Proceedings of the ASME 2022 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2022
August 14-17, 2022, St. Louis, Missouri, USA

DETC2022/DAC-89054

AUTOMATIC SHAPE MODIFICATION FOR SELF-SUPPORTING STRUCTURES IN ADDITIVE MANUFACTURING

Jiangce Chen
Computational Design Lab
Department of Mechanical Engineering
University of Connecticut
Storrs, CT 06269 USA
Email: jiangce.chen@uconn.edu

Matt Patterson
Computational Design Area
PARC, a Xerox Company
3333 Coyote Hill Road
Palo Alto, CA 94304 USA
Email: mpatterson@parc.com

Amir M. Mirzendehdel
Computational Design Area
PARC, a Xerox Company
3333 Coyote Hill Road
Palo Alto, CA 94304 USA
Email: amirzend@parc.com

Morad Behandish
Computational Design Area
PARC, a Xerox Company
3333 Coyote Hill Road
Palo Alto, CA 94304 USA
Email: moradbeh@parc.com

ABSTRACT

Additive manufacturing (AM) is a class of advanced manufacturing technologies that facilitates fabrication of geometrically complex, high-performance functional parts. AM processes often produce a "near-net" shape comprising the intended design and sacrificial support structures to be removed in post-processing. The amount of extra material in the near-net shape directly increases fabrication cost and clean-up time. This paper presents a shape modification method for AM parts to minimize the need for sacrificial material in the near-net shape. Given an existing triangulated boundary representation of the part, we formulate the problem as a generic shape optimization that minimizes the changes made to the original geometry by locally modifying vertex coordinates while both maintaining the manifold nature of the representation and satisfying the overhang angle constraints, when possible. Specifically, we augment a pseudo-energy regularization term to our primary objective function for self-supporting to ensure that the optimized shape is valid and suitable for AM. We propose two mesh regularization methods, using virtual springs and sphere packing, and provide differentiable expressions of the objective function and constraints to enable efficient gradient-based optimization. We show that the sphere packing is more robust and retains the quality of the mesh during optimization. Finally, we demonstrate the effectiveness and efficiency of our method in modifying realistic models in 3D.

1 Introduction

1.1 Background

The usage of additive manufacturing (AM) for part production has increased dramatically in the decades since its inception and AM technologies such as powder bed fusion, material extrusion, material jetting, and binder jetting have significantly improved manufacturing [1].

AM enables the manufacture of complicated internal geometries that are difficult or impossible to realize by subtractive methods. As such, parts designed to be manufactured by additive methods are constrained very differently than parts to be made

Copyright © 2022 by ASME

by subtractive methods. Notable examples of the benefits of AM can be found in soft robotics [2], the aviation industry [3], and bridge and house construction [4].

However, AM is not without constraints. For example, most AM methods share the problem of manufacturing parts with unsupported overhanging surfaces. Specifically, in the case of $2+1$-D methods, such as powder bed fusion and certain material-extrusion and material-jetting methods, it is common to introduce sacrificial (not required for part function) structure to support physical and/or thermal loads. Further, for almost any AM process,"islands" of material that occur when slicing must be supported by the added structure, sacrificial or otherwise. In the case of powder bed fusion, sacrificial structures may also be required to mitigate thermal gradients that could otherwise cause residual stresses and lead to cracking, curling, de-lamination, and shrinkage [5].

1.2 Motivation

In this paper, we present a method to modify the net-shape of a part, given as a boundary-represented, watertight mesh, to minimize the need for the sacrificial support structure under the constraints that

1. the part remains watertight,
2. the overhang surfaces are avoided,
3. preserves other figures-of-merit representing surface- and volume-integrity and
4. ensures the initial shape is locally, and optionally, a geometric subset of the modified shape (the initial shape is contained in the modified shape).

We propose that this method is helpful in two distinct phases of the product life-cycle: the *design phase* and the *manufacturing phase*.

A 3D part is printed in a specific orientation with respect to the normal of a planar *build plate*. Therefore, there is no notion of *overhang surfaces* requiring sacrificial support structure until this specific orientation is chosen. Our method can be used to reduce or remove the need for sacrificial support structures for a chosen orientation.

Our method generates net-shape modifications that avoid issues arising from the manufacturing process in the design phase. In addition, our method is parameterized by the overhang constraints of specific manufacturing processes. Thus, if the AM process is prescribed, our method can be used to help optimize the part for the specific process. Further, our method can be used in an *outer* optimization to down-select to a specific process. Note that both of these optimization workflows fit well within the larger engineering workflow commonly referred to as *design for manufacturing* (DfM). DfM in general, has realized benefits by, in some instances, significantly reducing both time-to-market and manufacturing cost [6].

In the manufacturing phase, removing the extra material that makes a near-net shape a net shape is not necessarily required if the support structure is added in places that do not affect part function. As our method ensures the surface- and volume-integrity of the resulting design, it is less likely to require removal. That is, our method adds as less material as possible to make the part *self-supporting* as opposed to adding support structures, for example, *pillars* or *fins*, which may be more likely to flex and/or break off.

Whatever the form of support structures, if the material must be removed by subtractive manufacturing, the net shape defined by the initial design must be a geometric subset of the near-net shape manufactured by AM, which means the near-net shape contains the net shape. Our method facilitates this by optionally allowing modifications only to be additions to the designed part.

Our method allows partitioning a part surface into multiple, distinct areas. Partitioning into multiple areas realizes computational efficiency via parallelization and better eventual results via granular application of constraints based on local functionality. Enforcing that the areas are distinct helps ensure the surface- and volume-integrity of the result. Note that our method must also avoid self-intersections to maintain integrity.

Lastly, we can formulate almost all metrics as continuous functions, allowing us to use gradient-based methods. The sole exception arises from ensuring the non-intersection of modifications from distinct partitions. Currently, our method opts for a binary but fast, yes/no check over a combinatorially explosive but continuous distance computation.

1.3 Related Work

There are many examples in the literature of generating and optimizing support structures to mitigate problems from overhang surfaces. For example, Jin et al. [7] proposes a supporting structure generation based on the polygonal overlap of *sliced* layers.Zhou et al. [8] generate tree-like supporting structure with a method based on the Lindenmayer system (L-system) and an octree.

To fully exploit the design space that AM allows, a research line is dedicated to making topology optimization (TO) generate printable results. Langelaar [9] presents a layer-wise filtering procedure that applies overhang angle restrictions at every step of the optimization process. Mirzendehdel and Suresh [10] introduced the concept of support structure topological sensitivity to reduce the amount of support structure while maximizing performance. Gaynor and Guest [11] embed an overhang projection step within the standard Heaviside Projection methodology (HPM) which ensures that structural features would form if they do not violate the overhang constraint. Thore et al. [12] introduce a penalty regulation of overhang filter approaches so that sharp inward corners could be avoided.

Copyright © 2022 by ASME

The specific methods cited above do not explicitly allow for temporary (or permanent, as applicable) part modifications to minimize the requirement for support before, or while, computing and optimizing that support. There are many articles on DfM in which an original design is modified for AM in general or a prescribed process. Mezzadri et al. [13] propose using a TO framework to minimize supports by solving a volume constrained compliance minimization. Hu et al. [14] reduce support structure by rotating overhang regions while preserving the global shape by minimizing the energy of rigidity. Luo et al. [15] presents an overhang angle control method only in enclosed voids which are identified by the nonlinear virtual temperature method. Wang et al. [16] give a TO and level set method to minimize support requirements from overhang surfaces. Their method, like ours, is formulated continuously, via the level set of a signed-distance field, and can take advantage of gradient-based approaches. However, in our understanding, their method does not currently allow for minimizing the presence of unsupported V-shapes, or *stalactites* (as denoted in this paper): see Section 2.3.

Our approach follows work that does factor in *stalactite* formation. Mezzadri and Qian [17] give a density-based method to suppress stalactites during TO, characterizing stalactites through the second derivatives of spatial density distribution. Wang et al. [18] propose a B-spline/TO-based method to minimize overhang and stalactite formation, where density values model stalactite formation at points where they may arise. Allaire et al. [19] gives a practical implicit method to minimize overhangs and stalactites by introducing a layer-based mechanical constraint.

As far as the authors are aware, the above approaches cannot partition the domain to treat different regions with different constraints.

1.4 Contributions

This paper proposes a method to optimize the shape of a mechanical part, represented as a watertight, triangulated mesh, for AM. See Figure 1 as an example. Specifically, our result is a new watertight, triangulated mesh as close as reasonable to the initial mesh that has minimal need for support structure (that is, it *self-supports*) and optionally contains the initial as a geometric subset.

