$  but is enclosed by  $M^1$ . This shape barrier constraint only needs to be considered in the geometry optimization where the positions of internal nodes are variables.

**Balance** To balance the printed object standing on a horizontal plane, the vertical projection  $G_{\text{proj}}$  of its gravity center  $G$  onto the plane should lie within the convex hull  $H$  of its contact points on the plane [Prévost et al. 2013]. Thus we have the constraint

$$G_{\text{proj}} \in H, \quad (8)$$

which can be expressed as a group of linear inequalities on  $\mathbf{r}$  (see more detail in the supplementary material).

**Printability** To make the frame structure printable, the strut radii should be no less than the minimum printable radius  $\eta$ . In addition, we also set upper bounds for strut radii. The struts on the skin should have a radius no larger than  $h_S$  to avoid lying outside of the object surface  $M$ , i.e.,

$$\eta \leq r_j \leq h_S, \quad \mathbf{e}_j \in E_{\text{skin}}, \quad (9)$$

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

$$\eta \leq r_j \leq \bar{\eta}, \quad \mathbf{e}_j \in E_{\text{int}}. \quad (10)$$

**Constraint categories** All the constraints are categorized into two types: equalities (1) and inequalities. We denote the inequality constraints including (2), (3), (4), (6), (7), (8), and (9) as

$$\mathbf{c}(\mathbf{r}, \mathbf{V}, \mathbf{D}) \leq \mathbf{0}. \quad (11)$$

The deformations of nodes  $\mathbf{D}$  are auxiliary variables which are coupled with  $\mathbf{r}$  in the constraints. Note that the constraints (5) and (10) are not included in (11) as these constraints will be relaxed during the topology optimization phase as we will see in Section 4.3.

### 3.3 Multi-objective programming formulation

**Frame volume** As the skin layer  $\mathcal{S}$  is solid, raising its thickness will significantly increase the total volume. We thus fix its thickness  $h_S$  as the minimum printable size (i.e.  $h_S = 2\eta$ ) and do not consider it as a variable. Therefore our major goal is to minimize the volume of all frame struts

$$\min_{\mathbf{r}, \mathbf{V}, E} \text{Vol}(\mathbf{r}, \mathbf{V}, E) = \sum_{\mathbf{e}_j \in E} \pi r_j^2 l_j, \quad (12)$$

where  $\mathbf{V}$  denotes geometric positions of the nodes  $V$ .

**Structure redundancy** As the radius of each strut should not be less than the minimum manufacturable size  $\eta$ , too many struts might lead to large volume. The redundant struts in the structure should be eliminated to reduce the frame volume. Thus we have a minor goal of minimizing the number of struts in the frame

$$\min_E |E|. \quad (13)$$

**Multi-objective programming** A straightforward method is to formulate a single objective optimization with a weighted combination of the frame volume and the structure redundancy. However, it is very difficult to construct an appropriate weight to trade off the two objectives. Instead we formulate a multi-objective programming (MOP) problem:

$$\begin{cases} \min_{\mathbf{r}, \mathbf{V}_{\text{int}}, E_{\text{int}}} \text{Vol}(\mathbf{r}, \mathbf{V}, E) & \& |E_{\text{int}}| & (14a) \\ \text{s.t.} & \{ (1), (11), (5), (10) \} & (14b) \end{cases}$$

where the frame volume  $\text{Vol}(\mathbf{r}, \mathbf{V}, E)$  is the higher-priority goal and the number of internal struts  $|E_{\text{int}}|$  is the lower-priority goal.

In our MOP formulation, the strut radii  $\mathbf{r}$ , the positions of internal nodes  $\mathbf{V}_{\text{int}}$ , and the topology connections  $E_{\text{int}}$  are decision variables, and the deformations  $\mathbf{D}$  are auxiliary variables which are coupled with  $\mathbf{r}$  in the constraints. The positions of skin nodes  $\mathbf{V}_{\text{skin}}$  and the set of skin struts  $E_{\text{skin}}$  are fixed rather than variables throughout the optimization process.

## 4 Algorithm

In the MOP optimization (14), the radii  $\mathbf{r}$  and the positions  $\mathbf{V}_{\text{int}}$  are continuous variables while the topology connections  $E_{\text{int}}$  are discrete variables. Theoretically, it is NP-hard to solve this kind of combinatorial (mixed integer) optimization problem. To find a compromise solution, we develop an alternating goal algorithm. The algorithm starts by ordering the goals according to priority, and iteratively optimizes the model using one goal at a time such that the optimum value of a higher-priority goal is never degraded by a lower-priority goal.

### 4.1 Overview

An overview of our algorithm is shown in Figure 4. Given an input mesh (e.g., the Hanging Ball), we generate an initial frame with the radii of its struts obtained from the size optimization. Then the algorithm runs alternately the topology optimization and the geometry optimization until the frame volume does not decrease. The topology optimization eliminates the redundant struts by approximately solving an  $\ell_0$  sparsity optimization under the frame volume constraint. The geometry optimization refines the positions of internal nodes, the radii of struts, and, to some extent, the topology of a frame structure. Finally an optimized frame structure with minimum volume is obtained as the final result.

