A POST-PROCESSING PROCEDURE FOR LEVEL SET BASED TOPOLOGY OPTIMIZATION

Y. Xian, and D. W. Rosen*

George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA 30332

* Corresponding author. Email: david.rosen@me.gatech.edu

Abstract

This paper addresses two issues: 1. Topology optimization yields designs that may require support structures if additively manufactured, which increase material and clean-up costs. 2. Topologically optimized designs consist of discretized geometry which makes subsequent engineering difficult, hence the increasing need to somehow render TO results to parameterized CAD models. This paper presents a procedure that, after a standard level set based topology optimization, firstly identifies certain regions on the part boundary that may require support materials or may cause staircase effect during 3D printing, then replaces these boundary segments with similar-shaped printable design features. Additionally, other boundary regions are fitted with simple geometric entities, so that the part boundary can be completely defined by geometric parameters of design primitives.

Introduction

Additive manufacturing (AM) refers to processes in which 3D objects are built layer by layer under computer control. Compared to traditional manufacturing processes, where material is "subtracted" from a block of material, AM fabricates highly complex parts more easily.

Topology optimization (TO) redistributes material in a design space, based on sensitivity analysis. Being a discrete method itself, topology optimization uses the finite element method to determine the objective function value at each iteration. After the topology optimization, the structure usually needs to go through a series of post-processing, for example, smoothing and/or a subsequent shape optimization, because topology optimization results cannot be used as is for design purposes.

There is significant interest today in integrating additive manufacturing and topology optimization, for several reasons, a major one being that topology optimization puts material only where it's needed, which produces light-weight results that help save weight a great deal in AM. Weight saving is a benefit during the product's useful life. For example, weight savings for a part in an aircraft can save thousands of gallons of fuel, millions of dollars, and greatly reduce pollution. Secondly, topologically optimized designs are geometrically complex, which are hard to manufacture using traditional processes, but could often be easily additively manufactured.

Though AM gives much freedom in the designs, making proper use of topology optimization to render additively manufacturable designs remains a challenging and vibrant

research area for the past decade. Some of the common challenges to additively manufacture certain TO structures are, for example, building designs with overhangs requires support structures, which directly increase build time and material cost; also, machines have minimum allowable feature size limitations which TO results may not meet. Another issue we want to address in this work is that, topologically optimized designs consist of discretized geometry and/or implicit boundary representation, which makes subsequent design and manufacturing difficult due to the lack of parametric surfaces and dimensions.

Driven by the above challenges, we propose a post-processing procedure for topologically optimized structures, that gives the two-dimensional discretized geometry an explicit boundary with smooth parametric curves, which designers could easily work with for subsequent engineering. A second objective of this work is that overhangs are removed from the TO designs. A third objective is to introduce parametric features into the structure to achieve a certain volume objective.

Literature Review

Topology optimization

In the literature, one can find a multitude of approaches for solving topology optimization problems. The following is a brief review of some of the key approaches.

One popular approach called homogenization method was firstly proposed by Bendsøe and Kikuchi (1988) [1]. The main idea of the homogenization method is to introduce periodically distributed micro-scale voids in a given homogeneous material. The effective material properties of the composite are then computed using homogenization theory. In this way, the material layout problem could be treated as an easier sizing problem, the sizing variable being the density of the perforated composite (i.e. sizes of holes). However, homogenization method often produces designs with infinitesimal pores that make the structure non-manufacturable.

One variation of the homogenization method later investigated was the SIMP approach (Solid Isotropic Material with Penalization), firstly by Bendsøe (1989) [2]. In SIMP, elements' relative densities are the design variables, which could be updated using several updating schemes, such as OC (Optimality Criteria), SLP (Sequential Linear Programming) methods, and MMA (Method of Moving Asymptotes). To penalize the intermediate densities, material properties are modeled to be proportional to the relative density raised to some power [3].

An alternative class of methods to the above density-based approaches is the boundary-based methods for structural optimization [4], a major approach of which is to use level set method to represent implicit, moving boundaries for topology optimization.

Level set method was first used by Osher and Sethian (1988) [5] as a method to implicitly represent the moving interfaces. In such analyses, the equations of motion of propagating fronts are approximated by Hamilton-Jacobi equation. It was later applied in topology optimization to track the structural boundary, using an appropriate velocity normal to the boundary interface. Design variables are usually level set function values at the nodes. Level set based TO will be presented in more detail in the next section.

Reverse engineering

Generating a CAD model from topology optimization result for subsequent design and engineering remains a challenge. For that, various techniques of surface reconstruction and feature detection have been proposed [6, 7, 8]. The level set based TO involves essentially converting a point cloud to a CAD model, which corresponds mostly to geometric segmentation. A brief overview of the literature related to geometric segmentation is provided below.

Geometric segmentation methods organize a triangular mesh into clusters that correspond to surface features, so as to identify features out of discrete geometric models. One main approach evaluates the discontinuities of surfaces in the point cloud; this "edge-based approach" can directly identify the "sharp corners", but would possibly miss geometric features along a relatively smooth transition [6]. In contrast to the above method which analyses the data points and detects "corners", a different type of approach, called "region-based approach", identifies geometric features in a zone-by-zone fashion. Starting with a set of "seed" points, this method involves growing regions by appending to each seed the neighboring pixels that have similar properties, such as specific ranges of gray level.