We impose overhang constraints for faces, edges, and vertices of the triangle mesh. Note that we address the *stalactite*problem by imposing overhang-like constraints on edges and vertices. In addition, we optionally impose the constraint that the initial shape is a geometric subset of the modified shape by enforcing that the vertices only move in the direction of their outward normals, which is called "Outward constraint" in this paper. Furthermore, these constraints can be selectively imposed on different regions for efficiency and accommodating hybrid-manufacturing requirements. To the best of our knowledge, this is the first method that simultaneously handles the constraints mentioned above, allowing for distinct surface areas to be optimized with different constraints and objectives.

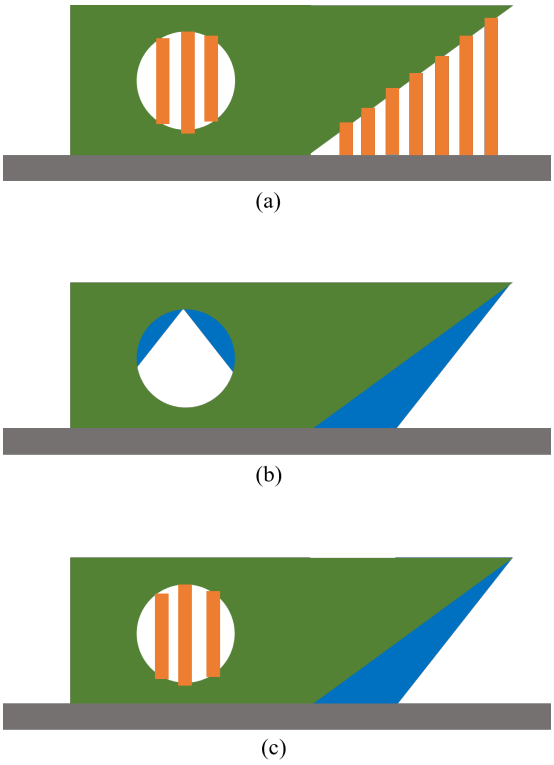


FIGURE 1. Shape modification for additive manufacturing. A common AM practice to address *overhang* surfaces is to add sacrificial support material underneath and then remove that structure after printing (a). Our proposed alternative is to modify the net-part shape to be self-supporting. The modifications are then retained or removed, depending on the effect on part function (b). Our method allows for selecting certain surface areas of the part to be modified, and different areas may be processed with different constraints and objectives (c). The supports for such regions could either be designed as scaffolds by existing methods as shown in (c) or designed as supporting crust by our method shown in (b).

In Section 2, we formulate the optimization problem of shape modification for additive manufacturing. The objective function, geometric constraints corresponding with the additive manufacturing limitations, mesh partition, and regularization methods are defined. In Section 3, we try our method under different settings. Two methods of mesh regularization, spring analog and sphere packing, are compared. In addition, we implement an efficient unconstrained optimization method for realistic models. We conclude our work in Section 4.

Copyright © 2022 by ASME

2 Optimization Formulation

For a surface triangle mesh $\mathbb{T}$, we have its associating vertex set $\mathbf{V}$, edge set $\mathbf{E}$, and face set $\mathbf{F}$. We denote the number of vertices as n_v, the number of edges as n_e, and the number of faces as n_f. For each vertex index $u = 1, 2, \cdots, n_v$, we denote its position by $\mathbf{p}_u := [x_u, y_u, z_u]^T \in \mathbf{V}$. The design variables are the coordinates of the vertices denoted by $\mathbf{x} := [x_1, y_1, z_1, ..., x_{n_v}, y_{n_v}, z_{n_v}]$.

Two vertices u and v are linked by an edge e if $e = (u, v) \in \mathbf{E}$. We do not distinguish the direction of the edge. Three vertices u, v, and w form a triangle f if $f = (u, v, w) \in \mathbf{F}$. The vertices of a face are always ordered in the counter-clock direction, which reflects the orientation of the surface. We denote the set of vertices that share an edge with u as $\mathbf{V}_u$, which is called the neighbors of u; we denote the set of edges that have u as one of their ends as $\mathbf{E}_u$; we denote the set of faces that have u as one of its nodes as $\mathbf{F}_u$.

2.1 Objective Function: Vertex Motion

We measure the changes made on the original design as the overall motions of vertices. For a vertex u, its original position is denoted as $\mathbf{p}_u^{(0)} := [x_u^{(0)}, y_u^{(0)}, z_u^{(0)}]^T$. We express the vertex motion function as:

$$f_{change}(\mathbf{x}) = \sum_{u \in \mathbf{V}} \left[(x_u - x_u^{(0)})^2 + (y_u - y_u^{(0)})^2 + (z_u - z_u^{(0)})^2 \right] / l_c^2, \tag{1}$$

where l_c is a characteristic length of the mesh, which is used to normalize the motions of the vertex. We choose l_c as the mean length over all the edges.

2.2 Face Overhang Constraints

Given a face $f_i = (u, v, w) \in \mathbf{F}$, the overhang angle is defined as the angle between its normal vector and building direction $\mathbf{b}$. The normal vector $\mathbf{n}_{f_i}$ is calculated by the positions of its three vertices as:

$$\begin{aligned} \mathbf{n}_{f_i} &= (\mathbf{p}_v - \mathbf{p}_u) \times (\mathbf{p}_w - \mathbf{p}_u) \\ &= \begin{bmatrix} (y_v - y_u)(z_w - z_u) - (z_v - z_u)(y_w - y_u) \\ (z_v - z_u)(x_w - x_u) - (x_v - x_u)(z_w - z_u) \\ (x_v - x_u)(y_w - y_u) - (y_v - y_u)(x_w - x_u) \end{bmatrix} \end{aligned} \tag{2}$$

The overhang angle of a face is illustrated in Figure 2. The face overhang constraints, denoted as $\mathbf{h} = \{h_i\}$, are formulated as:

$$h_i = \frac{\mathbf{n}_{f_i} \cdot \mathbf{b}}{||\mathbf{n}_{f_i}||} + \cos(\alpha) \geq 0, \quad \forall f_i \in \mathbf{F}, \tag{3}$$

where α is the prescribed overhang angle limit.

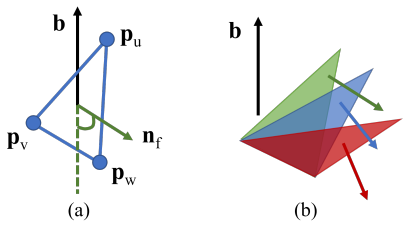


FIGURE 2. Given a face $f = (u, v, w)$, its overhang angle is defined as the angle between its normal vectors $\mathbf{n}_f$ and the opposite of the building direction $\mathbf{b}$ (a). Assume the middle (blue) face's overhang angle is the overhang limit. The upper (green) face's overhang angle is larger than the middle's, satisfying the overhang limit. While the lower (red) face's overhang angle is smaller than the middle's, violating the overhang limit (b).

2.3 Point Overhang Constraints

It was observed in [17] that only implementing the overhang constraints stated in Section 2.2 might result in "stalactites" as shown in Figure 3. Though all the faces satisfy the overhang constraints as stated in Equation (3), there are extremum points with no supporting material below them. Since the face overhang constraint only considers the normal vector of each face, there may be cases where the overhang point u is at a local minimum, while its adjacent faces $\mathbf{F}_u$ satisfy Equation (3). This paper refers to this type of overhang point as "stalactites" and denotes the stalactite vertices by $\mathbf{V}_{stalactite}$.

For a vertex u, we define its corresponding normal vector $\mathbf{n}_u$ as:

$$\mathbf{n}_u := \frac{1}{|\mathbf{F}_u|} \sum_{f \in \mathbf{F}_u} \frac{\mathbf{n}_f}{||\mathbf{n}_f||}. \tag{4}$$

As illustrated by Figure 3, $\mathbf{V}_{stalactite}$ is the set of the vertices at the local minimum of height function whose normal vectors directing downward. So, there are two coupled conditions for us to distinguish $\mathbf{V}_{stalactite}$ from the rest of the vertices in $\mathbf{V}$. Instead of formulating a compound constraint that combines the two conditions, we decouple it into two conditions.

