

Manufacturability Oriented Model Correction and Build Direction Optimization for Additive Manufacturing

Erva Ulu*, Nurcan Gecer Ulu, Walter Hsiao, Saigopal Nelaturi
Palo Alto Research Center
Palo Alto, California 94304
Email: eulu@parc.com

We introduce a method to analyze and modify a shape to make it manufacturable for a given additive manufacturing (AM) process. Different AM technologies, process parameters or materials introduce geometric constraints on what is manufacturable or not. Given an input 3D model and minimum printable feature size dictated by the manufacturing process characteristics and parameters, our algorithm generates a corrected geometry that is printable with the intended AM process. A key issue in model correction for manufacturability is the identification of critical features that are affected by the printing process. To address this challenge, we propose a topology aware approach to construct the allowable space for a print head to traverse during the 3D printing process. Combined with our build orientation optimization algorithm, the amount of modifications performed on the shape is kept at minimum while providing an accurate approximation of the as-manufactured part. We demonstrate our method on a variety of 3D models and validate it by 3D printing the results.

1 Introduction

Recent advances in additive manufacturing technologies have triggered the development of powerful design methodologies allowing designers to create highly complex functional parts [1–5]. Many such methods often operate under the assumption that any designed shape can be fabricated using a 3D printer and the resulting part matches the designed shape perfectly or with negligible errors. In reality, however, the selected AM process, mechanical characteristics of the printer or the material being used may introduce significant limitations to printability of a particular design in the form of a minimum printable feature size. For instance, the minimum feature size might be relatively large for a consumer level fused deposition modeling (FDM) printer while an order of magnitude smaller features could be achieved with a high-end stereolithography (SLA) printer [6]. Similarly, in electron beam melting (EBM) process, higher beam power

leads to a larger melt pool, thereby resulting in a larger minimum printable feature size [7]. Therefore, attempting to manufacture a complex shape with very small features using a consumer level FDM printer or an EBM machine with high beam power settings could lead to poor quality parts or even complete failure of the print. In order to avoid such problems during the printing process, it is crucial to take the manufacturing constraints into account during the design stage and modify the shape accordingly.

We propose a new approach for analyzing and altering the designed geometry to make it manufacturable with the intended 3D printing approach and corresponding process parameters. Our approach takes as input (1) a 3D shape represented by its boundary surface mesh and (2) a minimum printable feature size dictated by the manufacturing process, and produces a *corrected* model that accurately approximates the as-manufactured part which can be 3D printed without any failures (Figure 1).

In analyzing and correcting a model for manufacturability, the main challenge lies in identifying the important features and determining the modifications to the design that affects the shape minimally. Our approach overcomes this challenge by constructing a *meso-skeleton*—the maximal area within each slice where a print head can be positioned during the printing process—that is topologically equivalent to the corresponding slice of the input shape. This approach allows us to thicken topologically important parts that are smaller than the minimum printable size by creating a minimal path for the print head to deposit material. We keep these modifications minimal by optimizing the build orientation of the part. Our approach operates on the slice level, simulating the layer-by-layer printing process, and thus, the resulting geometry accurately approximates the printed part revealing the downstream geometric changes in a design-to-manufacturing pipeline.

Our main contributions are:

1. a novel topology aware model correction method for manufacturability,
2. a method to construct a meso-skeleton that serves as a

*Address all correspondence to this author.



Fig. 1. We present a method for correcting 3D models to make them manufacturable with the intended additive manufacturing process. Our approach analyzes the input model (gray) layer by layer and provides modifications to the shape (in the form of addition (red) or removal (blue) of material) to avoid fabrication errors due to the minimum printable feature size. Required modifications are minimized by optimizing the build direction (right).

proxy to the maximal area that the print head can traverse,

3. a build orientation selection algorithm to minimize the amount of required shape modifications.

2 Related Work

Our review is comprised of studies highlighting *design for fabrication constraints*, *manufacturability analysis and model correction*, and *build direction selection* with an emphasis on approaches involving additive manufacturing.

2.1 Design for Fabrication Constraints

A large body of work has investigated automatic techniques offering design aids in creating shapes that satisfy a variety of fabrication related constraints [8,9]. Recent examples include partitioning and packing approaches to accommodate large prints [10–12], advanced slicing methods to improve the print quality and build time [13–16], automatic support structure design methods [17, 18] as well as shape modification approaches to minimize or avoid support structures [19–21]. Other shape modification approaches have been explored to meet geometric requirements such as minimum feature size or wall thickness [22–24]. Our approach falls under this general category in that we alter the geometry to make it manufacturable under the given minimum printable feature size limitations. Our aim, however is to provide an accurate approximation of the as-manufactured part by performing a slice based analysis to construct an allowable space for the print head to traverse during the 3D printing process.