Level Set based Topology Optimization

Based on the objective of this work, which is to render an explicit boundary for TO result, we chose the level set method for TO over density based approaches, because level set models are more precise in describing the boundary shape of the structure [4].

Implicit representations of shapes have a long tradition in geometric modeling and computer graphics. Level set function is able to describe shapes implicitly, among many other representations used in geometric modeling [9]. For a general 3D structure with surface boundaries, a level set model specifies a surface in an implicit form as an iso-surface of a scalar function, ϕ: R^3 -> R, embedded in 3D, i.e.,

$$S = \{x : \Phi(x) = k\}$$

For topology optimization, part boundary is implicitly represented by the zero iso-lines of a level set function; the material domain could be defined where the level set function has a positive, or negative, value. For example, Figure 1(b) shows a three-dimensional level set function field. A cutting-plane of an arbitrary value could be used to extract a topology; using a zero-level cutting plane gives the two-dimensional shape in Figure 1(a). In this case, function values larger than zero are defined to be material region.

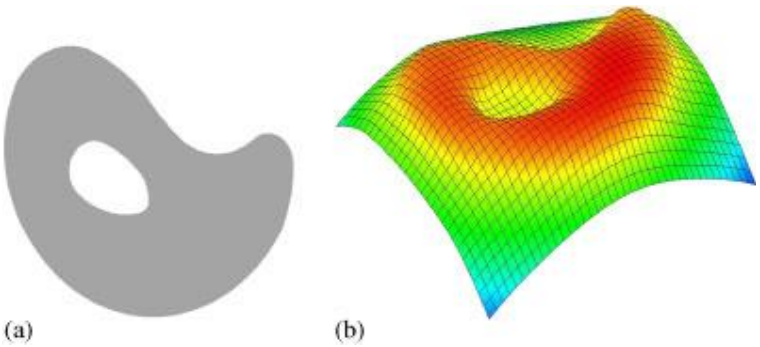

Fig. 1 Implicit representation of a shape using level set function.

In the optimization algorithm of a conventional level set method for topology optimization, evolution of the structural boundary is tracked by solving a Hamilton-Jacobi equation [4]:

$$\frac{\partial \Phi(x,t)}{\partial t} + \nabla \Phi(x,t) \frac{\mathrm{d}x}{\mathrm{d}t} = 0$$

As shown in the above equation, ϕ is the level set function; velocity dx/dt is chosen based on the shape sensitivity of the objective function. Additionally, to allow potential nucleation of new voids, another term that takes into account the topological sensitivity needs to be added in the evolution equation.

An innovative work for level set based TO was conducted by Otomori, Yamada and others in 2014 [10], that used a reaction diffusion equation to track the boundary. With this method, the geometrical complexity of optimized configurations can be controlled by adjusting a regularization parameter; it allows potential topological changes as well.

For this work, we used level set method with this newly developed evolution equation to update the part boundary.

Post-Processing Procedure

Procedure overview

Steps of the post-processing procedure are as follows:

1. Conduct level set based topology optimization to obtain the implicit boundary result.
2. Identify corner points on the implicit geometry.
3. Generate an explicit boundary representation by replacing the boundary with geometric entities.
4. Eliminate overhangs by adjusting or removing the overhanging edges.
5. Add diamond-shaped features to satisfy a prescribed volume constraint.

A design problem

Post-processing methods were investigated upon an example problem from Otomori's work and their topology optimization result [10]. This is a classic case of a compliance minimization problem with a volume constraint of 0.5.

Figure 2 shows the design domain and boundary conditions of the problem. Displacement is fixed at the left boundary, and a downward force is applied at the center of the right boundary. The design domain is discretized using quadrilateral elements.

Figure 3 shows the optimized configuration using an 80-by-64 mesh. Note that the binary image is merely a binary representation of the structure; it's the zero level-set contours that represent the true boundary.

Fig. 2 Design domain and boundary conditions of a TO problem.

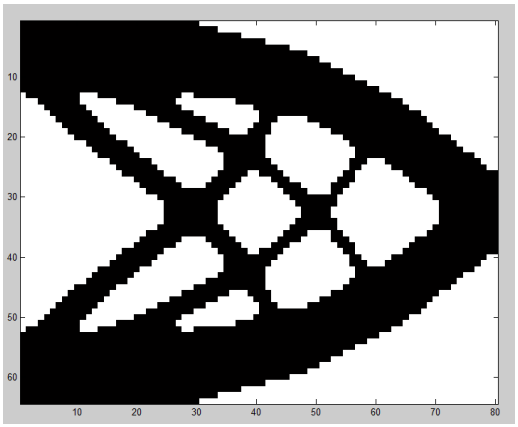

Fig. 3 Optimized configuration using an 80 * 64 mesh.

Corner detector

To perform post-processing on the part boundary, we extracted firstly the zero level-set contours from the topology optimization output, i.e. nodal values of the level set function; again, the true boundary obtained directly from TO is formed by points that lie on the cutting plane of value zero of the level set function field, as shown in Figure 4.