First, we search the set of the vertices facing downward, denoted as $\mathbf{V}_{face_down}$, based on the vertex normal vectors, which is defined as:

$$\mathbf{V}_{face_down} := \{u \in \mathbf{V} \mid \frac{\mathbf{n}_u \cdot \mathbf{b}}{||\mathbf{n}_u||} \leq \gamma\}, \tag{5}$$

where γ is a real number in $[-1, 1]$. The lager γ is, the more vertices are included in $\mathbf{V}_{face_down}$, and the larger the possibility that all the formations of stalactites are restricted.

Copyright © 2022 by ASME

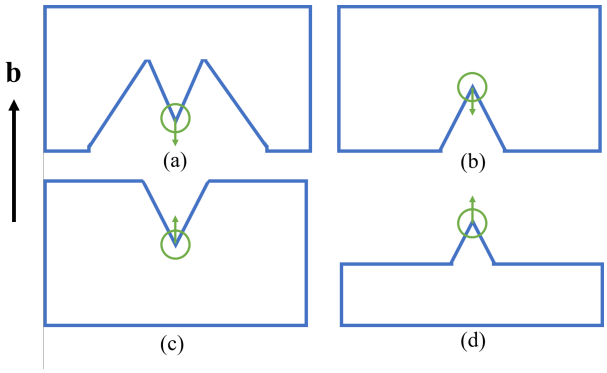


FIGURE 3. Different types of extremum points in an AM part. The circled region in (a) is the point overhang region and is not printable, while the circled points in (b)-(d) are not overhangs. In other words, the normal vectors of (a) and (b) face down, while the ones of (c) and (d) face up. Further, the overhang in (a) is at the local minimum of height function, while the one in (b) is at the local maximum of height function. Thus, we can distinguish the stalactite tip in (a) from all other cases.

Then, we impose the point overhang constraints $\mathbf{k}$ defined in Equation (6) to ensure that any vertex in $\mathbf{V}_{face_down}$ is not a local minimum among its neighbors.

$$k_i = z_{u_i} - \min_{v \in \mathbf{V}_{u_i}} z_v > 0, \quad \forall u_i \in \mathbf{V}_{face_down}. \tag{6}$$

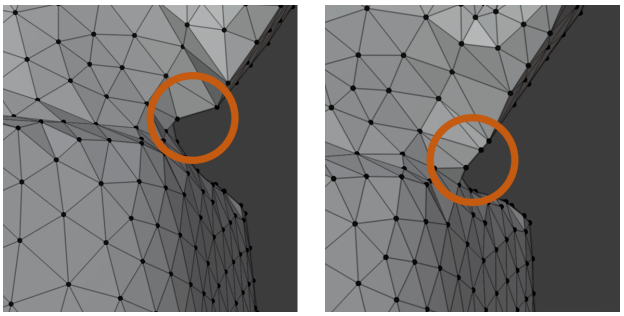


FIGURE 4. The almost horizontal edge (a);the problem is solved by considering the edge slope in the stalactite constraints (b).

2.4 Edge Overhang Constraints

Now, we have defined self-supporting constraints in Equation (3) and Equation (3) to constrain faces and vertices, respectively. However, these constraints have not covered all the self-supporting-related issues. For example, Figure 4 shows the edge overhang problem. To avoid the almost horizontal edges, we need to consider the slopes of the edges. So, we have the modified point overhang constraints Equation (6) as:

$$k_i = \max_{v \in \mathbf{V}_{u_i}} \frac{z_{u_i} - z_v}{||\mathbf{p}_{u_i} - \mathbf{p}_v||} - \sin(\alpha) > 0, \quad \forall u_i \in \mathbf{V}_{face_down}. \tag{7}$$

To ensure differentiability, we use a smooth function to approximate the maximization, and the constraint functions become:

$$\begin{aligned} \hat{k}_i &= S_\beta(u_i) > 0, \quad \forall u_i \in \mathbf{V}_{face_down}, \\ S_\beta(u_i) &= \frac{\sum_{v \in \mathbf{V}_{u_i}} \zeta_v e^{\beta \zeta_v}}{\sum_{v \in \mathbf{V}_{u_i}} e^{\beta \zeta_v}}, \\ \zeta_v &= \frac{z_{u_i} - z_v}{||\mathbf{p}_{u_i} - \mathbf{p}_v||}, \end{aligned} \tag{8}$$

where β is a positive scalar value and as $\beta \to \infty$, Equation (8) converges to Equation (7) or $\hat{k} \to k$.

2.5 Outward Constraints

In many practical applications, the net shape defined by the design (or some regions of the design) must remain unchanged due to functional or assembly requirements. In these cases, the intended design (or at least the regions of interest) must be contained in the self-supporting near-net shape produced by the AM process. The sacrificial support structures are subsequently removed by post-process machining. Such constraints could be realized by restricting the vertices' moving direction to be outward.

Let's denote the set of vertices that can only go outward as $\mathbf{V}_{outward}$. The outward constraints, denoted as $\mathbf{m}$, are formulated as:

$$m_i = (\mathbf{p}_{u_i} - \mathbf{p}_{u_i}^{(0)}) \cdot \mathbf{n}_{u_i}^{(0)} > 0, \forall u_i \in \mathbf{V}_{outward}. \tag{9}$$

It restricts that the moving direction of a vertex from its original position must have an angle less than $\pi/2$ with its original local normal vector of the surface, which means it can only move the outward of the original surface.

2.6 Topological Validity

In many applications, the meshes' quality is related to the accuracy of the quantity measured from the elements. In our case, the concerned property is the surface's normal vectors, approximated by the mesh's face orientations. Moving the vertices of the

Copyright © 2022 by ASME

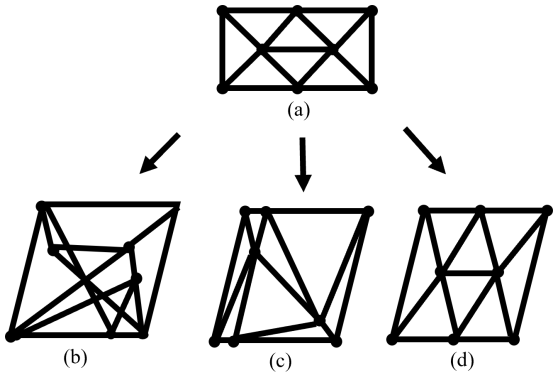


FIGURE 5. Mesh deformation examples. The original mesh (a); The deformed mesh that has self-intersecting triangles (b); The deformed mesh with low quality, which is at the edge of self-intersecting (c); The deformed mesh with high quality where the angle distortions are uniformly distributed (d).

meshes carelessly can result in low-quality, degenerate, or self-intersecting triangles that render the entire design topologically invalid, as shown in Figure 5 (b) and (c).

We introduce the pseudo-energy to hold the quality of the mesh in the form of a regularization term added in the objective function in Section 2.1. Thus, our objective function consists of vertex motion and a regularization term according to Equation (10).

$$f(\mathbf{x}) = \sum_{u\in\mathbf{V}} \left[(x_u - x_u^{(0)})^2 + (y_u - y_u^{(0)})^2 + (z_u - z_u^{(0)})^2\right]/l_c^2 + w_E E(\mathbf{V},\mathbf{E}), \tag{10}$$

where $E(\mathbf{V},\mathbf{E})$ is the pseudo-energy defined on edges, and w_E is the weight of the pseudo-energy. $E(\mathbf{V},\mathbf{E})$ would be minimized if the angle distortions are uniformly distributed over the mesh, which preserves the mesh quality, as shown in Figure 5 (d).

2.6.1 Linear Spring One traditional way to define the pseudo-energy $E(\mathbf{V},\mathbf{E})$ is by modeling edges as linear springs, where the optimal displacements are found such that the overall elastic energy of all springs is minimized. Spring analogy methods have been widely used in updating and smoothing dynamic meshes for computational models, such as fluid-structure interaction [20], free surface flow [21], and bio-fluid mechanics [22], where the domains deform and the quality of the mesh needs to be maintained. The spring balanced length is defined as the initial length of the corresponding edge. So the pseudo-energy of linear springs is defined as:

$$E_{spring}(\mathbf{x}) = \sum_{e\in\mathbf{E}} \left(\frac{l_e - l_e^{(0)}}{l_e^{(0)}}\right)^2, \tag{11}$$

where l_e is the modified length of edge e, and $l_e^{(0)}$ is the initial length of e.