2.2 Manufacturability Analysis and Model Correction

In the context of additive manufacturing, conventional manufacturability analysis approaches are often intended for detecting artifacts in the input surface mesh such as self-intersections, non-manifoldness and non-watertightness [28, 29]. However, even models free of such artifacts may

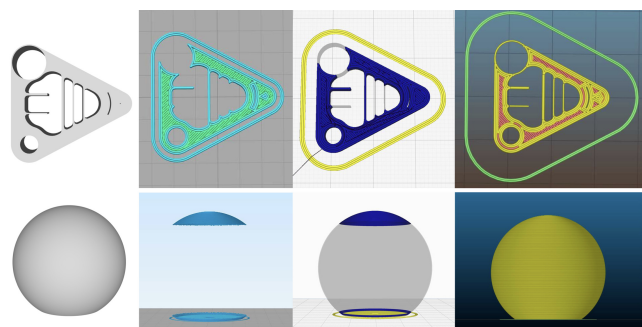


Fig. 2. Toolpaths generated by some of the commonly used slicers for two example models: a test part with thin features (top) and a thin spherical shell (bottom). Left-to-right: the input model, toolpaths generated by Simplify3D [25], Ultimaker Cura [26] and Slic3r (Prusa edition) [27].

not be 3D printable as the size of certain regions of the object might drop below the printing resolution [30]. Morphological operations and distance transforms [31], ray casting [32] and feature recognition [33] based approaches have been presented recently to identify such problematic regions. Although these approaches are demonstrated to provide valuable design feedback to user by determining compatibility of a 3D model with the printing hardware, very few solutions exist to automatically modify the shape to mitigate these manufacturability problems.

In [34], problematic regions are identified using 2D shape diameter function and then, a physics based mesh deformation technique followed by a post-processing step using morphological operations is used to make local corrections to improve printability of the design. In this approach, small holes and narrow intrusions are aimed to be eliminated as a part of the correction process. While the provided corrective actions improve the design significantly, it has been shown that the printability is not guaranteed. Nela-turi *et al.* [35] use medial axis transformation together with techniques from mathematical morphology to construct a printability map and recommend design modifications. How-

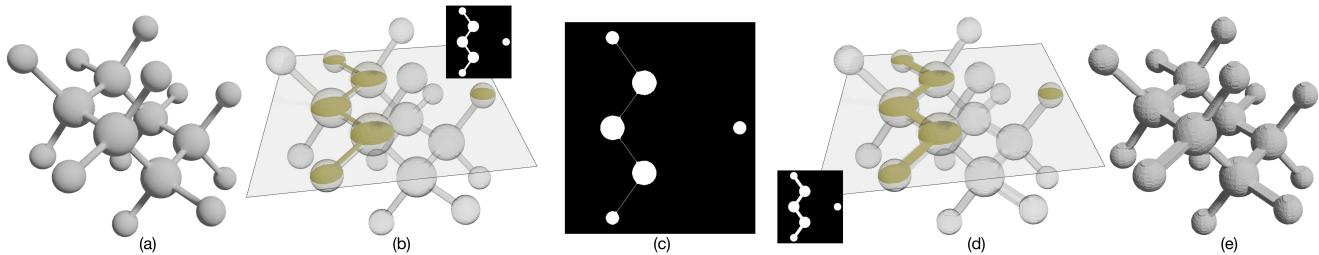


Fig. 3. Given an input 3D shape (a) and a minimum printable feature size (or a structuring element size), our algorithm voxelizes the shape to generate 2D slices corresponding to layers in the printing process (b). Each layer is then analyzed to construct a meso-skeleton (c). Dilation of each meso-skeleton with the structuring element results in the corrected slice (d). The stack of corrected slices is used to generate the boundary surface of the corrected 3D model (e). Note that our approach preserves thin skeletal paths (c) that would disappear through conventional morphological opening operations.

ever, as the medial axis computation is very sensitive to noise, many unwanted branches may be created, especially on high curvature regions such as corners. Therefore, a special care has to be taken to remove them. We are inspired by this approach in that we use a skeletal structure to preserve topologically and geometrically important features when correcting the shapes for manufacturability. In order to address the sensitivity problems however, we utilize thinning methodologies [36] in binary image processing to construct our meso-skeleton.

2.3 Build Direction Selection

Effects of build orientation on the build time and cost [37, 38], the amount of support material [39, 40], the mechanical properties [41, 42] and the surface quality [43] have been studied extensively and automatic build direction selection algorithms are proposed to minimize such directional biases. Other works focus on geometrical inaccuracies caused by the staircase artifacts inherent to layered manufacturing processes [44, 45]. Although driven by similar motivations, in this work, we optimize the build direction in order to minimize the amount of modifications required to make the shape manufacturable with the given manufacturing process. Moreover, our formulation is complementary to the aforementioned methods in that they may be used together to improve manufacturability of a design while also targeting other objectives.

3 Shape Modifications

In a traditional additive manufacturing pipeline, the 3D model is sliced into layers and then, each layer is individually processed to generate a set of machine instructions, usually in the form of a toolpath plan. The sequence of the movements in toolpath aims to track the outer boundary of the slice as well as to fill the inner regions and possible support structures. The conventional approach in toolpath planning is to generate machine instructions such that the deposited material is restricted to the interior of the slice. For a given slice S_i and a minimum printable feature F represented by a circle of diameter d , the maximal allowable region M_i that the print head can traverse can be calculated by the morpho-

logical erosion operation as

$$M_i = S_i \ominus F. \quad (1)$$

Therefore, the printed slice geometry P_i can be approximated by the morphological opening of S_i by F

$$\begin{aligned} P_i &= (S_i \ominus F) \oplus F \\ &= M_i \oplus F \end{aligned} \quad (2)$$