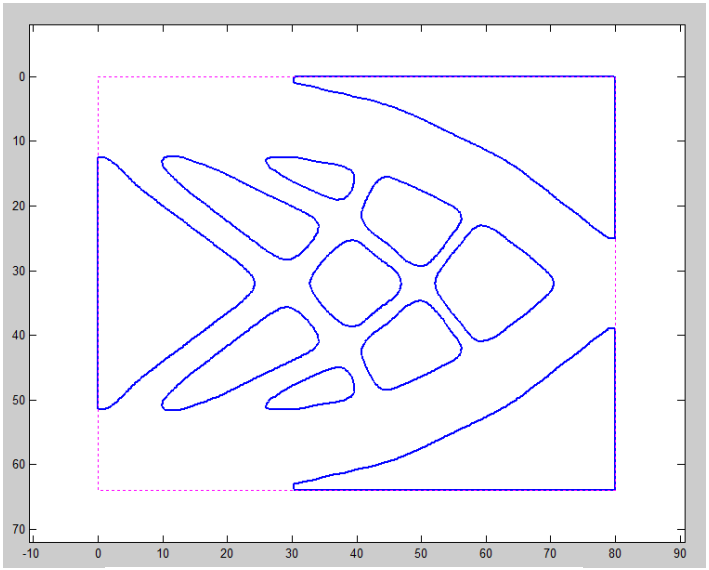

Fig. 4 Zero-value level set contours.

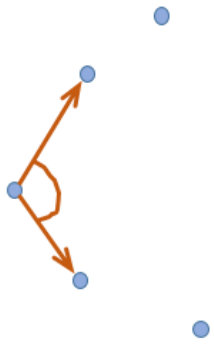

Fig. 5 A local view of the angle calculation step.

Before replacing the implicit contours in Figure 4 with geometric entities, we identified their corner points through three steps:

Step 1: Angle threshold

At each point on a zero-level contour, the angle between the two vectors formed by this point and its two adjacent points was calculated. Any point where the angle is larger than a threshold is picked out. Figure 5 zooms in on a few points on the contour, and demonstrates this angle calculation step.

Step 2: Density control

Step 1 could be selecting "corners" at a local scale, which means more corner points could have been selected than necessary. For example, in Figure 6, points selected at step 1 are highlighted in pink; it shows that multiple points were picked out where only one true "corner" exists in a zoomed-out point of view. Thus, we introduced corner density control, which detects the number of selected corner points within any small disk area; if more than one, then only one corner point stays.

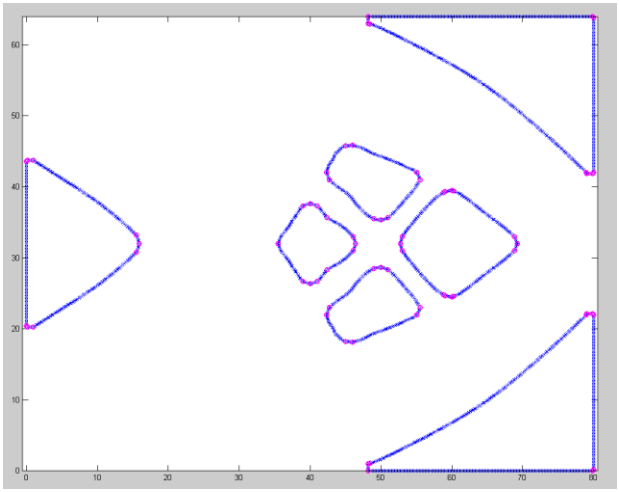

Fig. 6 Multiple "corner points" in close neighborhoods.

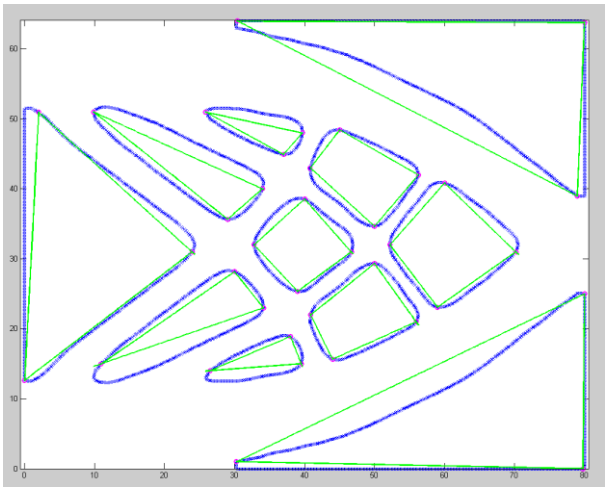

Fig. 7 A corner detector result that requires final clean-up.

Step 3: Final clean-up

In step 2, selecting one point in a close neighborhood was done randomly. Then situations such as shown in Figure 7 may occur, where sharp angles are formed with the border of the design space. Thus, as a final clean-up, the vertices whose distances to the design frame were smaller than a threshold were moved onto the frame.

After the corner points (or vertices) were detected from the above three steps, the zero-level contours in Figure 4 could be parameterized by connecting the vertices with geometric entities which fit the original shapes as precisely as possible, so that the implicit boundary representation could be converted to an explicit one with dimensional parameters. In this work, only straight lines were used to replace the boundary for simplicity.

Figure 8(b) was obtained as an approximate part boundary, which is represented explicitly by dimensional parameters, in this case, slopes and intercepts of the line segments (edges).