2.6.2 Sphere Packing Alternatively, we can incorporate mesh regularization in the objective function of Equation (10) by using discrete conformal mapping (DCM) widely used in computational geometry to scale and rotate a shape while approximately preserving the angles[23] and other distortion measures such as edge length ratio. Some DCM methods such as Dirichlet energy minimization [24, 25, 26, 27] or Least Square Conformal Maps (LSCM) [28] require the initial and final configurations to be known a priori. Although these methods are well-established, computationally efficient, and converge to smooth conformal mappings in the limit, they are not suitable for our application since the final shape is not known beforehand.

Another kind of method does not require that the final shape is known. Instead, it defines two triangle meshes are discrete conformal equivalence if specific properties are preserved between the mesh no matter how coarse they are. For example, Springborn et al. [29] put forward that a mapping between two triangle meshes is DCM if length cross ratios are preserved. Circle packings [30] define a kind of discrete conformal maps that preserve circles, originating from the observation that smooth conformal maps preserve infinitesimal circles. A circle packing is a configuration of disjoint circles with a tangency pattern, a graph that shares the same topology as a prescribed triangulation of the surface. The circles, defined by the geodesic distance on the embodying surface, reveal the geometric information. The existence and uniqueness of circle packing have been proved by Koebe-Andreev-Thurston (KAT) theorem and its extensions [31, 32].

This paper uses sphere packing to parameterize and modify 3D surface meshes to generate optimized self-supporting near-net shapes. A sphere packing $\mathbf{S}$ consisting of a set of spheres $\{s_1, s_2, ...\}$. We build a sphere packing from a triangle mesh by placing spheres at all the mesh's vertices. The indices of the spheres are the same as the ones of vertices. The center positions of the spheres are equivalent to the vertices' positions. For a sphere s_i, we denote its center position as $\mathbf{c}_i := [x_i, y_i, z_i]^T$, and its radius as r_i. Now, we expand the design variable by including the radii of the spheres as:

$$\mathbf{x} := [x_1, y_1, z_1, r_1, ..., x_{n_v}, y_{n_v}, z_{n_v}, r_{n_v}]^{\mathbf{T}}.$$

Copyright © 2022 by ASME

We define the pseudo-energy based on sphere packing as:

$$E_{sphere}(\mathbf{x}) = \sum_{e(u,v)\in\mathbf{E}} (\frac{l_e - r_u - r_v}{l_e^{(0)}})^2. \tag{12}$$

The pseudo-energy of one edge will be zero if the spheres located at its two endpoints are tangent.

The center of each sphere is initially located at its corresponding vertex position. The initial radius is prescribed as half-length of the shortest adjacent edge. The initial conditions can be mathematically expressed as:

$$\begin{aligned} \mathbf{c}_u^{(0)} &= \mathbf{p}_u^{(0)}, \\ r_u^{(0)} &= \frac{1}{2} \min_{v\in\mathbf{V}_u} ||\mathbf{c}_u^{(0)} - \mathbf{c}_v^{(0)}||, \quad \forall u \in \mathbf{V}. \end{aligned} \tag{13}$$

Note that this initial condition ensures no pair of spheres overlapped, but it might not satisfy the tangent constraint.

2.7 Domain Classification

To keep the near-net shape as close as possible to the intended design, we need to modify the overhanging regions of an AM part locally. Figure 6 illustrates a domain classification. The regions colored in red are violating the overhang limit; the regions colored in yellow are within the prescribed neighborhood distance of the overhang regions; the gray ones are the regions supporting by the build platform which are not constrained by the overhang limit; the green regions can be printed and should not be altered.

Our shape optimization only takes the vertices belonging to the red and yellow regions into account, called 'active' vertices. Therefore, the design variables of the optimization only include the positions and radii of active vertices. On the other hand, other vertices are 'inactive,' meaning their positions and sphere radii would not be changed.

We consider an edge to be active if and only if at least one of its ends are active vertices. Similarly, a face is called active if and only if at least one of its three nodes is active. Therefore, only active vertices, edges, and faces are considered in the optimization. The inactive vertices play the role of 'boundary conditions.' Without explicitly pointing out, we always mean the active when we refer to vertices, edges, or faces.

2.8 Adaptive Constraint Tightening

From the experiments, we observed that as the initial conditions violate the overhang constraints, there is often a large discrepancy between the given design and feasible space. Thus, the optimizer might converge to a 'bad' local optimum or fail to converge. Instead of making the optimizer jump from the initial condition to the feasible space in one step, we break down the optimization process into steps and adaptively tighten the overhang constraint. In each step, a sub-optimization is solved under an increasingly more strict overhang limit until we converge to an optimized near-net shape. Note, at the beginning of each sub-optimization, the set of vertices facing downward $\mathbf{V}_{face_down}$ is updated. The possibility of missing newly formed stalactites could be diminished by reducing the step size $\Delta\alpha$ and increasing γ.

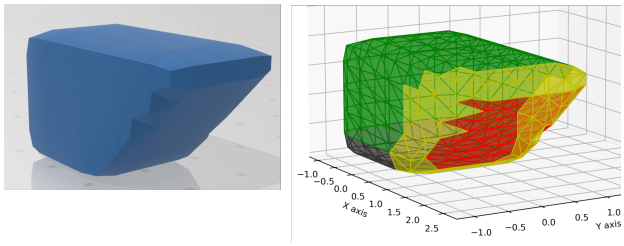


FIGURE 6. Given an initial shape and a building direction, the surface of the shape is classified into four types. The regions violating the overhang angle limit are colored in red. The regions within the prescribed neighborhood distance of the red regions are colored in yellow. The regions supported by the platform are colored in gray. The rest regions are colored in green.

The methods used to preserve mesh quality in Section 2.6 cannot prevent different parts of mesh from an intersection. A complete non-intersection constraint can be constructed by running an intersection check over all pairs of triangles. However, the quadratic time complexity of such computation would deteriorate the overall algorithm efficiency dramatically. Instead of running the intersection check at every iteration, we apply a trade-off between efficiency and reliability by introducing a binary yes/no check for the non-intersection of shape modifications at the end of each sub-optimization step.

Figure 7 illustrated the proposed shape modification workflow. The key parts of the process are:

1. **Specify the regions related to part functionality**: The vertices in these regions can only move outward during the optimization, which are collected in $\mathbf{V}_{outward}$. By doing so, the resulted near-net shape contains the net shape defined by the initial design, and the initial design could be achieved by removing the extra material added above these regions.
2. **Determine regions subject to modifications**: analyze the orientations of the triangles and collect triangles violating self-support constraints and their neighbors in prescribed levels.
3. **Maintain topological validity**: the critical part of the methodology which maintains the mesh quality along the optimization process.

Copyright © 2022 by ASME

4. **Formulate optimization**: Setup objective function and constraints for the shape optimization problem.
5. **Control the optimization process**: Decompose the overall optimization process into multiple steps where the changes on the original shape are limited to strengthen the stability and avoid global self-intersection.
6. **Check for non-intersection**: At the end of each sub-optimization, an intersection check is run over the whole domain to see if the triangle mesh has any self-intersection.

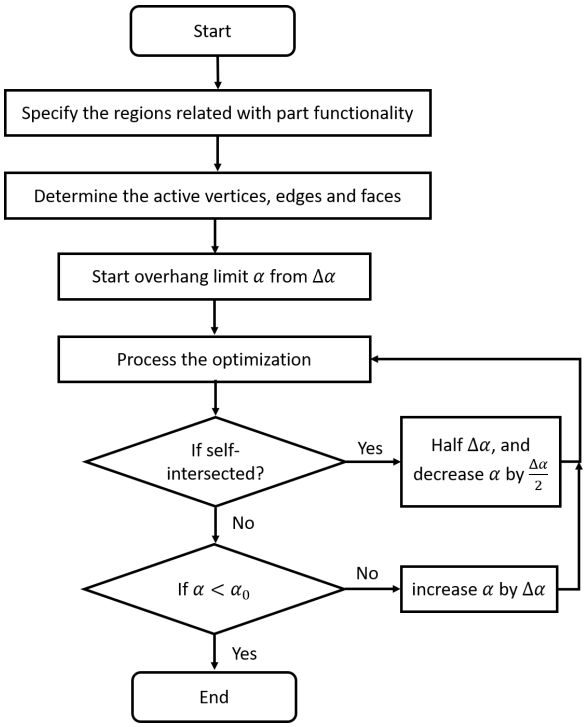


FIGURE 7. The workflow for shape optimization

2.9 Weak Form Constraints

To implement a practical version of our method, we could relax the strong forms of the constraints into weak forms, and add those weak forms into the objective function, so that the optimization becomes unconstrained. The weak form of face overhang constraint, denoted as g_{face} is defined as sum of heaviside function. So, we have

$$\begin{aligned} g_{face}(\mathbf{x},\lambda) &= \sum_{f_i \in \mathbf{F}} e^{-\xi_i} \frac{1}{1+e^{2\lambda\xi_i}}, \\ \xi_i &= \frac{\mathbf{n}_{f_i}\cdot\mathbf{b}}{||\mathbf{n}_{f_i}||} + \cos(\alpha). \end{aligned} \tag{14}$$