where \ominus and \oplus denote morphological erosion and dilation operations, respectively. However, this approach causes features that are smaller than F to disappear in the created tool-path and thus, in the fabricated part. Figure 2 illustrates example cases and corresponding toolpaths generated by various slicer software commonly used for FDM process. Note that, in some cases such as the thin spherical shell model, these problems manifest themselves as complete removal of layers and therefore, result in failure of the entire print process. A number of slicers attempt to address this issue by performing corrective actions in toolpath level. For instance, Slic3r adjusts the filament feed rate to an amount below what is optimal to be able to print walls thinner than the nozzle diameter (Figure 2, right column), or IceSL [46] provides an option to thicken the critical regions by over-depositing material. However, modifications that are made in the tool-path level usually target a particular AM process (*e.g.*, the above two examples target FDM process) and they may create ambiguities between what is designed and what is manufactured. Moreover, most commercial AM machines (such as Stratasys Objet or Dimension Elite) only support toolpath planning through their proprietary software that may not perform any corrective actions. Although our approach is similar to above mentioned tools in that it operates on slice level, we aim to correct the input model during the design stage, such that the design itself approximates the as-manufactured part as closely as possible and can be printed using the intended AM process.

Figure 3 illustrates the overview of our model correction approach. We start by voxelizing the input 3D shape \mathcal{B} such

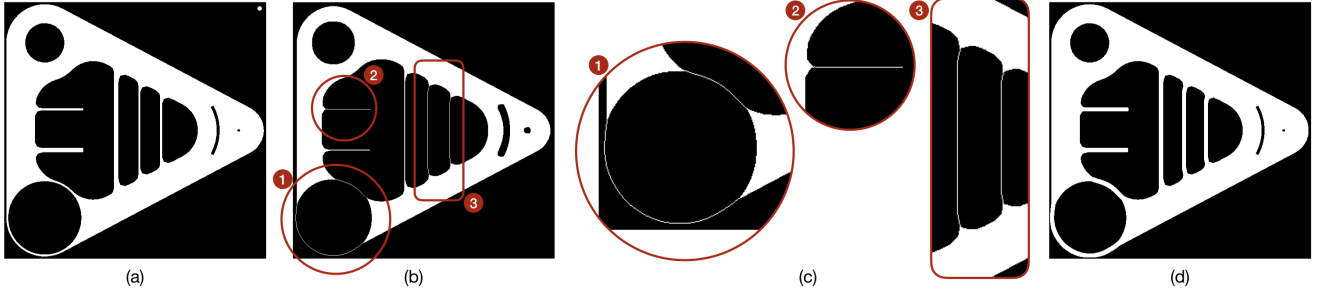


Fig. 4. An example slice of the test part (a) and corresponding meso-skeleton created by our algorithm (b). Notice that skeletal paths are created for topologically important features such as (1) and (3) as well as geometrically distinct regions such as the protrusion (2) in (c). In the resulting corrected model (d), all of these problematic thin features are automatically thickened. The minimum printable feature (or the structuring element) is shown in upper right corner in (a).

ALGORITHM 1: Our shape modification algorithm

Input : \mathcal{B}, F
Output: $\hat{\mathcal{B}}$
 $\mathcal{V} \leftarrow \text{Voxelize } \mathcal{B};$
 $\hat{\mathcal{V}} \leftarrow \emptyset;$
foreach *Slice* $S_i \in \mathcal{V}$ **do**
 $\hat{M}_i \leftarrow \text{Compute meso-skeleton};$
 $\hat{S}_i \leftarrow \hat{M}_i \oplus F;$
 Append \hat{S}_i to $\hat{\mathcal{V}};$
end
 $\hat{\mathcal{B}} \leftarrow \text{Compute boundary surface of } \hat{\mathcal{V}};$

that each slice of voxels in z-direction represents a layer in the 3D printing process. Then, for each slice S_i , we construct a *meso-skeleton*, \hat{M}_i . Similar to M_i in (1), \hat{M}_i defines the allowable region that the print head can traverse while printing. However, it contains additional skeletal paths for the features that are smaller than F in size. Therefore, sweeping the structuring element F over the meso-skeleton creates the corrected slice where the smallest feature is guaranteed to be equal or larger than F in size. Finally, stacking all the corrected slices together results in the corrected 3D model. Algorithm 1 summarizes our method. Here, \mathcal{V} is the voxelized model, $\hat{\mathcal{V}}$ is the corrected voxelization that is composed of corrected slices \hat{S}_i and $\hat{\mathcal{B}}$ represents the corrected surface mesh that is ready to be printed. Note that the corrected slices \hat{S}_i can also be used directly to print the modified geometry using AM processes that support image based representation of layers, such as digital light processing (DLP). For other AM methods, perimeter polygons may be extracted from \hat{S}_i in order to generate the toolpath required by the printer. For the sake of generality, we construct the corrected model $\hat{\mathcal{B}}$ in the form of a boundary surface mesh that is widely supported by various AM processes. In addition, $\hat{\mathcal{B}}$ provides an accurate approximation of the as-manufactured part to the user.

3.1 Meso-Skeleton Computation

We compute the meso-skeleton of a slice by performing a topology preserving thinning operation. Given an input

ALGORITHM 2: Our meso-skeleton computation algorithm

Input : S_i, F
Output: \hat{M}_i
 $M_i \leftarrow S_i \ominus F;$
 ${}^p\hat{M}_i \leftarrow \emptyset;$
 $\hat{M}_i \leftarrow S_i;$
 $j \leftarrow 0;$
while $\hat{M}_i \neq {}^p\hat{M}_i$ **do**
 ${}^p\hat{M}_i \leftarrow \hat{M}_i;$
 Apply thinning to \hat{M}_i ;
 if $j < d/2$ **then**
 Remove spur pixels in \hat{M}_i ;
 end
 $\hat{M}_i \leftarrow \hat{M}_i \vee M_i;$
 $j \leftarrow j + 1;$
end

slice S_i represented by a binary image where white pixels represent internal points and black pixels are background, we remove contour pixels that are not essential to its topology. Iterative removal of such pixels result in a *thinned image* that constitutes our meso-skeleton.