---

**Algorithm 1** Our algorithm for frame optimization

---

**Input:** a given 3D mesh

**Output:** an optimized frame

---

```

1: Generate an initial frame presented in Section 4.2
while (the frame volume is reduced)
{
2: Call TopOpt (the topology optimization in Section 4.3)
3: Call GeoOpt (the geometry optimization in Section 4.4)
}

```

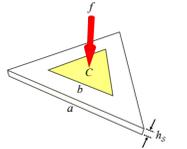
---

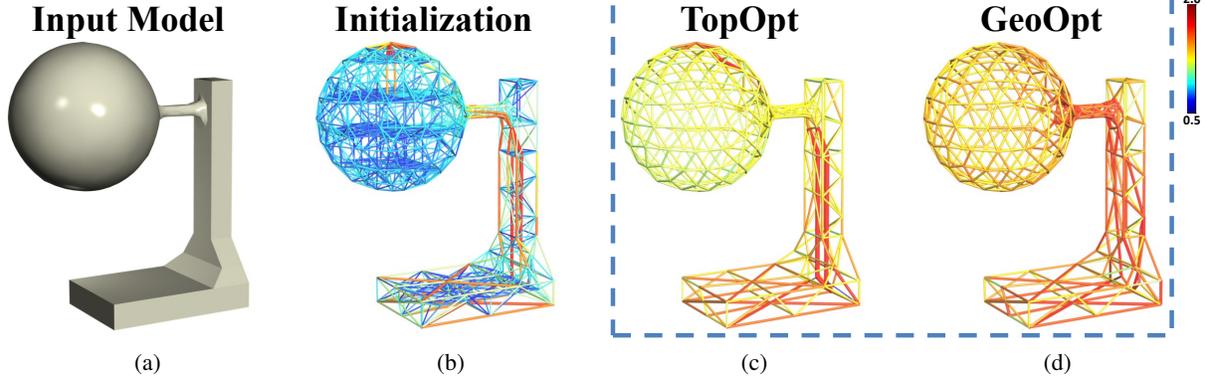
### 4.2 Frame initialization

Our optimization starts with an initial frame structure for which we need to determine the number of the frame nodes, their positions and connectivity, and the initial strut radii.

**Determining the number of skin nodes** The user is allowed to manually specify the number of frame nodes and uniformly sample the nodes on  $M^1$ . However, it is difficult to specify a reasonable number of nodes. Dense nodes may lead to material waste while sparse nodes may result in fragile triangles on the skin which might not withstand external loads like a pinch grip with two fingers [Stava et al. 2012].

The number of nodes on  $M^1$  (i.e.,  $|V_{\text{skin}}|$ ) is determined by considering the physical properties of the surface. Suppose we put a force  $f$  perpendicular to an equilateral triangle with edge length  $a$  on its center  $C$  by a human finger (see the figure on the right). Let  $b$  be the edge length of the equilateral triangle approximating the contact region of the finger. The displacement of  $C$  can be estimated (from the definition of shear stress) as  $d \approx \frac{f(a-b)}{3\sqrt{3}\mu h_S b}$ . The deformation  $d$  is expected to be no larger than the upper bound of deformation  $\varepsilon$  to avoid apparent visual alteration. Thus we have  $a \leq 3\sqrt{3}\mu h_S b \varepsilon / f + b$  and set  $a$  as the upper bound. Then the number of sampling nodes on  $M^1$  can be determined as  $|V_{\text{skin}}| = 4\text{Area}(M^1) / (\sqrt{3}a^2)$  assuming all triangles are equilateral. In our implementation, we set  $f = 10\text{N}$  (Newtons) and  $b = 10\text{mm}$  (millimeters).





**Figure 4:** Overview of our algorithm. Given an input model (a), an initial frame structure (b) is generated. Our algorithm runs alternately the topology optimization (c) and the geometry optimization (d). The struts in (b), (c), and (d) are shown with color visualizations of their radii. Note that the frame in (c) is much sparser than that in (b). The frame volumes of (b) and (d) are  $3.790e4$  and  $2.875e4 \text{ mm}^3$  respectively. The saving ratio of the frame volume is about 24%. In this example, an external force of  $5N$  is loaded vertically downside on top of the model.

**Node sampling and their connectivity** A stress map can be computed on the solid volume enclosed by  $M^1$  based on the finite element method (FEM). We adaptively sample  $|V_{\text{skin}}|$  nodes on  $M^1$  according to the density map [Yan et al. 2009] given by the computed stress values on the surface. In addition, a user-specified number of internal nodes are evenly sampled in the solid volume enclosed by  $M^1$ . Finally, the frame structure  $\mathcal{T}$  is generated by connecting the sampling nodes via the ANN algorithm [Arya et al. 1998]. Note that there are few requirements on the generation of  $\mathcal{T}$ , because redundant struts in  $\mathcal{T}$  will be automatically eliminated by the topology optimization (see Section 4.3). The struts might lie outside of the shape at very concave regions. We perform local subdivision by adding nodes and struts adaptively to resolve this.

**Size optimization** We then determine the initial radii of struts by the following size optimization (SizeOpt in short):

$$\begin{aligned} \min_{\mathbf{r}} \quad & \text{Vol}(\mathbf{r}, \mathbf{V}, E) \\ \text{s.t.} \quad & \{ (1), (11), (5), (10) \}. \end{aligned} \quad (15)$$

Note that the geometric positions of internal nodes (i.e.,  $\mathbf{V}_{\text{int}}$ ) and the topological connectivity of internal struts (i.e.,  $E_{\text{int}}$ ) are fixed rather than variable in the size optimization. We solve the size optimization (15) using the interior-point algorithm provided in [Nocedal and Wright 2006, Chapter 19, Page 568].

### 4.3 Topology optimization

Let  $\bar{\mathbf{r}}$  be the solution of the size optimization (15) and  $\widetilde{\text{Vol}} = \text{Vol}(\bar{\mathbf{r}}, \mathbf{V}, E)$  be the corresponding frame volume. We have observed that many strut radii obtained by the size optimization reach the lower bound  $\underline{\eta}_j = \max(\underline{\eta}, l_j/\alpha)$ . Some of the struts may have negligible influence on the whole frame structure and would probably be eliminated. This can be achieved by the topology optimization in the following.