In a similar way, the weak form of point overhang constraint g_{point}, is defined as

$$g_{point}(\mathbf{x},\lambda) = \sum_{u_i \in \mathbf{V}_{face_down}} e^{-\hat{k}_i} \frac{1}{1+e^{2\lambda\hat{k}_i}}, \tag{15}$$

where $\hat{k}_i$ is the point overhang constraint defined in Equation 8. As we increase λ, the heaviside function approaches to sharp step function, which makes the faces that violate overhang limits have a more significant contribution in the objective function, while those satisfying the overhang limits become negligible.Now, we have the objective function as

$$f(\mathbf{x}) = f_{change}(\mathbf{x}) + w_E E(\mathbf{x}) + w_O(g_{face}(\mathbf{x},\lambda) + g_{point}(\mathbf{x},\lambda)). \tag{16}$$

where w_O is the weight for the overhang constraint.

2.10 Sensitivity Analysis

The derivatives of the vertex motion function in Section 2.1 with respect to the independent variable $\mathbf{x}$ are:

$$\nabla f_{change}(\mathbf{x}) = \frac{2}{l_c^2}[(x_1 - x_1^{(0)}), (y_1 - y_1^{(0)}), (z_1 - z_1^{(0)}), ...]^T. \tag{17}$$

In the following, we denote the part of the gradient that involves the position and radius of sphere s_u as ∇_u. The derivatives of $\hat{k}_i$ in Equation (8) against the position coordinates of $u_i \in \mathbf{V}_{face_down}$ is

$$\nabla_{u_i}\hat{k}_i = \sum_{v \in \mathbf{V}_{u_i}} \begin{bmatrix} \frac{\partial S_\beta(u_i)}{\partial \zeta_v}\frac{\partial \zeta_v}{x_{u_i}} \\ \frac{\partial S_\beta(u_i)}{\partial \zeta_v}\frac{\partial \zeta_v}{y_{u_i}} \\ \frac{\partial S_\beta(u_i)}{\partial \zeta_v}\frac{\partial \zeta_v}{z_{u_i}} \end{bmatrix}. \tag{18}$$

For a vertex $v \in \mathbf{V}_{u_i}$, the gradient is

$$\nabla_v\hat{k}_i = \begin{bmatrix} \frac{\partial S_\beta(u_i)}{\partial \zeta_v}\frac{\partial \zeta_v}{x_v} \\ \frac{\partial S_\beta(u_i)}{\partial \zeta_v}\frac{\partial \zeta_v}{y_v} \\ \frac{\partial S_\beta(u_i)}{\partial \zeta_v}\frac{\partial \zeta_v}{z_v} \end{bmatrix}, \tag{19}$$

The gradient of the outward constraint m_i in Equation (9) against the position coordinates of $u_i \in \mathbf{V}_{outward}$ is

$$\nabla_{u_i} m_i = \mathbf{n}_{u_i}. \tag{20}$$

Copyright © 2022 by ASME

The gradient of the outward constraint is exactly the vertex normal vector in the last step.

The pseudo-energy function for topology validity can be expressed generically as

$$E(\mathbf{V},\mathbf{E}) = \sum_{e\in\mathbf{E}} (g_e)^2, \tag{21}$$

where g_e is the edge pseudo-energy function defined on the coordinates and radii of the two vertices linked by edge e.

The derivatives of the pseudo-energy with respect to vertex u are

$$\begin{aligned}\nabla_u(\sum_{e\in\mathbf{E}}(g_e)^2) &= \nabla_u(\sum_{e=(u,v)\in\mathbf{E}_u}(g_e^2)\\ &= 2\sum_{e=(u,v)\in\mathbf{E}_u} g_e(\nabla_u g_e).\end{aligned} \tag{22}$$

We list the edge set explicitly as $\mathbf{E} := \{e_1, e_2, ..., e_{n_e}\}$ and assembly all the edge pseudo-energy function into one vector as:

$$\mathbf{g} = \begin{bmatrix} g_{e1} \\ \vdots \\ g_{e_{n_e}} \end{bmatrix}. \tag{23}$$

The Jacobian matrix of $\mathbf{g}$, with $n_e \times 4n_v$ dimensions, is defined as:

$$\mathbf{J_g} := \begin{bmatrix} \frac{\partial g_{e1}}{\partial x_1} & \frac{\partial g_{e1}}{\partial y_1} & \frac{\partial g_{e1}}{\partial z_1} & \frac{\partial g_{e1}}{\partial r_1} & \dots & \frac{\partial g_{e1}}{\partial x_{n_v}} & \frac{\partial g_{e1}}{\partial y_{n_v}} & \frac{\partial g_{e1}}{\partial z_{n_v}} & \frac{\partial g_{e1}}{\partial r_{n_v}} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial g_{e_{n_e}}}{\partial x_1} & \frac{\partial g_{e_{n_e}}}{\partial y_1} & \frac{\partial g_{e_{n_e}}}{\partial z_1} & \frac{\partial g_{e_{n_e}}}{\partial r_1} & \dots & \frac{\partial g_{e_{n_e}}}{\partial x_{n_v}} & \frac{\partial g_{e_{n_e}}}{\partial y_{n_v}} & \frac{\partial g_{e_{n_e}}}{\partial z_{n_v}} & \frac{\partial g_{e_{n_e}}}{\partial r_{n_v}} \end{bmatrix}. \tag{24}$$

Note that $\mathbf{J_g}$ is a sparse matrix; each row only has eight non-zero values, which correspond to the two ends of an edge.

The derivatives of $E(\mathbf{V},\mathbf{E})$ could be expressed as:

$$\nabla E(\mathbf{V},\mathbf{E}) = 2(\mathbf{g^T J_g})^\mathbf{T}. \tag{25}$$

For linear spring pseudo-energy, we have $\nabla_u g_e$ as

$$\nabla_u g_e = \frac{1}{l_e^{(0)}}\left[\frac{x_u - x_v}{||\mathbf{p}_u - \mathbf{p}_v||}, \frac{y_u - y_v}{||\mathbf{p}_u - \mathbf{p}_v||}, \frac{z_u - z_v}{||\mathbf{p}_u - \mathbf{p}_v||}, 0\right]^\mathbf{T}. \tag{26}$$

For sphere packing pseudo-energy, we have $\nabla_u g_e$ as

$$\nabla_u g_e = \frac{1}{l_e^{(0)}}\left[\frac{x_u - x_v}{||\mathbf{p}_u - \mathbf{p}_v||}, \frac{y_u - y_v}{||\mathbf{p}_u - \mathbf{p}_v||}, \frac{z_u - z_v}{||\mathbf{p}_u - \mathbf{p}_v||}, -1\right]^\mathbf{T}. \tag{27}$$

3 Experiments

3.1 Simple Demonstration

We first demonstrate our method on the simple models as shown in Figure 8. We use sphere packing method for mesh regularization. The optimization method employed is interior point method [33] implemented in Matlab. γ in Equation 5 is set as $\cos(5\pi/6)$. β in Equation 8 is set as 10. The neighborhood distance is set as 3.

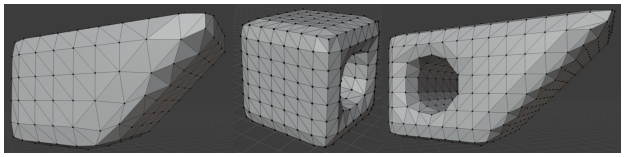


FIGURE 8. The simple models. From left to right, they are Door holder(V: 212,F: 420), Cube with hole(V: 360,F: 720), and Door holder with hole (V: 532, F: 1064).