Algorithm 2 summarizes our approach to compute the meso-skeleton in detail. We start by computing the conventional erosion M_i of the input slice. We use the erosion as an upper bound in the thinning process, *i.e.*, the meso-skeleton can not be shrunk beyond M_i . We ensure this by performing thinning operations on the input image only until the union of the thinned image \hat{M}_i with M_i does not change. We employ the two-subiteration thinning algorithm in [47]. Here, the deletion or retention of a pixel p is determined by the configuration of the pixels in its local 8-connected neighborhood, $N(p)$. Under this scheme, a contour pixel p is deleted if

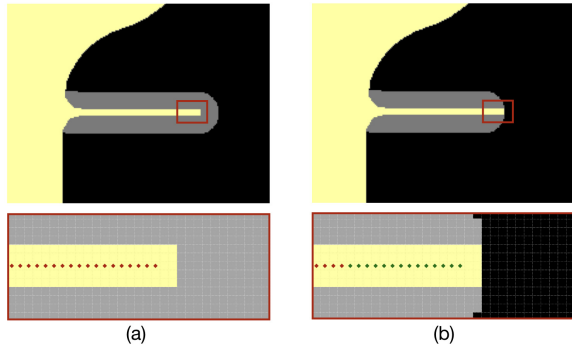


Fig. 5. For the same slice (yellow), the protrusion is elongated in the corrected model (gray) when the skeleton is obtained using the thinning process alone (a). Our spur pixel removal allows us to achieve the intended length by deleting the end-point pixels (marked green) (b). Top row is the original slice (yellow) overlaid with the corrected slice (gray) and the bottom row shows the close-up view of the tip of the protrusion demonstrating the pixel grid. Red marked pixels represent the meso-skeleton pixels.

$$\begin{aligned}
 (a) \quad & X_H(p) = 1, \\
 (b) \quad & 2 \leq \min(n_1(p), n_2(p)) \leq 3, \\
 & n_1(p) = \sum_{k=1}^4 x_{2k-1} \vee x_{2k}, \\
 & n_2(p) = \sum_{k=1}^4 x_{2k} \vee x_{2k+1}, \\
 (c.1) \quad & (x_2 \vee x_3 \vee \neg x_8) \wedge x_1 = 0, \\
 (c.2) \quad & (x_6 \vee x_7 \vee \neg x_4) \wedge x_5 = 0,
 \end{aligned} \tag{3}$$

where $X_H(p)$ is the crossing number as defined in [36,48] and x_k are the values of pixels in $N(p)$ with $k = 1, \dots, 8$, starting with the east neighbor and numbered in counter-clockwise order. In the first subiteration, pixels satisfying (a), (b), (c.1) and in the second subiteration, pixels satisfying (a), (b), (c.2) are removed.

Condition (a) and alternating conditions (c.1) and (c.2) ensure that the local connectivity is maintained. Here, the constraint on the crossing number serves as a necessary condition and prevents deletion of pixels in the middle of the medial curves. Condition (b) provides as an endpoint check. While redundant pixels in the middle of the curves are allowed to be deleted, endpoints are detected and preserved through this condition. With this approach, the object shrinks to an 8-connected skeleton when applied repeatedly, *i.e.*, an object without holes reduces to a minimally connected stroke whereas an object with holes shrinks to a connected ring halfway between each hole and the outer contour. As the Euler number is preserved, a meso-skeleton \hat{M}_i that is homotopic to S_i is generated. Moreover, this approach allows us to preserve a skeletal structure for geometrically distinct features such as protrusions, in addition to the topologically

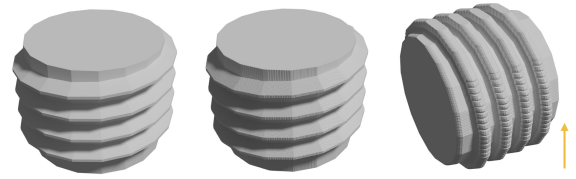


Fig. 6. A threaded cylinder (left) can be printed without any corrections in the vertical orientation (middle) whereas a significant amount of corrections are required in the horizontal orientation (right). Build direction is indicated with a yellow arrow.

essential parts (Figure 4). For detailed overview of various thinning methodologies and their comparisons to the approach we employed here, the reader is referred to [36].

For long and thin protrusions, the thinning operation in (3) may converge to the skeletal structure quickly and terminate further removal of pixels. This tends to create undesirable overdepositions later in constructing the corrected model. We demonstrate an example problematic case in Figure 5. As the red marked pixels are assigned to be skeleton pixels in the thinning process, they do not get removed in later iterations. However, following dilation operation of this meso-skeleton with F would extend the protrusion beyond what was originally designed. We address this problem by detecting the spur pixels that are not removed during the thinning process and removing them. To ensure that the protrusion has the intended length in the corrected model, we perform this operation only for the first $d/2$ iterations. A corrected slice obtained with and without spur pixel removal is illustrated in Figure 5.