**Formulation** The solution of the size optimization prompts us to possibly obtain a sparser frame structure without increasing the frame volume. We have observed that one strut vanishes when its radius becomes 0. Thus the number of struts is equal to the number of non-zero strut radii, i.e.,  $|E_{\text{int}}| = \|\mathbf{r}_{\text{int}}\|_0$ . Therefore, an  $\ell_0$  sparsity optimization can be formulated to eliminate redundant struts and achieve simplicity of the frame structure. A straightforward idea is to replace the buckling constraint (5) and the printable

radius bound (10) for each internal strut by a looser constraint

$$r_j \in [0, \bar{\eta}], \quad \mathbf{e}_j \in E_{\text{int}} \quad (16)$$

where  $r_j = 0$  means that the corresponding strut  $\mathbf{e}_j$  vanishes. Then we present the  $\ell_0$  sparsity optimization (TopOpt in short) as:

$$\begin{aligned} \min_{\mathbf{r}} \quad & |E_{\text{int}}| = \|\mathbf{r}_{\text{int}}\|_0 \\ \text{s.t.} \quad & \{ (1), (11), (16), \text{and } \text{Vol}(\mathbf{r}) \leq \widetilde{\text{Vol}} \} \end{aligned} \quad (17)$$

where the  $\ell_0$ -norm means the number of non-zero elements of  $\mathbf{r}$  and  $\widetilde{\text{Vol}} = \text{Vol}(\bar{\mathbf{r}}, \mathbf{V}, E)$  is the frame volume obtained in the size optimization. The goal of reducing  $E_{\text{int}}$  holds a lower priority in our multi-objective programming. To prevent degrading the higher-priority goal (i.e., minimizing the frame volume), we are required to add a volume constraint  $\text{Vol}(\mathbf{r}, \mathbf{V}, E) \leq \widetilde{\text{Vol}}$  when optimizing the lower-priority goal.

Beyond conceptual issues of uniqueness and verification of solutions, one is easily overwhelmed by the apparent difficulty of solving (17) due to the discrete and discontinuous nature of the  $\ell_0$ -norm, and the standard convex analysis theories which underpin the analysis of differentiable optimization problems do not apply. This is a classical problem of combinatorial search: one sweeps exhaustively through all possible sparse subsets and checks whether the constraints can be satisfied. The complexity of exhaustive search is exponential in the scale of problem and, indeed, it has been proved that the combinatorial search problem is NP-hard. However, some useful techniques, i.e., a convex relaxation and reweighting strategy, can be incorporated in our algorithm. We present a reweighted  $\ell_1$  minimization procedure to compute an approximate solution as follows.

**Reweighted  $\ell_1$  minimization** A way to render (17) more tractably is to regularize the highly discontinuous  $\ell_0$ -norm, replacing it with a continuous and convex approximation. Therefore we convexify the objective function with the weighted  $\ell_1$ -norm and come to a new minimization problem

$$\begin{aligned} \min_{\mathbf{r}} \quad & \|\mathbf{W}\mathbf{r}\|_1 = \mathbf{w}^T \mathbf{r} \\ \text{s.t.} \quad & \{ (1), (11), (16), \text{and } \text{Vol}(\mathbf{r}) \leq \widetilde{\text{Vol}} \} \end{aligned} \quad (18)$$

where  $\|\mathbf{W}\mathbf{r}\|_1 = \mathbf{w}^T \mathbf{r}$  holds due to the non-negativity of  $\mathbf{r}$ , and  $\mathbf{W}$  is a diagonal positive definite matrix whose diagonal entries are given by the weight vector  $\mathbf{w}$ .

The reweighted  $\ell_1$ -norm minimization (18) can be regarded as a continuous and convex approximation of an  $\ell_0$ -norm optimization problem (17). It is desired that the weights counteract the influence of the radius magnitude on the  $\ell_1$  penalty. There exists such a possibility of constructing a favorable set of weights based on the solution  $\hat{\mathbf{r}}$  of the size optimization. Therefore, a weight vector  $\mathbf{w}$  can be accordingly designed by

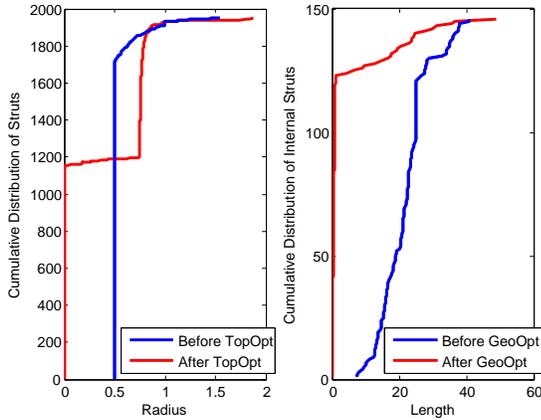
$$w_j = \frac{1}{\xi + (\hat{r}_j - \underline{\eta}_j)} \quad (19)$$

where  $\xi$  is a small number ( $10^{-7}$  by default) that provides numerical stability.

**Sparse frame solution** Let  $\hat{\mathbf{r}}$  be the solution of reweighted  $\ell_1$  minimization (18). Figure 5 (Left) shows the cumulative distributions of strut radius before and after TopOpt on the Hanging Ball model. It is seen that the solution obtained from the topology optimization clearly reflects the simplicity of the frame structure. So we eliminate redundant internal struts indicated by  $\hat{\mathbf{r}}$  and reduce the set of struts inside the skin as

$$\hat{E}_{\text{int}} = \{\mathbf{e}_j \in E_{\text{int}} \mid \hat{r}_j \geq \zeta\}, \quad (20)$$

where the threshold  $\zeta$  is set to  $10^{-5}$  mm in our implementation.



**Figure 5:** Distributions of strut radii and lengths before and after topology optimization and geometry optimization on the Hanging-Ball model shown in Figure 4. Left: the cumulative distributions of strut radii before and after topology optimization. Right: the cumulative distributions of internal strut lengths before and after geometry optimization.