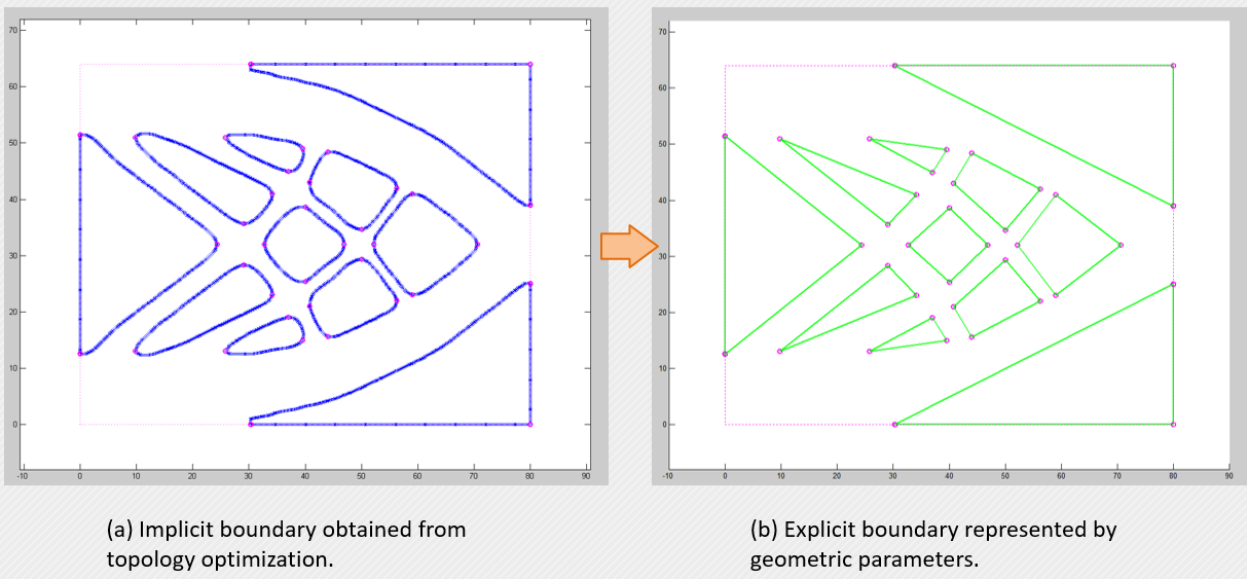

Fig. 8 Conversion of implicit boundary to parametric geometry.

Overhang removal

The next step is to remove the overhanging edges from the explicit boundary in Figure 8(b). Remind that material region lies outside the polygonal contours.

We know that the minimum angle threshold for overhangs may be different for each 3D printer. For validation purpose, one FDM machine was chosen in advance, with which we conducted overhang angle tests, and determined its minimum build angle was 50°.

The method that we adopted to remove overhangs was indicated in Figures 9-11, where edges in black and red form the original polygon; edges in red indicate overhanging edges, while edges in green would replace the red ones for eliminating overhangs. Overhanging edges were categorized into two types: slanted overhangs and horizontal overhangs.

Take the slanted overhanging edge 2-3 in Figure 9 as an example. The edge is adjusted as follows: Move the lower vertex 2 of the overhanging edge 2-3 towards its other adjacent vertex 1, along its (non-overhanging) connecting edge 2-1, until the modified edge 2'-3 forms a 50° with horizontal axis.

Note that certain polygons have to be completely removed based on this method; taking the polygon in Figure 10 as an example, vertex 3 would need to be positioned on the other side of the lower edge 1-2 of the original polygon, to not form an overhang. This means polygon 1-3'-2 cannot exist.

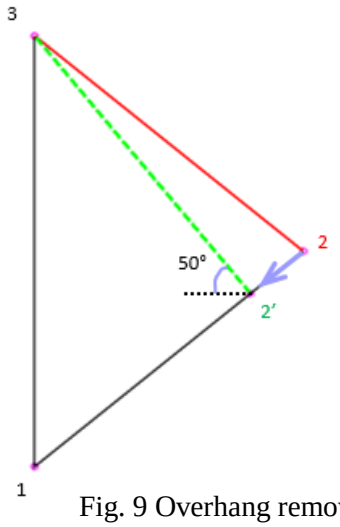

Fig. 9 Overhang removal step of a slanted overhanging edge.

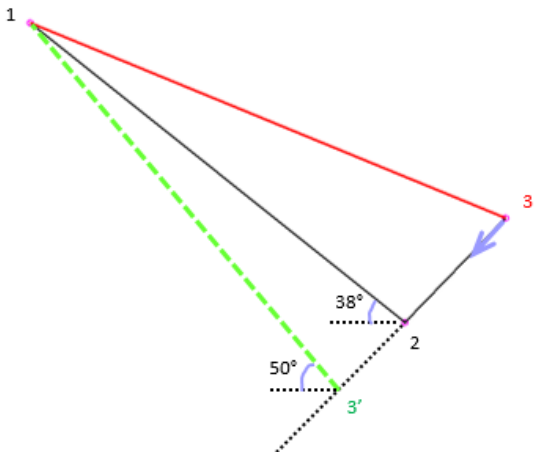

Fig. 10 A special case of the slanted overhangs, where the polygon is removed.

As for horizontal overhangs as shown in Figure 11, a point on the overhanging edge 1-2 is picked, from which two edges are drawn that both form 50° with the horizontal axis, until they meet other (non-overhanging) edges of the polygon.