4 Build Direction Optimization

As our shape modification method mimics the additive manufacturing process and operates on the slice level, the quality of a printed part and thus, our correction result depends on the selected build direction. The amount of change in the shape (*i.e.*, amount of added or removed material with the correction process) may be significantly different for different build directions. For instance, a threaded cylinder can be 3D printed in a vertical orientation perfectly, while certain modifications (*e.g.*, thickening and rounding sharp thread and cylinder edges) are required for the horizontal orientation (Figure 6) to be printable. Therefore, it is essential to select the right build direction to minimize the difference between the designed shape and the manufactured object.

Formal definition of the build direction optimization problem is as follows:

$$\begin{aligned}
 & \underset{\theta}{\text{minimize}} \quad C(\theta) = \sum_i |\hat{S}_i(\theta) - S_i(\theta)| \\
 & \text{subject to} \quad \theta_1 \in [-\pi/2, \pi/2] \text{ and } \theta_2 \in [0, \pi], \\
 & \quad \quad \quad \theta = [\theta_1, \theta_2]^T.
 \end{aligned} \tag{4}$$



Fig. 7. Model correction results. Left-to-right: input uncorrected models with their corresponding failed prints, corrected models and their successful prints. Added and removed material are highlighted in red and blue, respectively.

Here, θ is the vector of design variables where θ_1 and θ_2 denote the extrinsic Euler angles representing a sequential rotation of the build direction vector about the x and z axes, respectively. Note that the build direction is initialized at +z direction for $\theta = [0, 0]^T$. As the slices are identical for any two opposite build directions, we only explore the half-space by setting the bounds for θ_1 and θ_2 as given in (4).

The objective function in (4) measures the absolute difference between the input shape and the corrected shape. Therefore, both the amount of added material through the thickening operations and the amount removed material due to the rounded corners/edges are taken into account in determining the optimum build direction. Note that it is also possible to enforce the optimizer to favor one over the other (*i.e.*, material addition might be tolerated in some applications while removal is not) by constructing the objective function as a weighted sum of these two components. Our objective function in (4) corresponds to equally weighted sum in such formulation.

As the algebraic definition of the gradients are not available, we use a derivative-free simulated annealing approach to solve our optimization problem. We implement the simulated annealing algorithm as described in [49]. The algorithm mimics the thermal annealing process of heating and slowly lowering the temperature of a part to minimize system energy. At each optimization step, we generate new candidate states by sampling a new design vector around the current state. The new states are drawn from a Gaussian distribution. Variance of the distribution is proportional to the temperature of the system. Since the optimization starts with a high temperature value that decreases through the iterations, the extent of the search (*i.e.*, the distance of the new

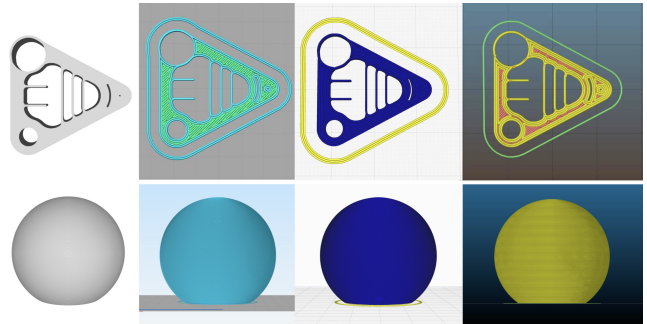


Fig. 8. For the corrected models, slicer software can generate toolpaths successfully without creating any ambiguities. Left-to-right: the corrected model, toolpaths generated by Simplify3D [25], Ultimaker Cura [26] and Slic3r (Prusa edition) [27].

point from the current point) is higher at the beginning of the optimization allowing a broad search and gets smaller as the optimization converges. Any new state with a lower objective value is accepted. However, the algorithm allows accepting a solution with a higher objective value through a certain probability and alleviates the issue of getting stuck at a local minima. For cooling schedules, we are using exponential cooling in all of our examples. We also experimented with linear and logarithmic cooling schedules and observed similar convergence properties.

5 Results and Discussion

Model Correction Figure 7 illustrates the results of our model correction on various models. Here, we assume the build direction to be $\theta = [0, 0]$. Our meso-skeleton approach

Table 1. Performance of our model correction algorithm on a variety of models.

Model	Slices	Image Size	Time [s]
Test part	65	[583 × 597]	3.44
Molecule	383	[513 × 462]	11.22
Threaded cylinder	421	[623 × 632]	44.78
Spherical shell	712	[795 × 795]	67.96
Playground	555	[1417 × 1479]	187.57
Dragon	687	[284 × 253]	4.86
Helical tower	1472	[376 × 382]	27.75



Fig. 9. Distribution of the maximum allowable structuring element size suggested by our algorithm. Left-to-right: corrected models obtained using varying structuring element size range of $[0.1, 1.2]mm$, $[0.6, 1.2]mm$ and the corresponding printed model.

detects the problematic regions and modifies each layer automatically to make them manufacturable. Printed results verify our method that manufacturing failures are resolved in the corrected models. Note that in some extreme cases such as the uniform thickness spherical shell model (middle row), our correction makes the completely unprintable model printable using the same manufacturing process and same set of process parameters. Similarly, disappearing regions that are geometrically or topologically important in the design is recovered for the test part (top row) and the playground (bottom row). Table 1 summarizes our model correction results with various metrics relevant to these models.