#### 4.4 Geometry optimization

In our multi-objective programming, the goal of minimizing the volume  $\text{Vol}(\mathbf{r}, V, E)$  has the higher priority. The geometric positions of internal nodes and the radii of struts are refined by the following geometry optimization (GeoOpt in short):

$$\begin{aligned} \min_{\mathbf{r}, \mathbf{V}_{\text{int}}} \quad & \text{Vol}(\mathbf{r}, \mathbf{V}, \hat{E}) \\ \text{s.t.} \quad & \{(1), (11), (5), (10)\} \end{aligned} \quad (21)$$

where the topological connectivity of the frame structure is fixed as  $\hat{E} = E_{\text{skin}} \cup \hat{E}_{\text{int}}$  and the reduced set of internal struts  $\hat{E}_{\text{int}}$  is obtained from the topology optimization. We solve the geometry optimization (21) using the interior-point algorithm in [Nocedal and Wright 2006, Chapter 19]. Note that in the phase of geometry

optimization the radii of frame struts (i.e.,  $\mathbf{r}$ ) and the positions of internal nodes (i.e.,  $\mathbf{V}_{\text{int}}$ ) are variables.

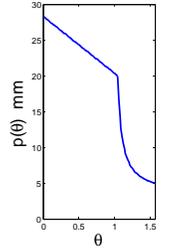
Let  $(\mathbf{r}^*, \mathbf{V}_{\text{int}}^*)$  be the solution from the geometry optimization (21). If the length of an internal strut vanishes in the phase of geometry optimization, the two corresponding nodes merge into one (i.e., their geometric positions are identical). Figure 5 (Right) shows the cumulative distributions of internal strut lengths, which are computed before and after GeoOpt on the Hanging Ball model. It is seen that the geometry optimization can also remove some redundant nodes in the frame structure.

Ultimately, we get an optimized frame structure  $\mathcal{T}^*$  in which  $\mathbf{V}^* = \mathbf{V}_{\text{skin}} \cup \mathbf{V}_{\text{int}}^*$  is the positions of frame nodes,  $E^* = \hat{E} = E_{\text{skin}} \cup \hat{E}_{\text{int}}$  is the set of frame struts, and  $\mathbf{r}^*$  is the vector of strut radii.

#### 4.5 Self-supporting extension for extrusion-type 3D printers

For powder-type 3D printers, the frame structure can be printed due to supporting material. However, for extrusion-type 3D printers which do not have the supporting material during the printing process, extra supporting struts should be added for printability. To this end, we propose a scheme to resolve it by adding some extra struts to make the frame printable for such printers.

We define a metric function  $p(\theta)$  as the maximum length of a printable strut  $\mathbf{e}$ , with a certain range of radius, which has an angle of  $\theta \in [0, \pi/2]$  to the vertical direction. That is,  $\mathbf{e}$  is self-supporting printable if its length is less than  $p(\theta)$ . The function  $p(\theta)$  is estimated experimentally as shown in the figure on the right. The vertical component of deformation of the strut  $\mathbf{e} = \mathbf{v}_{i_1} \mathbf{v}_{i_2}$  is given as



$$d_{\mathbf{e}}^z = (0, 0, 1) \cdot (\mathbf{d}_{i_2} - \mathbf{d}_{i_1}), \quad (22)$$

where node  $\mathbf{v}_{i_1}$  lies below node  $\mathbf{v}_{i_2}$ . The algorithm for generating self-supporting frame by adding extra struts for extrusion-type 3D printers is given as Algorithm 2.

---

#### Algorithm 2 Self-supporting extension for extrusion-type printers

---

**Input:** a frame  $\mathcal{T}^* = (V^*, E^*)$  generated by Algorithm 1

**Output:** a self-supporting frame  $\mathcal{T}^{(s)}$

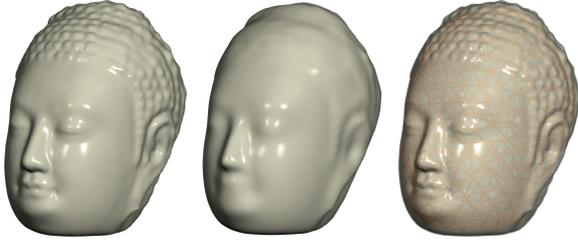
---

- 1: Let  $V^{(s)} = V^*$  and  $E^{(s)} = E^*$ , and define a base plane  $B$  for the input frame.
  - 2: For each  $\mathbf{e}_j \in E^*$ , compute its length  $l_j$  and vertical angle  $\theta_j$ .  
**if**  $p(\theta_j) < l_j$ , let  $m(\mathbf{e}_j) = \lceil \frac{l_j}{p(\theta_j)} \rceil - 1$  and insert equidistant nodes  $\mathbf{v}_k, k = 1, \dots, m(\mathbf{e}_j)$ , on the strut  $\mathbf{e}_j$ .  
**for**  $k = 1, \dots, m(\mathbf{e}_j)$   
Find a closest node  $\mathbf{v}'_k \in V^{(s)} \cup B$ , which lies below  $\mathbf{v}_k$  such that the new strut  $\mathbf{e}_k = \mathbf{v}'_k \mathbf{v}_k$  is self-supporting printable.  
Update the frame by setting  $V^{(s)} = V^{(s)} \cup \{\mathbf{v}'_k\}$  and  $E^{(s)} = E^{(s)} \cup \{\mathbf{e}_k\}$ .  
**endfor**
  - endif**
  - 3: Call the size optimization with  $d_{\mathbf{e}}^z \leq 0$  added for each  $\mathbf{e} \in E^{(s)}$  to the constraints and obtain the optimized radius vector  $\mathbf{r}^{(s)}$ . Output a self-supporting frame  $\mathcal{T}^{(s)}$  with its strut radii  $\mathbf{r}^{(s)}$ .
-

## 4.6 Implementation details

**Thin structures** Some models have thin parts like the Shell model in Figure 11. If the thickness of the thin part is very small, we connect the nodes on  $M^1$  directly to create the frame without sampling internal nodes inside the solid volume of  $M^1$ . Some models have even thinner structures like the arms and legs of Bananamam (Figure 11) which reach approximately the minimum printable size. We regard them as solid layers or struts in the optimization.