The process of modifying horizontal overhangs is yet to be improved, because picking the top point was done randomly in this work. There should be an optimal point that makes volume loss of the polygon as low as possible, which would be investigated in future work.

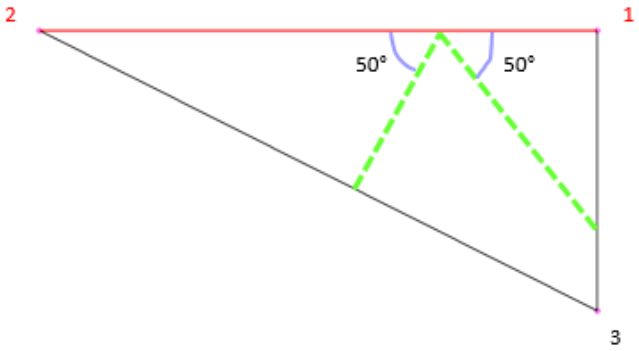

Fig. 11 Overhang removal step of a horizontal overhanging edge.

In this way, all the new polygons (without overhangs) are guaranteed to be contained inside their original ones, so that edges of one polygon would not intercept with another. This is illustrated in Figure 12(b), where all the overhanging edges highlighted in red in Figure 12(a) have been adjusted or removed.

Though the overhang removal process was simple, an obvious flaw was that the new polygons as in Figure 12(b) are always smaller than, or at most the same as, the polygons with overhangs as shown in Figure 12(a). Note that two of them were removed completely.

As a consequence, we noticed a volume increase of as high as 30% of this new design, compared to the explicit boundary with overhangs, due to the fact that polygon sizes are inevitably reduced during overhang removal.

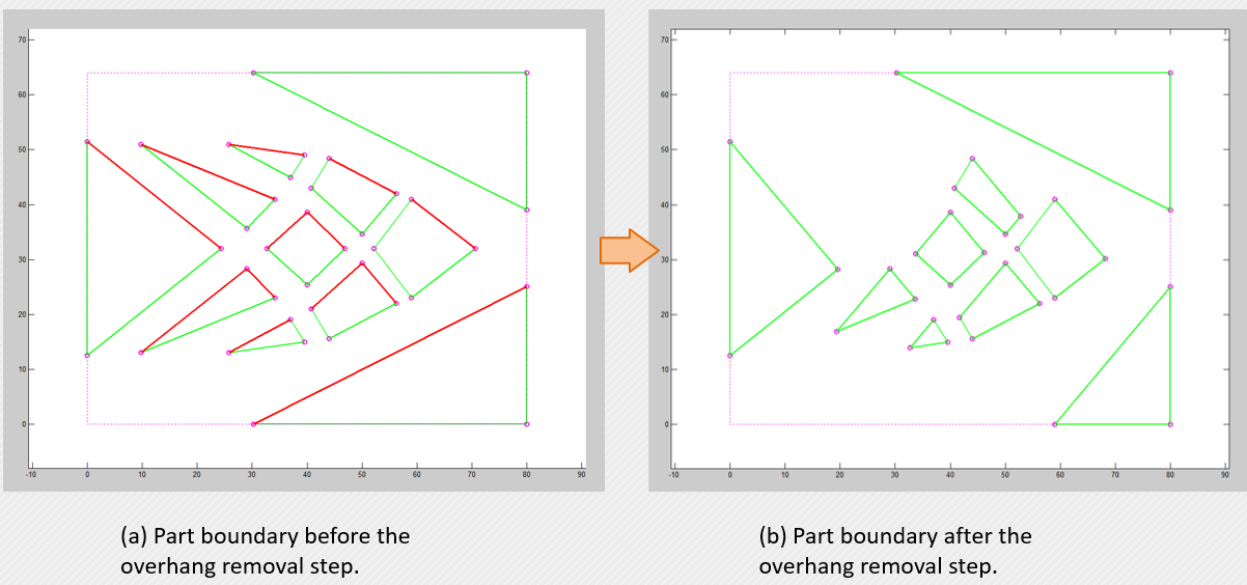

Fig. 12 Overhang removal step: before and after.

The huge volume change after overhang removal poses a problem, because as a future step, we would like to integrate the entire post processing into topology optimization, so that the topologically optimized part boundary comes out directly formed by non-overhanging parametric curves.

This brought the third objective of this work mentioned in the Introduction: Part volume after post-processing should remain within a small margin of error (6% for this work) with the TO volume constraint.

Volume control

Our solution to controlling the part volume was to insert diamond-shaped features as void regions after the overhang removal step.

As shown in Figure 13, the diamond-shaped feature is essentially a parallelogram; its four edges are of the same length, all forming a 50° angle with the horizontal axis. Remind that 50° is the minimum build angle of the FDM machine used for validation. Then the feature is controlled by three parameters: center point coordinates (x_c, y_c), and half height H.

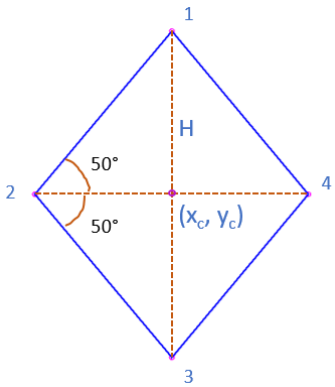

Fig. 13 Diamond-shaped design feature.