In Figure 8, we show the toolpaths generated by different slicer software for our corrected test part and spherical shell models. Unlike the toolpaths generated for the original models (Figure 2), there are no ambiguities in the resulting shape, *i.e.*, all the slicers were able to generate similar results with no indication of any fabrication failure.

Structuring Element Size As the meso-skeleton is generated through iterative removal of contour pixels, we can compute the exact state where the skeleton is reduced to a

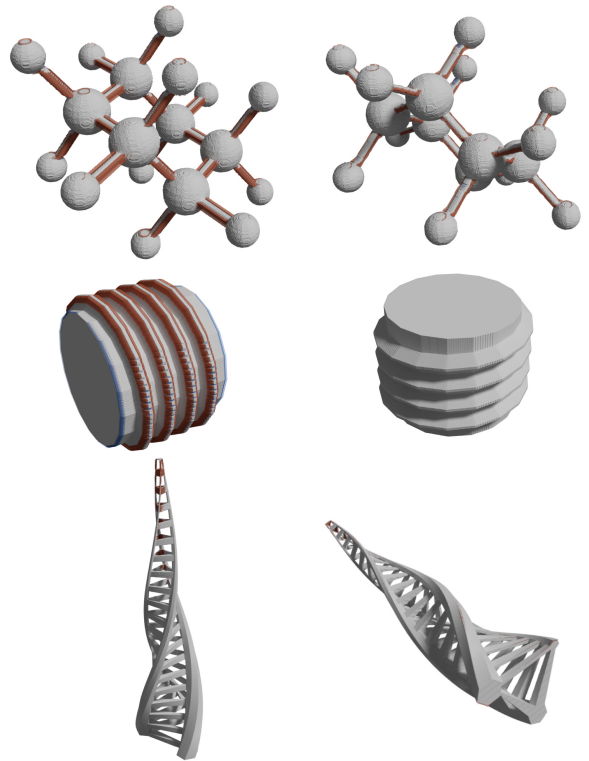


Fig. 10. Build direction optimization results. The amount of corrections required to make the model manufacturable is significantly smaller in the optimized orientation (right) compared to the arbitrary initial orientation (left). Added and removed material are highlighted in red and blue, respectively.

single pixel wide line for the first time. At this state, the number of removed contour layers corresponds to the maximum allowable structuring element size to print the current slice without making any topologically or geometrically important changes. This allows us to suggest a per-layer structuring element size that can be used to fabricate the input geometry without requiring any correction. Figure 9 illustrates the distribution of the required structuring element size on an example model. It can be observed that for layers involving thin features (such as the tip of the wings, the claws and the wing membranes), our algorithm suggests smaller structuring elements as expected. In order to validate this result, we have modified an open-source slicer (gsSlicer [50]) to adjust the extrusion width for each layer individually in the FDM toolpath generation. Due to the physical constraints of the printer, we limited the change in the extrusion width to $2\times$ only by setting the allowable range to $0.6 - 1.2mm$. In that case, it can be observed in Figure 9 (middle) that layers containing critical features are assigned the lower bound value. Such an adjustment in the extrusion width results in $\sim 8\%$ decrease in the build time as the layers without any critical features can be printed faster using a larger extrusion width. The improvement in build time can be more significant for complex models containing large bulky regions with more localized small and thin features.

Table 2. Performance of our build direction optimization on a variety of models.

Model	Objective Value		θ	Time
	Initial	Optimized		
Molecule	180661	118927	[-0.75, 0.03]	62m
Threaded cylinder	2040590	26687	[1.52, 0.00]	3h 56m
Dragon	131748	63694	[-1.98, 1.57]	35m
Helical tower	134276	112601	[-1.51, 1.09]	2h 31m

Build Direction Figure 10 and Figure 1 illustrate our build direction optimization results. We achieved $\sim 15\%$ to $\sim 50\%$ reduction in the objective value for the molecule, the dragon and helical tower models. In order to validate our optimization approach, we also experimented with the threaded cylinder model. Our optimizer was able to converge to the obvious solution of the vertical print configuration at which the model can be printed almost perfectly. It can be observed that major problems in the horizontal orientation including the rounding of the cylinder edges and the thickening of the sharp threads are avoided in the vertical orientation, resulting in $\sim 98\%$ reduction in the objective value. Table 2 summarizes our build direction optimization results together with other performance metrics.

5.1 Validation and Performance

Fabrication We 3D printed the original designs as well as the corrected models using a consumer level FDM printer with PLA material. We use Simplify3D [25] to generate the toolpaths for our prints. In order to compare the results, we have kept all the printer and slicer parameters the same for both the original designs and the corrected models. Although we have only experimented with FDM printing technology, our approach is applicable to a wide variety of additive manufacturing methods where the printing process is driven by the planar translation of a material deposition or fusion mechanism, including but not limited to stereolithography, sintering and binding based methods.

Performance We tested our method on a 3.7GHz 8 Core Intel Xeon computer with 32GB of memory. As each slice can be processed independently, we use parallel processing in our shape modification algorithm. Our parallel implementation is only limited the outer-most loop where the individual slices are processed. However, the thinning algorithm can also be parallelized for further improvement in the performance. We use OpenVDB [51] for sparse voxel operations.