**Fine geometric details** For models with fine geometric details, we perform preprocessing by over-smoothing [Taubin 1995] them in our implementation. If some parts of the smoothed surface  $M'$  lie outside the input surface  $M$ , we project those vertices onto  $M$  and perform a local remeshing to guarantee that  $M'$  is inside  $M$  as a whole. We then regard the smoothed surface  $M'$  as the input object mesh and apply our algorithm on it. An example, the Buddha Head model, is shown in Figure 6. At this point, the volume between the original surface and the smoothed one is considered as part of the skin layer.



**Figure 6:** For the Buddha Head model (left) with fine geometric details, we over-smooth the mesh and ensure that the smoothed mesh (middle) lies inside the original surface by local modification. Then our algorithm is applied on it to produce a skin-frame (right) for printing. The resulting frame is rendered with a transparent skin as shown on the right.

## 5 Experimental results

**System configurations** We test our algorithm and fabricate the objects using both powder-type and extrusion-type 3D printers. The powder-type printer we use is the Sinterstation<sup>®</sup> HiQ<sup>™</sup> SLS<sup>®</sup> (P-1) with tray size 457mm × 380mm × 330mm [3DSYSTEMS 2012]. For P-1 printer, we practically made holes with diameters about 20mm on the models. Then the internal powder can be removed by gently shaking the printed objects.

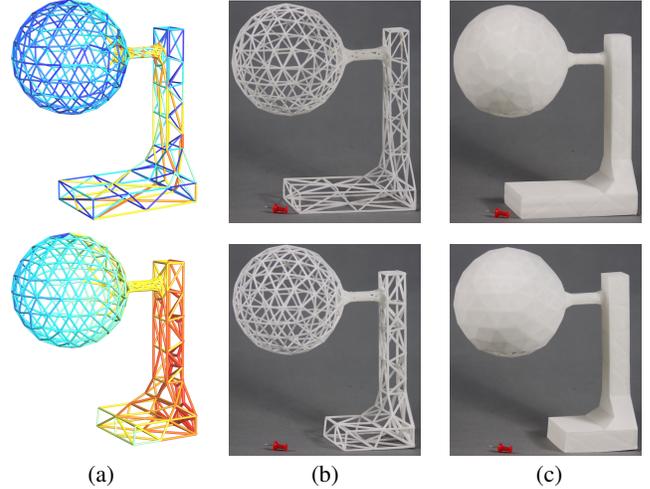
The extrusion-type printer we use is the MakerBot Replicator<sup>™</sup> 2 (P-2) with tray size 225mm × 145mm × 150mm [Makerbot 2012]. For P-2 printer, we need to manually remove only the external supporting struts that lie outside the printed models.

**Material properties and parameters** Table 1 shows the parameters and properties of materials used in printing the models by both printers. The upper bound for internal strut radius is set to  $\bar{\eta} = 5\text{mm}$  and the upper bound of deflection is  $\epsilon = 0.05\text{mm}$  for both printers.

Printer	$\eta$	Material	$\gamma$	$\mu$	$\sigma$	$\tau$	$\alpha$
P-1	0.5	PA plastic	1586	1387	43	48	60
P-2	0.4	PLA	2673	1533	92	52	60

**Table 1:** The parameters and properties of printing materials for each of the printers. The unit of minimum printable radius  $\eta$  is mm. Units of  $\gamma$ ,  $\mu$ ,  $\sigma$ , and  $\tau$ , are MPa (Mega Pascals). The slenderness ratio  $\alpha$  is dimensionless.

**Performance** We have applied our algorithm on a number of 3D models (also see the accompanying video). Our algorithm starts with an initial frame of the input model and obtains the final result by running in turn the topology optimization and the geometry optimization until the reduction of frame volume is less than 1%. In our experiments, the algorithm needs only 1 iteration to obtain the final results. It takes about 40 minutes to compute the final frame for a model with an initial frame of 500 nodes and 3000 struts. The manufacturing time is about 27 hours on average to print altogether 4 models of 200mm in height using the P-1 printer. With P-2, it takes about 5 hours on average to print a model of 150mm in height.



**Figure 7:** The Hanging-Ball model in the lower row has a smaller base than the one in the upper row. The struts in (a) are coded with the same color bar in Figure 4. For the model with a smaller base, our algorithm produces thicker struts on the vertical pillar in the right part than the counterparts in the upper row due to the balancing constraint (8). Photos of the printed naked frame and the printed objects using P-1 are shown in (b) and (c), respectively.

### 5.1 Results and discussions

Figure 7 shows two Hanging-Ball models with different sizes of bases printed using the powder-based printer P-1. If the strut radii are set uniformly, the model with the smaller base is too front-heavy and cannot stand erect. As we consider self-balance as a constraint in the optimization, using our algorithm, many of the struts on its vertical pillar will become much thicker. More models processed by our algorithm and then printed using P-1 are shown in Figure 8.

For the extrusion-type printer P-2, we need to call the self-supporting extension scheme (Algorithm 2) to create extra struts, as shown in Figure 9. With the help of extra struts, both the frame and the skin-frame can be correctly printed using P-2. The final printed models by removing the external supporting struts can be seen in the accompanying video.