Given a solid, as shown in Figure 9 (a), which contains one horizontal tunnel inside it and an inclined facet. Figures 9 (b)-(c) show the pre-processing steps before the shape optimization process. Let us assume the user specifies the tunnel as a region related to critical functionality, such as the pathway of fluid flows, so its net shape cannot be redesigned. To deal with the overhang problem in this tunnel region, we can only add supports under this region. On the other hand, the inclined face is not a functional surface and can be redesigned, meaning we are free to add or remove material from it. Our algorithm can generate a near-net shape for this model, which satisfies the overhang limits for AM, and a net shape that satisfies the original design requirements could be made from the near-net shape through the subtractive manufacturing process. In this example, we could enforce that the vertices near the tunnel region move outward, so the net shape of the tunnel could be made from the near-net shape by removing extra material. Figure 9 (b) shows the domain classification based on overhang regions. The red and yellow regions are subject to modification in the following shape optimization process. Figure 9 (c) shows the initial parameterization of the domain via sphere packing.

In the experiments, the overhang limit is set as 50°. These experiments are conducted on an Intel i7-10875H CPU and 16 GB RAM machine. Figures 9 (d)-(f) show the results for different shape modification strategies. Figure 9 (d) shows the modified shape without any outward constraints, where vertices can move inward or outward. Figure 9 (e) illustrates the modified shape by only apply outward constraint to the channel, while the inclined facet can be freely deformed. Figure 9 (f) shows the modified shape by applying outward constraints to both the channel and the inclined facet.

Copyright © 2022 by ASME

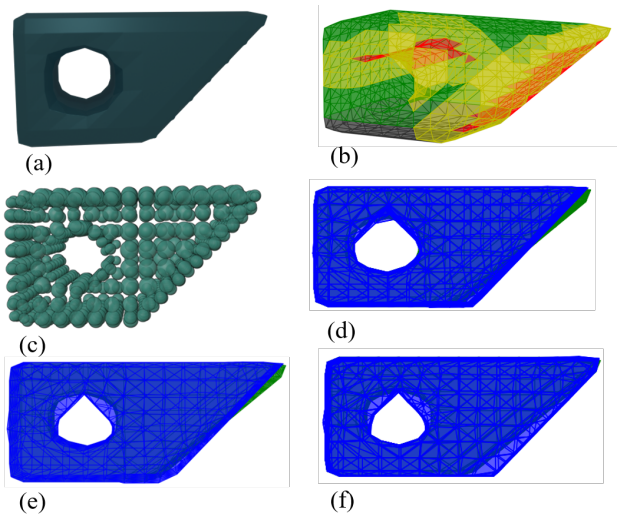


FIGURE 9. Shape modification experiments on a part with a horizontal tunnel and an inclined plain side (a); The domain classification (b); The initial sphere packing over the mesh (c); The resulted shape without outward constraint where the original mesh is colored in green, and the modified one is colored in blue (d); The resulted shape by only imposing outward constraint on the tunnel (e); The resulted shape with the outward constraint imposed on the whole domain (f).

3.2 Linear Spring Versus Sphere Packing

This section compares the performance and the computational efficiency of the two mesh regularization method: linear spring and sphere packing. As shown in Figure 10, we test on the three models in Figure 8 with different overhang limits. The horizontal axis of the plots represents the overhang limits, and the vertical axis represents the minimal angle which is defined as the smallest angle over the triangle meshes. As the overhang limit increase, more changes on the original meshes are required consequently. A triangle mesh with good quality means that the angles are distributed near 60°. As its quality deteriorates, the minimal angle in a triangle decreases. So does the minimal angle over the triangle mesh. Therefore, the minimal angle shows how much the mesh deteriorates. The more it approaches 60°, the better the mesh's quality is.

Figure 11 shows the modified solids. As we can see from the results, sphere packing regularization has a consistently better performance in maintaining the quality of mesh than linear spring regularization does.

Table 1 shows the information of the meshes and the time cost for the optimization. The optimizations that use linear spring as mesh regularization cost less time than those using sphere packing. Because sphere packing has an extra dimension, which is the radii of the spheres, to compute.

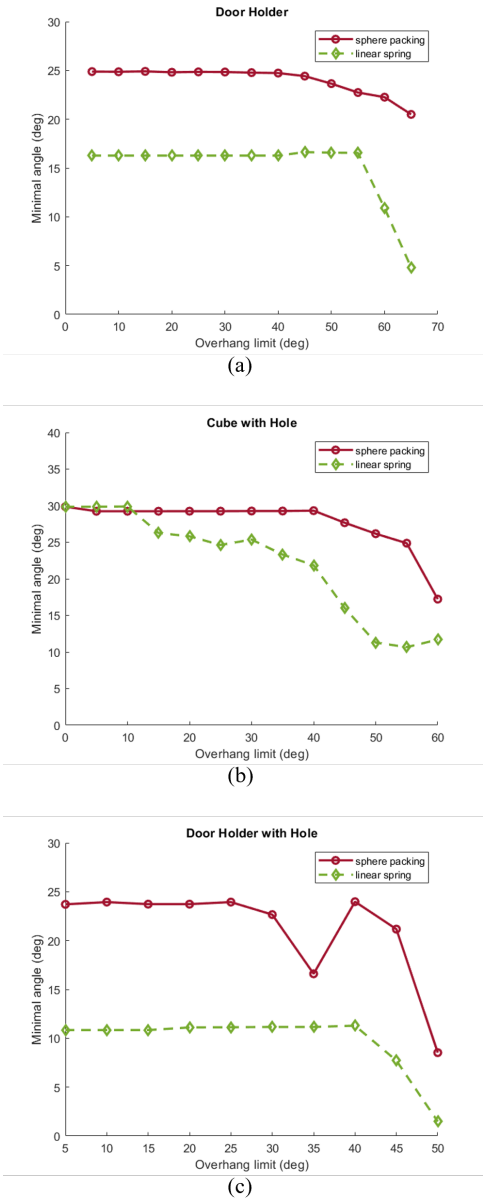


FIGURE 10. The two mesh regularization methods, sphere packing, and linear spring, are used respectively in the experiments. As the overhang limit increase, the mesh needs more modifications to satisfy the overhang limit. The minimal angle over the mesh evaluates how much the mesh deteriorates; the larger, the better.

3.3 Unconstrained Optimization

When given the realistic models, as shown in Figure 12, the constrained optimization methods used in previous experiments cannot converge in a reasonable time on an off-the-shelf desktop. Because in our problem, the initial values of variables are always infeasible, and it is estimated that the computation com-

Copyright © 2022 by ASME

Models	Sphere Packing		Linear Spring	
	sec	sec/vertex	sec	sec/vertex
Door holder	2.66	1.25e-2	1.814	8.56e-3
Cube with hole	8.74	2.43e-2	8.66	2.41e-2
Door holder with hole	99.42	1.87e-1	39.27	7.38e-2

TABLE 1. The comparison of the time cost of optimizing with sphere packing and linear spring as mesh regularization.

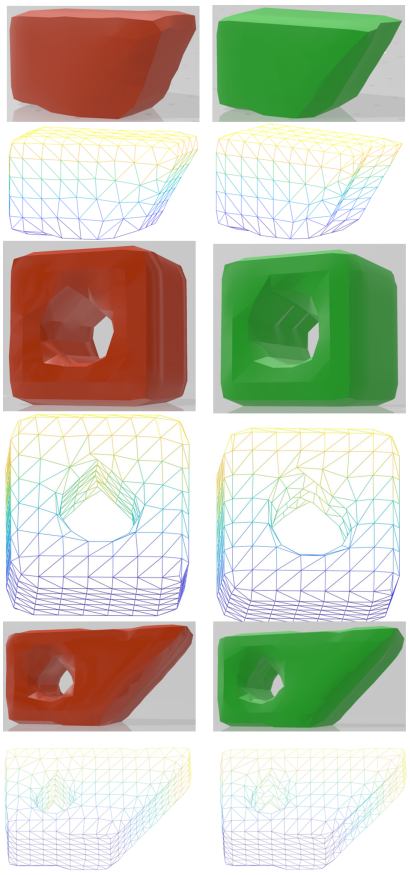


FIGURE 11. The comparison of the modified solids using the two mesh regularization methods. The red color represents the results of sphere packing, and the green color represents the results of linear spring.

plexity for the interior point algorithm to converge starting from an infeasible point is higher than quadratic [34].