Table 1 shows the performance of our correction algorithm for a predetermined build direction ($\theta = [0, 0]$). As the thinning operation is performed repeatedly for each slice, it constitutes the majority of the computational cost in our approach. Therefore, the performance is heavily affected by both the number of slices as well as the size of each slice (*i.e.*, image size).

Table 2 reports the performance of our build direction optimization algorithm. In all of our examples, the optimization converges under 300 iterations. As the number of slices and the image size change for each new candidate build direction, average per iteration processing time may change significantly in comparison to the single iteration time reported in Table 1. We have observed 10% to 40% increase in our example models. As only the per-slice operations are parallelized in our implementation, we observe longer processing times for candidate orientations that have smaller number of slices with larger image sizes.

5.2 Limitations and Future Work

In our model correction approach, we depend on the meso-skeleton for topological correctness of the resulting model. We found this approach to work well in preserving the topology for models where the topologically important thin features are not very close to each other. However, for shapes where the minimum distance between two thin features are smaller than the structuring element diameter, the topology may change when the meso-skeleton is dilated with the structuring element. For example, when there is a small channel or tunnel close to a thin feature that is required to be corrected, it may be filled and disappear due to the thickening operation on the thin feature next to it. Although detection of these local topological changes is possible [52], model correction under such problematic cases remains an open problem.

Our approach relies on voxel/pixel based representation of the model and individual slices in various stages of the correction and the optimization processes. Although it is a widely accepted practical approach in discretizing the 3D shape, it introduces larger aliasing error compared to the other boundary conformal representations including ray-rep and its variations [53]. Complementing our approach with such representations could improve the quality of the resulting models and corresponding printed parts.

As the meso-skeleton created for thin features is approximately in the middle with respect to the contour, the resulting correction thickens the object equally in both directions. However, in certain applications, this may not be desirable due to the geometrical constraints. A natural extension of our approach would be to allow unidirectional corrections such that the geometrical tolerances or constraints are satisfied.

In our build direction optimization, we only focus on

minimizing the change in the shape. However, build direction affects many other properties significantly, including support structures, strength and surface roughness. Such properties may also influence the printability of a model. In future, our optimization could be extended to take these effects into account and select a build direction minimizing the negative aspects concurrently to improve the printability.

6 Conclusion

We present a model correction and build orientation optimization method for manufacturability. We propose a thinning based approach to compute the allowable space a print head can traverse in each slice during the printing process. With this approach, we show that both topologically and geometrically important features that are smaller than the minimum printable feature size are thickened automatically to allow a minimal path for the print head to deposit material. Combined with our build direction optimization, we demonstrate that the printability of a model can be improved with a minimal amount of shape modifications. We evaluate the performance of our method on variety of 3D models. Our results show that corrected model accurately approximates the printed part reducing ambiguities in the downstream design-to-manufacturing pipeline.

Acknowledgements

The authors would like to thank Chaman Singh Verma for his help with OpenVDB. We are grateful to the designers whose 3D models we used: Louise Driggers (Thingiverse) for dragon, marcaxi (Thingiverse) for threaded cylinder and Daniel Lu (Grabcad) for molecule. This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA), United States. The views, opinions and/or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or U.S. Government.

References

- [1] Schumacher, C., Bickel, B., Rys, J., Marschner, S., Daraio, C., and Gross, M., 2015. “Microstructures to control elasticity in 3d printing”. *ACM Trans. Graph.*, **34**(4), July, pp. 136:1–136:13.
- [2] Martínez, J., Dumas, J., and Lefebvre, S., 2016. “Procedural voronoi foams for additive manufacturing”. *ACM Trans. Graph.*, **35**(4), July, pp. 44:1–44:12.
- [3] Liu, H., Hu, Y., Zhu, B., Matusik, W., and Sifakis, E., 2018. “Narrow-band topology optimization on a sparsely populated grid”. *ACM Trans. Graph.*, **37**(6), Dec., pp. 251:1–251:14.
- [4] Arora, R., Jacobson, A., Langlois, T. R., Huang, Y., Mueller, C. T., Matusik, W., Shamir, A., Singh, K., and Levin, D. I. W., 2019. “Volumetric michell trusses for parametric design & fabrication”. In *Proceedings of the 3rd ACM Symposium on Computation Fabrication (SCF)*.
- [5] Ulu, E., McCann, J., and Kara, L. B., 2019. “Structural design using laplacian shells”. *Computer Graphics Forum*, **38**(5), pp. 85–98.
- [6] Bhushan, B., and Caspers, M., 2017. “An overview of additive manufacturing (3d printing) for microfabrication”. *Microsystem Technologies*, **23**(4), Apr, pp. 1117–1124.
- [7] Ulu, E., Huang, R., Kara, L. B., and Whitefoot, K. S., 2019. “Concurrent structure and process optimization for minimum cost metal additive manufacturing”. *Journal of Mechanical Design*, **141**(6), pp. 061701–061701–10.
- [8] Livesu, M., Ellero, S., Martínez, J., Lefebvre, S., and Attene, M., 2017. “From 3d models to 3d prints: an overview of the processing pipeline”. *Computer Graphics Forum*, **36**(2), pp. 537–564.
- [9] Bermanno, A. H., Funkhouser, T., and Rusinkiewicz, S., 2017. “State of the art in methods and representations for fabrication-aware design”. *Computer Graphics Forum*, **36**(2), pp. 509–535.
- [10] Luo, L., Baran, I., Rusinkiewicz, S., and Matusik, W., 2012. “Chopper: Partitioning models into 3d-printable parts”. *ACM Trans. Graph.*, **31**(6), Nov., pp. 129:1–129:9.
- [11] Vanek, J., Galicia, J. A. G., Benes, B., Měch, R., Carr, N., Stava, O., and Miller, G. S., 2014. “Packmerger: A 3d print volume optimizer”. *Computer Graphics Forum*, **33**(6), pp. 322–332.
- [12] Chen, X., Zhang, H., Lin, J., Hu, R., Lu, L., Huang, Q., Benes, B., Cohen-Or, D., and Chen, B., 2015. “Dapper: Decompose-and-pack for 3d printing”. *ACM Trans. Graph.*, **34**(6), Oct., pp. 213:1–213:12.
- [13] Huang, P., Wang, C. C. L., and Chen, Y., 2013. “Intersection-free and topologically faithful slicing of implicit solid”. *Journal of Computing and Information Science in Engineering*, **13**(2), p. 021009.
- [14] Wang, W., Chao, H., Tong, J., Yang, Z., Tong, X., Li, H., Liu, X., and Liu, L., 2015. “Saliency-preserving slicing optimization for effective 3d printing”. *Computer Graphics Forum*, **34**(6), pp. 148–160.
- [15] Alexa, M., Hildebrand, K., and Lefebvre, S., 2017. “Optimal discrete slicing”. *ACM Trans. Graph.*, **36**(4), Jan.
- [16] Mao, H., Kwok, T.-H., Chen, Y., and Wang, C. C., 2019. “Adaptive slicing based on efficient profile analysis”. *Computer-Aided Design*, **107**, pp. 89 – 101.
- [17] Vanek, J., Galicia, J. A. G., and Benes, B., 2014. “Clever support: Efficient support structure generation for digital fabrication”. *Computer Graphics Forum*, **33**(5), pp. 117–125.
- [18] Dumas, J., Hergel, J., and Lefebvre, S., 2014. “Bridging the gap: Automated steady scaffoldings for 3d printing”. *ACM Trans. Graph.*, **33**(4), July, pp. 98:1–98:10.
- [19] Hu, K., Jin, S., and Wang, C. C., 2015. “Support slimming for single material based additive manufacturing”. *Computer-Aided Design*, **65**, pp. 1 – 10.
- [20] Xie, Y., and Chen, X., 2017. “Support-free interior