The procedure of the volume control step is described as follows:

First off, the volume loss of each polygon was tracked during the overhang removal step.

Then, regions where the relative volume loss was highest among all the polygons were picked out. In this case study, the two regions shaded in blue in Figure 14 were returned, whose total area made up about 90% of the total volume increase during the overhang removal process.

These shaded regions are returned in the form of polygons, whose vertices are those of the original polygons prior to the overhang removal step. The shaded region would be returned as a polygon enclosing the lost area. Note that the upper shaded polygon was constructed to enclose both polygons that vanished, as they were located close to each other.

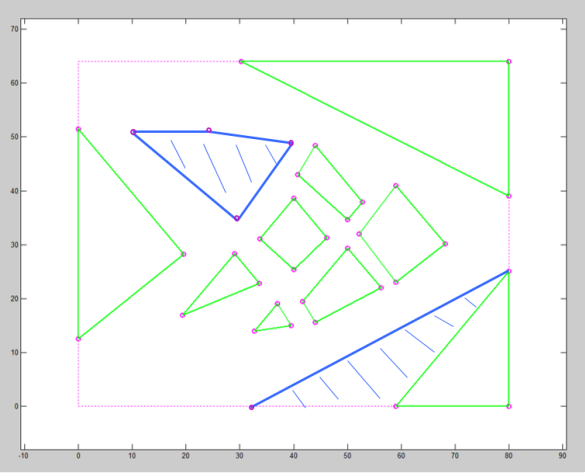

Fig. 14 The two blue regions indicate areas with the highest volume change.

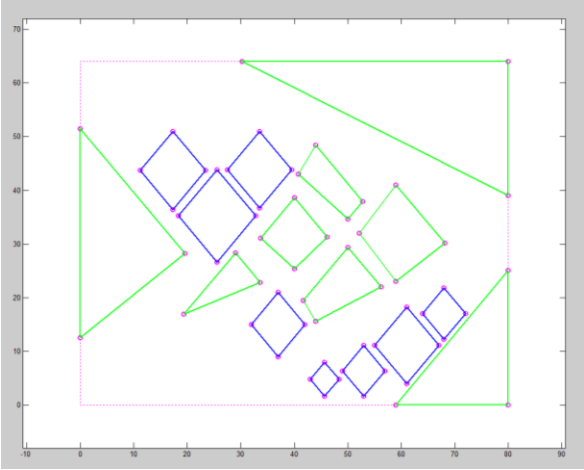

Fig. 15 Part boundary after diamond shapes were inserted.

Next step was to locate the center point for each diamond feature to be inside the shaded regions; the edge lengths were then fixed based on volume calculation. In the end, the part volume must be controlled within (0.5 + 6%).

The configuration as shown in Figure 15 was generated as the post-processing result. Several advantages of the diamond-shaped feature were observed: The fact that the edges of the diamond shapes are parallel to each other makes it easier to position them without intersecting with other polygons. Furthermore, diamond shapes form thin walls between each other. For validation, we made the distance between two adjacent diamond shapes to be a little above the minimum allowable feature size of the FDM machine. After we tested the FDM machine used in this work, and determined the minimum wall thickness it could build was 0.6 mm, a thickness of 1.5 pixel widths was selected for the thin members formed between two diamonds; remind that the size of the design space is 80 * 64 pixels. We believe this FDM machine could safely build thin members of 1.5 pixel widths, because it is at least twice its minimum feature size (without units). The reason why diamond shapes are kept close to one another is that, in stiffness maximization problems for cantilever structures, a structure with an infinite number of infinitely thin members is known to be optimal.

Validation

The first step of validation compares the compliance and volume of each configuration that has been discussed above. The comparison is summarized in Table 1. Plots of all four configurations are shown in Figure 16, where the first column contains the explicit boundary represented by dimensional parameters, and the second column contains their binary representations.

Structure	Compliance	Volume
(a) TO result (reference)	1.0448e-01	0.5
(b) Boundary with overhangs	1.0596e-01 (+1.42%)	0.5266 (+5.3%)
(c) Overhangs removed	8.5406e-02 (-19.4%)	0.6650 (+33%)
(d) Post-processing result	1.225e-01 (+17.25%)	0.5280 (+5.6%)

Table 1 Compliance and volume values of the four configurations discussed in this work.