Here, we implement the weak-formed constrained optimization described on section 2.9 on the realistic models. The method in Section 2.8 is used to decompose the optimization into multiple small steps and gradually increase the overhang angle limit. In the experiments, $\Delta\alpha$ set as 5°. The overhang angle limit is

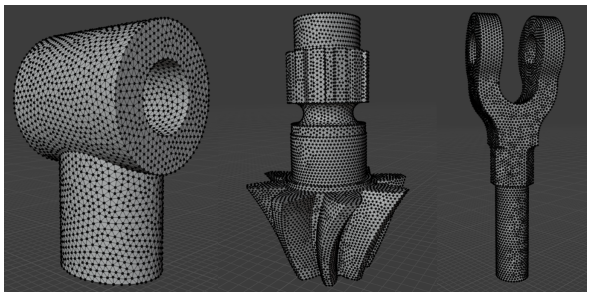


FIGURE 12. Examples of models with fine meshes. From left to right, Hair-dryer: V 4,193 F 8,386; Helix: V 13,783 F 27,566; Fork: V 7,198 F 14,400.

increased to 55°. We output the modified models under the overhang angle limits of 45°, 50°, and 55°. We gradually increase λ first in each step, then gradually increase w_O. So, the optimizer is guided to prioritize dealing with the regions with the most violation against the overhang limit. Finally, we apply sphere packing for mesh regularization. Table 2 shows the time cost.

Models	Sphere Packing	
	sec	sec/vertex
Hair-dryer	163.52	3.90e-2
Helix	152.46	1.11e-2
Fork	177.21	2.46e-2

TABLE 2. The time cost of unconstrained optimization with sphere packing as mesh regularization

The change of the loss in overhang and point overhang constraints during the optimization processes can be found in Figure 13. Since the overhang angle limits α increases at the end of each step, repetitive peaks of losses are shown in the plots. In all the experiments, g_{face} decays to zero, which means that the face overhang constraints are satisfied. While g_{point} does not converge

Copyright © 2022 by ASME

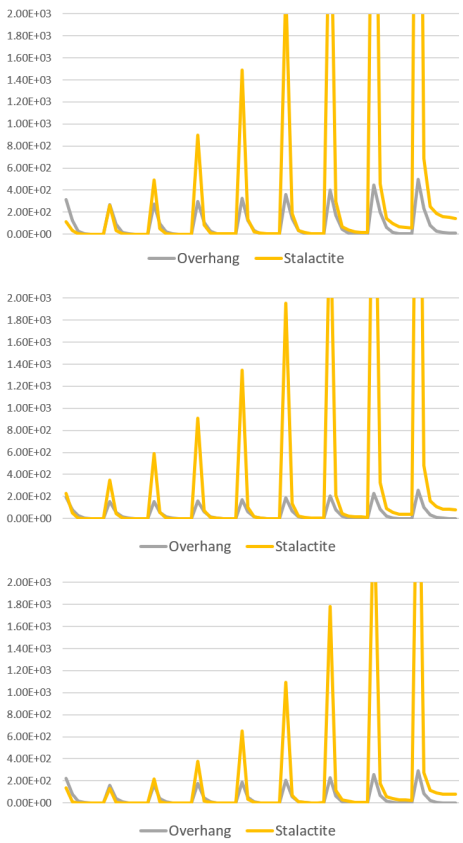


FIGURE 13. The values of the weak forms of overhang and point overhang constraints during optimization process. Each row shows the results of the computation of each model in Figure 12, which is Hair-dryer, Helix, and Fork, from top to down.

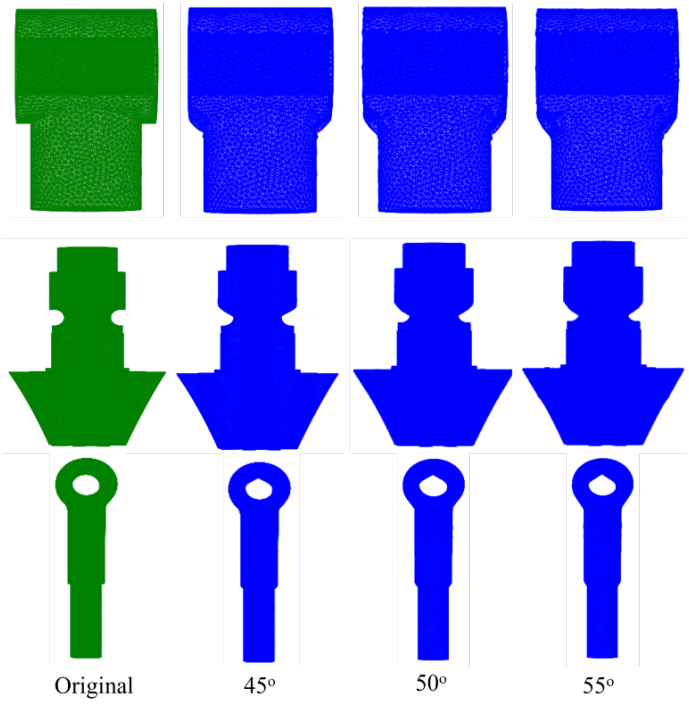


FIGURE 14. Experiments results. From the left to right, the original meshes colored in green, the modified meshes under overhang angle limits of 45°, 50°, and 55° colored in blue.

to zero, it is reduced significantly as the fact that it has larger initial value than g_{face} does. We set a larger constant parameter in the exponential term of g_{point} than that of g_{face}. In such a way, the optimizer can focus on minimizing g_{point} first, then take care of g_{face}.

Figure 14 shows the original meshes and the outputted meshes under overhang angle limits of 45°, 50°, and 55°. Figure 15 shows the original model and the modified model rendered as solids. As we can see, the modifications are reasonable and successfully avoid unprintable point overhang. However, the modified regions are not as smooth as we get by strong form of constraints with sphere packing. It indicates that the effectiveness of sphere packing is weakened to hold the quality of the mesh under the weak form of optimization. It might be because the impacts of the sphere packing constraint are dissolved as the weak form of constraints are added.

4 Conclusions

In this paper, we put forward the first method to modify the mechanical part represented by triangle mesh in accommodation with the limitations of overhang for AM. Especially our method could effectively avoid point overhang when dealing with the overhang limits. In addition, we also cover the situation when some regions on the mechanical part whose net shape cannot be changed. We can impose outward constraints over there. So we can ensure that the material is always added above these regions, and that the original design could be manufactured by removing the extra material from the near-net shape made by AM.

We implement two formulations of optimization, constrained and unconstrained, to solve the problem. The constrained formulation could cover all the AM constraints, including face, edge and point overhang, and outward constraints. However, it isn't easy to scale up to realistic models. Therefore, we transform the constrained formulation into an unconstrained one for computation efficiency by adding the weak form of the constraints into the objective function. As shown in the experiments, it could successfully handle realistic models and finish the modification within a short time. However, given the complexity of the objective function, our current formulation only includes face, edge, and point overhang constraints. To include more constraints, designing a stable hyper-parameter control strategy

Copyright © 2022 by ASME

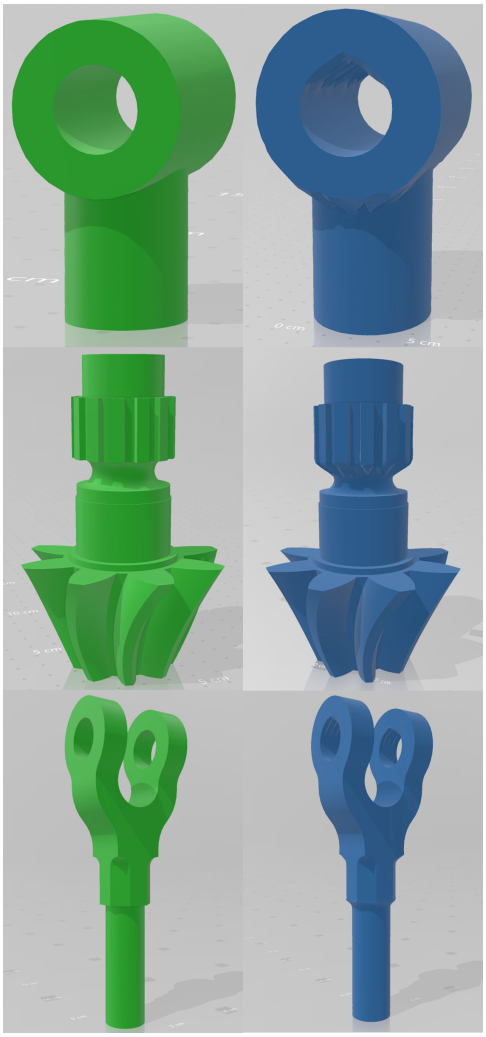


FIGURE 15. The original and modified models rendered as solids. The green ones are the original solids, and the blue ones are the modified solids with sphere packing.

would be our future work.