**Statistics** Table 2 shows the statistics of all examples produced by our algorithm. ‘Ratio’ denotes the ratio of the total volume (i.e., the sum of frame volume and skin volume) to the solid volume. The material savings is about 70% (even up to 92.1% for the Buddha-Head model) with respect to the solid printing. It shows that the number of struts has been reduced between 30% (Shell) and 60% (Hanging-Ball) during the optimizations. Thus we can see that our topology optimization can eliminate redundant struts and identify primary struts from the volume constraint for the w-hole frame. Note that the geometry optimization can also, to some extent, remove redundant internal nodes in the frame.



**Figure 8:** Printed objects using powder-type printers (P-1) produced by our algorithm. From left to right: Fighter, TV-Alien, Fishing-Frog, and Buddha-Head. The upper row shows the rendering results with half skin and half frame. The lower row shows photos of the printed objects. The largest edge length of the bounding box of each object is 200mm. A small red drawing pin is put beside each object as a size reference.

Model	Solid Vol. ( $10^4\text{mm}^3$ )	#Nodes / #Struts			Frame Vol. ( $10^4\text{mm}^3$ )	Total Vol. ( $10^4\text{mm}^3$ )	Ratio	Weight (Gram)
		Initial	TopOpt	GeoOpt				
Horse (Fig. 1)	51.361	689 / 3167	689 / 1913	604 / 1828	2.233	7.717	15.0%	90.5
Hanging-Ball (Fig. 4)	81.653	380 / 1949	380 / 884	266 / 770	2.875	9.475	11.6%	109.3
Fighter (Fig. 8)	12.839	598 / 2945	598 / 1598	503 / 1503	2.498	4.119	32.1%	48.5
TV-Alien (Fig. 8)	12.880	464 / 2243	464 / 1282	412 / 1230	1.718	2.944	22.9%	34.8
Fishing-Frog (Fig. 8)	32.317	720 / 3365	720 / 1952	616 / 1848	2.812	5.231	16.2%	61.1
Buddha-Head (Fig. 8)	164.030	575 / 2642	575 / 1369	400 / 1194	3.827	12.940	7.9%	149.9
Shell (Fig. 11)	25.977	414 / 1791	414 / 1236	414 / 1236	4.106	8.754	33.7%	103.7
Bananaman (Fig. 11)	23.530	462 / 2002	462 / 1222	407 / 1167	1.930	6.073	25.8%	72.1

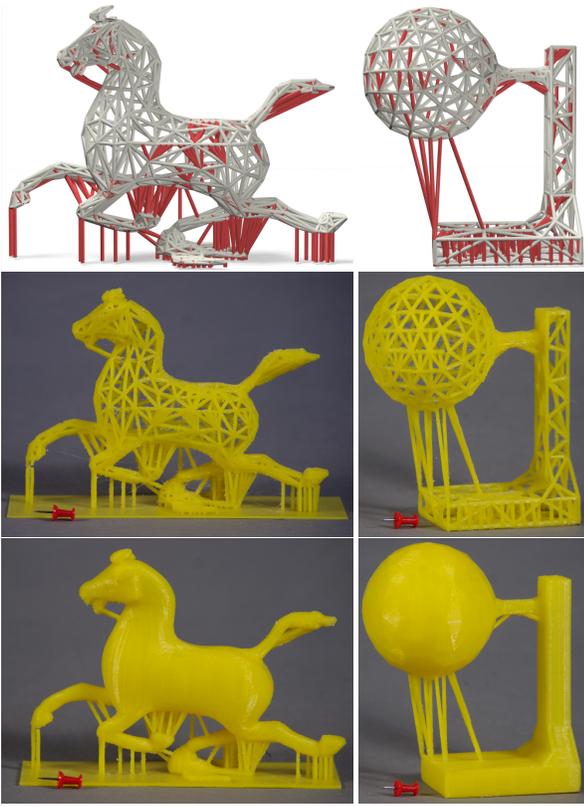
**Table 2:** Statistics of applying our algorithm to various models shown in the paper. The numbers of nodes and struts are shown for Initialization, TopOpt, and GeoOpt, respectively. Solid Vol. denotes the volume of the solid object. Frame Vol. denotes the volume of the optimized frame struts. Total Vol. denotes the sum of frame volume and skin volume. Ratio denotes the ratio of Total Volume to Solid Volume. Weight denotes the actual weight of the printed object using P-1 with PA plastic material, whose density is  $1.15\text{Gram}/\text{cm}^3$ .

## 5.2 Comparisons

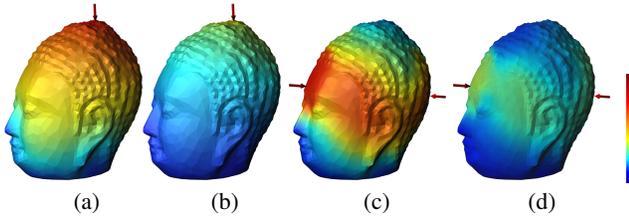
**Comparison to the hollowing method** We compare our algorithm with the hollowing method. Thanks to the excellent mechanical properties of the frame structure, the thin skin layer reinforced with the frame structure is strong and stiff. The printed objects obtained by our algorithm are rigid with a rather small mass. Figure 10 demonstrates the comparison on the strength and stiffness of the two objects produced by the naive hollowing method and our algorithm with same volumes respectively. If an external load of 20N acts on the objects (with different directions), the illustration shows that the object produced by our algorithm has smaller deformation. Furthermore, the self-balancing property is one of the major advantages of our method. Usually, uniformly hollowed objects are not stable, since the hollowing method does not take self-balance into account. Some comparison examples (the Horse and the Hanging-Ball) are shown in the accompanying video.