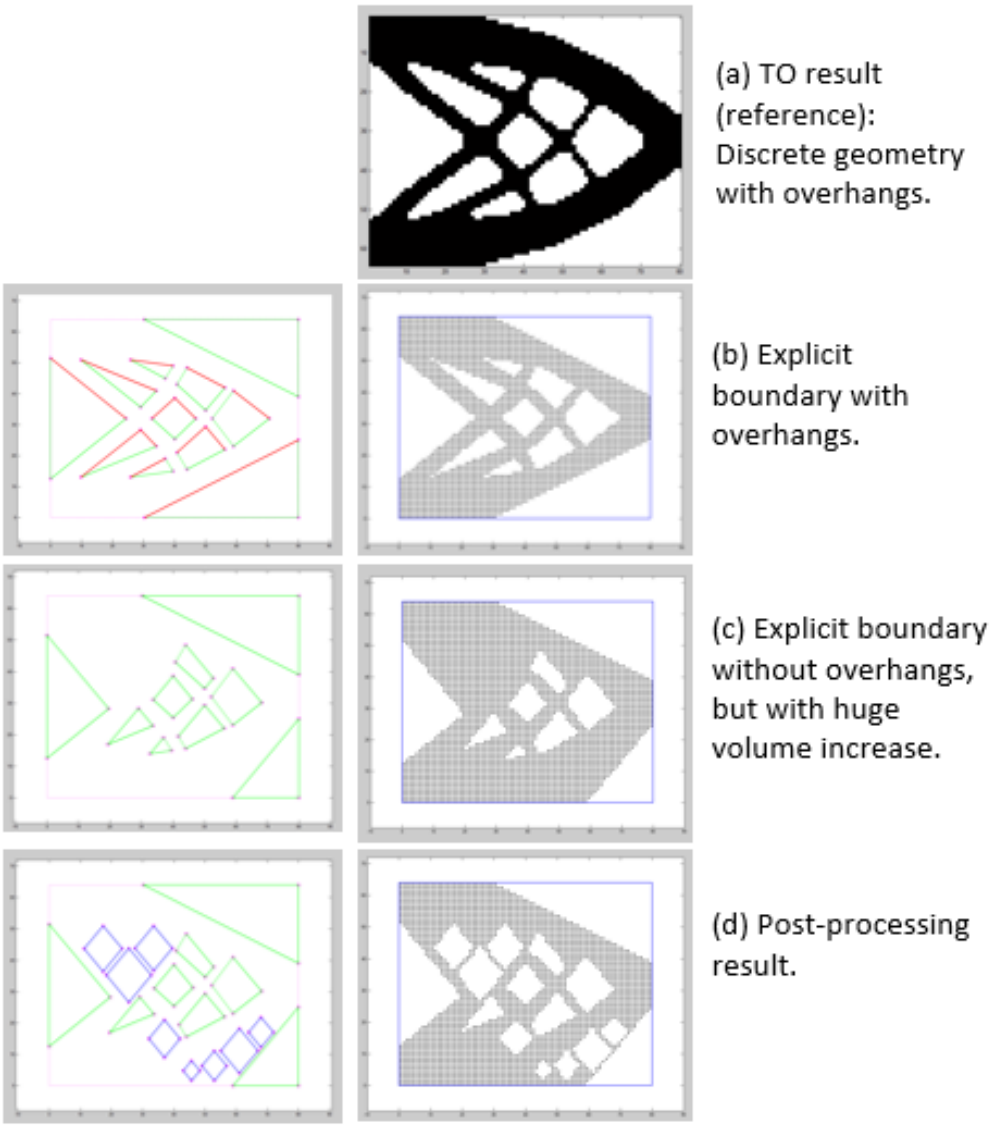

Fig. 16 Plots (explicit boundary + binary image representations) of the four configurations.

The topology optimization result, as shown in Figure 16(a), is served as reference to the comparison. After the implicit boundary was replaced with geometric entities, configuration (b) in Figure 16 was obtained, in which both compliance and volume saw a slight increase. Removing overhangs gave Figure 16(c), and led to a huge volume increase of the structure. Finally, as shown in configuration (d), the volume was brought back down by the volume control step to be only about 5% larger than the TO volume constraint.

This validated the third objective of this work, which is to control the part volume using design features to a prescribed volume objective.