We study two kinds of mesh regularization methods, linear spring and sphere packing. In constrained optimization, sphere packing has a better performance in holding the quality of mesh than linear spring does, while linear spring needs less time to converge. However, sphere packing does not hold the mesh as well as it does in a constrained framework for the unconstrained formulation. It might be caused by the complexity of the objective function that overwhelms the effectiveness of sphere packing. How to effectively hold the quality of the mesh under unconstrained optimization would be another issue we would like to study in the future.

Currently, the accessibility of machining tools [35, 36] is not considered in the optimization when the material is added to the shape. In the future study, we intend to design constraints of accessibility so that the vertices can only move into the accessible field.

REFERENCES

[1] Harry Bikas, Panagiotis Stavropoulos, and George Chryssolouris. Additive manufacturing methods and modelling approaches: a critical review. *The International Journal of Advanced Manufacturing Technology*, 83(1):389–405, 2016.

[2] Nina R Sinatra, Clark B Teeple, Daniel M Vogt, Kevin Kit Parker, David F Gruber, and Robert J Wood. Ultragentle manipulation of delicate structures using a soft robotic gripper. *Science Robotics*, 4(33), 2019.

[3] Sunil C Joshi and Abdullah A Sheikh. 3d printing in aerospace and its long-term sustainability. *Virtual and Physical Prototyping*, 10(4):175–185, 2015.

[4] Craig Buchanan and Leroy Gardner. Metal 3d printing in construction: A review of methods, research, applications, opportunities and challenges. *Engineering Structures*, 180: 332–348, 2019.

[5] Jingchao Jiang, Xun Xu, and Jonathan Stringer. Support structures for additive manufacturing: a review. *Journal of Manufacturing and Materials Processing*, 2(4):64, 2018.

[6] Aboma Wagari Gebisa and Hirpa G Lemu. Design for manufacturing to design for additive manufacturing: Analysis of implications for design optimality and product sustainability. *Procedia Manufacturing*, 13:724–731, 2017.

[7] Yu-an Jin, Yong He, and Jian-zhong Fu. Support generation for additive manufacturing based on sliced data. *The International Journal of Advanced Manufacturing Technology*, 80(9):2041–2052, 2015.

[8] Yong Zhou, Han Lu, Qingrong Ren, and Yang Li. Generation of a tree-like support structure for fused deposition modelling based on the l-system and an octree. *Graphical Models*, 101:8–16, 2019.

[9] Matthijs Langelaar. Topology optimization of 3d self-supporting structures for additive manufacturing. *Additive manufacturing*, 12:60–70, 2016.

[10] Amir M Mirzendehdel and Krishnan Suresh. Support structure constrained topology optimization for additive manufacturing. *Computer-Aided Design*, 81:1–13, 2016.

[11] Andrew T Gaynor and James K Guest. Topology optimization considering overhang constraints: Eliminating sacrificial support material in additive manufacturing through design. *Structural and Multidisciplinary Optimization*, 54(5): 1157–1172, 2016.

[12] C-J Thore, H Alm Grundström, Bo Torstenfelt, and Anders Klarbring. Penalty regulation of overhang in topology optimization for additive manufacturing. *Structural and Multidisciplinary Optimization*, 60(1):59–67, 2019.

Copyright © 2022 by ASME

[13] Francesco Mezzadri, Vladimir Bouriakov, and Xiaoping Qian. Topology optimization of self-supporting support structures for additive manufacturing. *Additive Manufacturing*, 21:666–682, 2018.

[14] Kailun Hu, Shuo Jin, and Charlie CL Wang. Support slimming for single material based additive manufacturing. *Computer-Aided Design*, 65:1–10, 2015.

[15] Yunfeng Luo, Ole Sigmund, Quhao Li, and Shutian Liu. Additive manufacturing oriented topology optimization of structures with self-supported enclosed voids. *Computer Methods in Applied Mechanics and Engineering*, 372: 113385, 2020.

[16] Yaguang Wang, Jincheng Gao, and Zhan Kang. Level set-based topology optimization with overhang constraint: Towards support-free additive manufacturing. *Computer Methods in Applied Mechanics and Engineering*, 339:591–614, 2018.

[17] Francesco Mezzadri and Xiaoping Qian. A second-order measure of boundary oscillations for overhang control in topology optimization. *Journal of Computational Physics*, 410:109365, 2020.

[18] Che Wang, Weihong Zhang, Lu Zhou, Tong Gao, and Jihong Zhu. Topology optimization of self-supporting structures for additive manufacturing with b-spline parameterization. *Computer Methods in Applied Mechanics and Engineering*, 374:113599, 2021.

[19] Grégoire Allaire, Charles Dapogny, Alexis Faure, and Georgios Michailidis. Shape optimization of a layer by layer mechanical constraint for additive manufacturing. *Comptes Rendus Mathematique*, 355(6):699–717, 2017.

[20] Frederic Blom, Penelope Leyland, Frederic Blom, and Penelope Leyland. Analysis of fluid-structure interaction on moving airfoils by means of an improved ale-method. In *28th Fluid Dynamics Conference*, page 1770, 1997.

[21] Blair Perot and Ramesh Nallapati. A moving unstructured staggered mesh method for the simulation of incompressible free-surface flows. *Journal of computational physics*, 184(1):192–214, 2003.

[22] Dehong Zeng and C Ross Ethier. A semi-torsional spring analogy model for updating unstructured meshes in 3d moving domains. *Finite Elements in Analysis and Design*, 41(11-12):1118–1139, 2005.

[23] Albert Chern, Ulrich Pinkall, and Peter Schröder. Close-to-conformal deformations of volumes. *ACM Transactions on Graphics (TOG)*, 34(4):1–13, 2015.

[24] Xianfeng Gu, Yalin Wang, Tony F Chan, Paul M Thompson, and Shing-Tung Yau. Genus zero surface conformal mapping and its application to brain surface mapping. *IEEE transactions on medical imaging*, 23(8):949–958, 2004.

[25] Patrick Mullen, Yiying Tong, Pierre Alliez, and Mathieu Desbrun. Spectral conformal parameterization. In *Computer Graphics Forum*, volume 27, pages 1487–1494. Wiley Online Library, 2008.

[26] Wei-Qiang Huang, Xianfeng David Gu, Wen-Wei Lin, and Shing-Tung Yau. A novel symmetric skew-hamiltonian isotropic lanczos algorithm for spectral conformal parameterizations. *Journal of Scientific Computing*, 61(3):558–583, 2014.

[27] Mei-Heng Yueh, Wen-Wei Lin, Chin-Tien Wu, and Shing-Tung Yau. An efficient energy minimization for conformal parameterizations. *Journal of Scientific Computing*, 73(1): 203–227, 2017.

[28] Bruno Lévy, Sylvain Petitjean, Nicolas Ray, and Jérome Maillot. Least squares conformal maps for automatic texture atlas generation. *ACM transactions on graphics (TOG)*, 21(3):362–371, 2002.

[29] Boris Springborn, Peter Schröder, and Ulrich Pinkall. Conformal equivalence of triangle meshes. In *ACM SIGGRAPH 2008 papers*, pages 1–11. 2008.

[30] Charles R Collins and Kenneth Stephenson. A circle packing algorithm. *Computational Geometry*, 25(3):233–256, 2003.

[31] Kenneth Stephenson. *Introduction to circle packing: The theory of discrete analytic functions*. Cambridge University Press, 2005.

[32] Philip L Bowers. Combinatorics encoding geometry: the legacy of bill thurston in the story of one theorem. In *In the Tradition of Thurston*, pages 173–239. Springer, 2020.

[33] Richard H Byrd, Mary E Hribar, and Jorge Nocedal. An interior point algorithm for large-scale nonlinear programming. *SIAM Journal on Optimization*, 9(4):877–900, 1999.

[34] Florian A Potra and Stephen J Wright. Interior-point methods. *Journal of computational and applied mathematics*, 124(1-2):281–302, 2000.

[35] Amir M Mirzendehdel, Morad Behandish, and Saigopal Nelaturi. Topology optimization with accessibility constraint for multi-axis machining. *Computer-Aided Design*, 122:102825, 2020.

[36] Amir M Mirzendehdel, Morad Behandish, and Saigopal Nelaturi. Topology optimization for manufacturing with accessible support structures. *Computer-Aided Design*, 142: 103117, 2022.

Copyright © 2022 by ASME