**Comparison to [Stava et al. 2012]** Although the approach of [Stava et al. 2012] strengthens fragile objects by local thickening and strut addition, it cannot always reduce the volume of printing material. Figure 11 (left) shows that two external struts were added in the Shell model (upper row) and one external strut was added in the Bananaman model (lower row) using their approach. The frame results produced by our algorithm are stiff enough to sustain themselves without adding any external struts, and thus have no visual impact on the appearance of the objects. The volumes of Shell and Bananaman by their approach are  $25.977\text{e}4$  and  $23.530\text{e}4 \text{mm}^3$ , respectively, without adding the volume of external struts. The volumes of our produced results are  $8.754\text{e}4$  and  $6.073\text{e}4 \text{mm}^3$  for these two models, respectively.

**Comparison to greedy strategy of topology cleaning** For the purpose of topology optimization, the approach of [Smith et al. 2002] used a greedy strategy (called the topology-cleaning step)

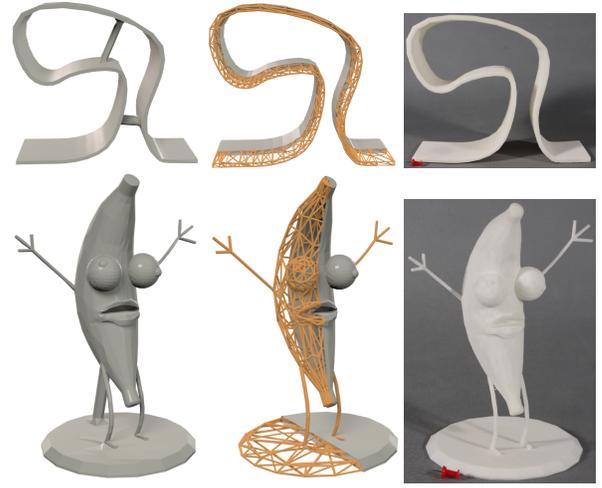


**Figure 9:** Printed objects using extrusion-type printers (P-2) produced by Algorithm 2. The first row shows the rendered images of the frame structures. The extra supporting struts are shown in red. With the extra struts, both the frame (the second row) and the skin-frame (the third row) can be correctly printed using P-2. The final printed models with the external supporting struts removed can be seen in the accompanying video.

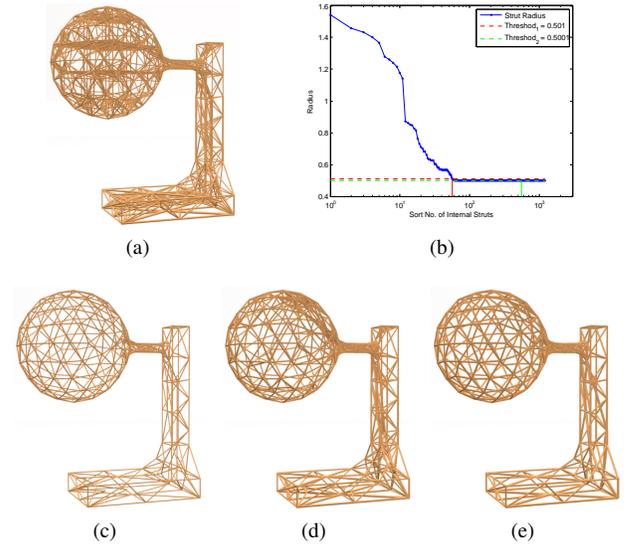


**Figure 10:** Deformation maps on the Buddha-Head models by the naive hollowing method (a,c) and our algorithm (b,d) under an external load of 20N. The models produced by the naive hollowing method (thickness of 1.30 mm) and ours have the same material volume ( $12.940e4 \text{ mm}^3$ ), and are shown with a color visualization of the deflection displacements of their vertices.

to eliminate beams that are exerting little force and merged any pairs of joints that are connected to one another by a beam that is at the minimum allowable length. The greedy strategy of topology cleaning needs to choose a threshold to remove the redundant struts whose radii are mostly around  $\eta$ . A small threshold value may keep too many redundant struts and thus it cannot achieve the purpose of saving material, while a large threshold value may remove a dominant set of internal struts and result in no feasible solution, i.e., the remaining struts with all allowable radii cannot meet the constraints of the problem. Thus the greedy method is unstable and sensitive to



**Figure 11:** Left: results produced by the method of [Stava et al. 2012] by adding external struts; Middle: the skin-frame produced by our algorithm (with half-naked rendering); Right: printed objects of our result using P-1.



**Figure 12:** The effect of applying the greedy topology-cleaning to (a) the initial frame of Hanging-Ball model. Two very close thresholds 0.501mm and 0.5001mm for greedy topology-cleaning (b) lead to very different results. The first threshold (c) retains 56 internal struts such that the model has no feasible solution in the geometry optimization while the second threshold (d) retains 552 internal struts to reach a solution with the frame volume of  $3.389e4 \text{ mm}^3$ . In contrast, the final frame produced by our algorithm (e) has a volume of  $2.875e4 \text{ mm}^3$  with only 52 internal struts.

the selected thickness threshold and it is very difficult for users to choose an appropriate threshold. The effect of applying this greedy strategy is illustrated in Figure 12. Instead of the greedy topology-cleaning, we achieve the topology optimization by approximately solving an  $\ell_0$  sparsity problem which can globally optimize the simplicity of the frame structure. Obviously, our topology optimization is more convincing, and can automatically eliminate the redundant struts in the frame structure while keeping smaller volume.

## 6 Conclusions and Future Work

In this work, we propose an automatic and practical method to generating a skin-frame structure for a given 3D model for the purpose of minimizing the material used in printing it and the number of struts in the structure. The frame structure generated by our algorithm, which is geometrically approximate to the shape of input model, is guaranteed to be physically stable and printable. For extrusion-type printers, we also develop a scheme to add extra struts to support the printed object during the printing process. A number of experimental results have shown the applicability and practicality of our proposed method and have proved our solution to be much more cost-effective than previous methods.