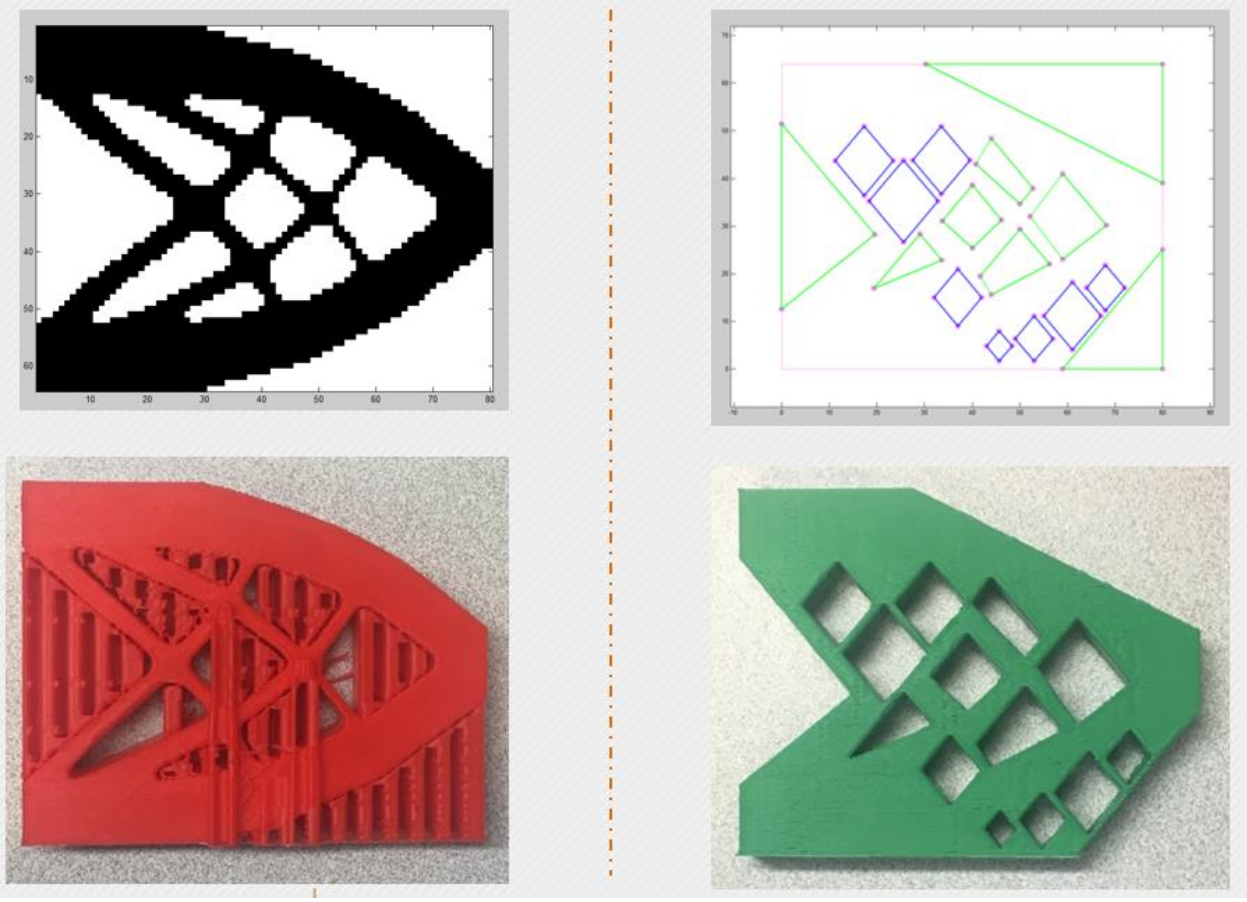

Fig. 17 Configurations before (left) and after (right) the post-processing procedure, and their fabricated parts using an FDM machine.

As the second part of the validation, two parts were built using that specific FDM machine, of which the overhang angle and minimum wall thickness were determined using test parts. In Figure 17, the red part is a slightly smoothed version of the topology optimization result of the 2D cantilever beam problem; the green one is the structure obtained from the post-processing procedure.

Comparing the two printed parts, we observe that the structure that directly came out of the optimization required support material in most void regions; while after post-processing, it became additively manufacturable.

This validated the second objective which is to remove overhangs from the topologically optimized configuration.

Conclusions / Future Work

In this work, a post-processing procedure was proposed for topologically optimized designs. The entire procedure is shown in Figure 16 in the order of (a) -> (b) -> (c) -> (d), in which each configuration presents the output of one step.

The output of the procedure (Figure 16d) satisfies the three objectives, so that:

1. Part boundary is represented explicitly with geometric parameters, so that the structure could easily be used for subsequent engineering.
2. Overhangs are removed.
3. Part volume is controlled within a small margin of error of the prescribed volume constraint.

Note that in Table 1, though the volume was well controlled, the post-processing result saw an increase in compliance by about 15%. This is because the post-processing does not consider mechanical properties of the structure, more specifically, the strain energy density. Thus, as future work, we would like to insert this complete post-processing procedure into the topology optimization scheme, so that the topologically optimized part boundary comes out directly to be a non-overhanging parametric one, while at the same time, the structure's stiffness reaches its maximum under the constraints.

To this end, a more robust technique of inserting diamond features is required at the last stage of post-processing, because the objective function value, in this case the compliance, is sensitive to the exact positions of the diamonds.

The diamond-shaped feature would be filleted to avoid stress concentrations.

References

[1] Bendsøe MP, Kikuchi N (1988) Generating optimal topologies in structural design using a homogenization method. Comput Methods Appl Mech Eng 71:197–224

[2] Bendsøe MP (1989) Optimal shape design as a material distribution problem. Struct Multidiscipl Optim 1:193–202

[3] O. Sigmund A 99 line topology optimization code written in Matlab Struct Multidiscip Optim, 21 (2) (2001), pp. 120–127

[4] Wang MY, Wang XM, Guo DM (2003) A level set method for structural topology optimization. Comput Methods Appl Mech Eng 192:227–246

[5] Osher SJ, Sethian JA (1988) Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations. J Comput Phys 79:12–49

[6] Angelo L D and Stefano P D (2014) Geometric segmentation of 3D scanned surfaces Comput. Des. 62 44–56

[7] Koguchi A, Kikuchi N (2006) A surface reconstruction algorithm for topology optimization. Engineering with Computers 22: 1–10

[8] Nana A, Cuillière JC, Francois V (2017) Automatic reconstruction of beam structures from 3D topology optimization results. Comput & Structures 189 pp. 62–82

[9] Chen JQ, Shapiro V, Suresh K, Tsukanov I (2007) Shape optimization with topological changes and parametric control. Int J Numer Methods Eng 71:313–346

[10] Otomori M, Yamada T, Izui K, Nishiwaki S (2015) Matlab code for a level set-based topology optimization method using a reaction diffusion equation. Struct Multidiscip Optim 51:1159–1172

[11] S. Deng, K. Suresh (2016) Multi-constrained 3D topology optimization via augmented topological level-set. Comput & Structures 170 pp. 1–12