**Limitation and future work** Our research opens many directions for future studies. First, although the computational cost of our optimization is much less than the printing time, users might want to design and manipulate the frame structure at an interactive rate. There is much potential in speeding up our algorithm based on advanced optimization techniques. Second, we only consider small objects so that the frame can be designed and printed as a whole. For large objects which exceed the tray size of the printer, segmentation of the objects is required [Luo et al. 2012]. Assembling parts of various frame structures while maintaining its strength and stiffness appears to be possible but would be challenging. Third, the extrusion-based printers are becoming more popular and adding extra supporting structure is necessary for printing many objects with these low-end printers. Study on how to choose the upright printing direction and how to design frame structures with the least extra supporting structure for practical printability is an intriguing direction for future research.

## Acknowledgements

We would like to thank the anonymous reviewers for their constructive comments. The work is supported by the 973 Program 2011CB302400, the NSF of China (Nos. 11031007, 11171322, 61173102 and 61222206), One Hundred Talent Project of the Chinese Academy of Sciences, the 111 Project (No. b07033) and Program for New Century Excellent Talents in University (No. NCET-11-0881).

## References

- 3DSYSTEMS, 2012. Desktop 3D printer. <http://www.3dsystems.com>.
- ARYA, S., MOUNT, D. M., NETANYAHU, N. S., SILVERMAN, R., AND WU, A. Y. 1998. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM* 45, 6, 891–923.
- BÄCHER, M., BICKEL, B., JAMES, D. L., AND PFISTER, H. 2012. Fabricating articulated characters from skinned meshes. *ACM Trans. Graph. (Proc. SIGGRAPH)* 31, 4, 47:1–9.
- BENDSØE, M. P., AND SIGMUND, O. 2003. *Topology optimization: theory, methods and applications*. Springer Verlag.
- BENDSØE, M. P., BEN-TAL, A., AND ZOWE, J. 1994. Optimization methods for truss geometry and topology design. *Structural Optimization* 7, 3, 141–159.
- CALÌ, J., CALIAN, D., AMATI, C., KLEINBERGER, R., STEED, A., KAUTZ, J., AND WEYRICH, T. 2012. 3D-printing of non-assembly, articulated models. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 31, 6, 130:1–8.
- CHEN, D., LEVIN, D. I. W., DIDYK, P., SITTHI-AMORN, P., AND MATUSIK, W. 2013. Spec2Fab: a reducer-tuner model for translating specifications to 3D prints. *ACM Trans. Graph. (Proc. SIGGRAPH)* 32, 4, 135:1–135:10.
- CHEN, Y. 2007. 3D texture mapping for rapid manufacturing. *Computer-Aided Design and Applications* 4, 6, 761–771.
- CHENG, G. 1995. Some aspects of truss topology optimization. *Structural Optimization* 10, 3-4, 173–179.
- GIBSON, L. J., AND ASHBY, M. F. 1999. *Cellular Solids: Structure and Properties*, 2nd ed. Cambridge University Press.
- HUGHES, T. J. R. 1987. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Prentice-Hall.
- KIRSCH, U. 1989. Optimal topologies of structures. *Applied Mechanics Reviews* 42, 223–238.
- LUO, L., BARAN, I., RUSINKIEWICZ, S., AND MATUSIK, W. 2012. Chopper: partitioning models into 3D-printable parts. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 31, 6, 129:1–9.
- MAKERBOT, 2012. Rapid prototyping and 3D printing. <http://www.makerbot.com/>.
- NOCEDAL, J., AND WRIGHT, S. 2006. *Numerical Optimization*, 2nd ed. Springer.
- PRÉVOST, R., WHITING, E., LEFEBVRE, S., AND SORKINE-HORNUNG, O. 2013. Make it stand: balancing shapes for 3D fabrication. *ACM Trans. Graph. (Proc. SIGGRAPH)* 32, 4, 81:1–81:10.
- RAJEEV, S., AND KRISHNAMOORTHY, C. 1997. Genetic algorithms-based methodologies for design optimization of trusses. *Journal of Structural Engineering* 123, 3, 350–358.
- ROSEN, D. W., JOHNSTON, S. R., AND REED, M. 2006. Design of general lattice structures for lightweight and compliance applications. In *Proc. Rapid Manufacturing Conference*, 1–14.
- SHAPEWAYS, 2012. Tutorial: How to hollow objects for 3D printing. <http://www.shapeways.com/tutorials/creating-hollow-objects.html>.
- SMITH, J., HODGINS, J., OPPENHEIM, I., AND WITKIN, A. 2002. Creating models of truss structures with optimization. *ACM Trans. Graph. (Proc. SIGGRAPH)* 21, 3, 295–301.
- STAVA, O., VANEK, J., BENES, B., CARR, N., AND MĚCH, R. 2012. Stress relief: improving structural strength of 3D printable objects. *ACM Trans. Graph. (Proc. SIGGRAPH)* 31, 4, 48:1–11.
- TAUBIN, G. 1995. A signal processing approach to fair surface design. In *Proc. SIGGRAPH*, 351–358.
- WANG, H., AND ROSEN, D. W. 2002. Parametric modeling method for truss structures. In *Proc. ASME Design Engineering Technical Conferences*, ASME, NY, USA, 759–767.
- WANG, H., CHEN, Y., AND ROSEN, D. W. 2005. A hybrid geometric modeling method for large scale conformal cellular structures. In *Proc. ASME Design Engineering Technical Conferences*, ASME, NY, USA, 421–427.
- YAN, D.-M., LÉVY, B., LIU, Y., SUN, F., AND WANG, W. 2009. Isotropic remeshing with fast and exact computation of restricted voronoi diagram. *Computer Graphics Forum* 28, 5, 1445–1454.
- ZHOU, Q., PANETTA, J., AND ZORIN, D. 2013. Worst-case structural analysis. *ACM Trans. Graph. (Proc. SIGGRAPH)* 32, 4, 137:1–137:12.