- carving for 3d printing”. *Visual Informatics*, **1**(1), pp. 9 – 15.
- [21] Wei, X., Qiu, S., Zhu, L., Feng, R., Tian, Y., Xi, J., and Zheng, Y., 2018. “Toward support-free 3d printing: A skeletal approach for partitioning models”. *IEEE Transactions on Visualization and Computer Graphics*, **24**(10), Oct, pp. 2799–2812.
- [22] Wang, C. C., and Chen, Y., 2013. “Thickening freeform surfaces for solid fabrication”. *Rapid Prototyping Journal*, **19**(6), pp. 395–406.
- [23] Cabiddu, D., and Attene, M., 2017. “ε-maps: Characterizing, detecting and thickening thin features in geometric models”. *Computers & Graphics*, **66**, pp. 143 – 153. Shape Modeling International 2017.
- [24] Attene, M., 2018. “As-exact-as-possible repair of unprintable stl files”. *Rapid Prototyping Journal*, **24**(5), pp. 855–864.
- [25] Simplify3D, Inc., 2019. Simplify3d. <https://www.simplify3d.com/>. Accessed: 2019-04-05.
- [26] Ultimaker B.V., 2019. Ultimaker cura. <https://ultimaker.com/en/products/ultimaker-cura-software>. Accessed: 2019-04-05.
- [27] Prusa Research s.r.o., 2019. Slic3r (prusa edition). <https://www.prusa3d.com/slic3r-prusa-edition/>. Accessed: 2019-04-05.
- [28] Ju, T., 2004. “Robust repair of polygonal models”. *ACM Trans. Graph.*, **23**(3), Aug., pp. 888–895.
- [29] Attene, M., Campen, M., and Kobbelt, L., 2013. “Polygon mesh repairing: An application perspective”. *ACM Comput. Surv.*, **45**(2), Mar., pp. 15:1–15:33.
- [30] Moylan, S., Slotwinski, J., Cooke, A., Jurrens, K., and Donmez, M. A., 2012. “Proposal for a standardized test artifact for additive manufacturing machines and processes”. In Proceedings of the 2012 annual international solid freeform fabrication symposium, Austin, TX, pp. 6–8.
- [31] Telea, A., and Jalba, A., 2011. “Voxel-based assessment of printability of 3d shapes”. In *Mathematical Morphology and Its Applications to Image and Signal Processing*, P. Soille, M. Pesaresi, and G. K. Ouzounis, eds., Springer Berlin Heidelberg, pp. 393–404.
- [32] Tedia, S., and Williams, C. B., 2016. “Manufacturability analysis tool for additive manufacturing using voxel-based geometric modeling”. In *Solid Freeform Fabrication 2016: Proceedings of the 27th Annual International Solid Freeform Fabrication Symposium—An Additive Manufacturing Conference*, pp. 3–22.
- [33] Shi, Y., Zhang, Y., Baek, S., Backer, W. D., and Harik, R., 2018. “Manufacturability analysis for additive manufacturing using a novel feature recognition technique”. *Computer-Aided Design and Applications*, **15**(6), pp. 941–952.
- [34] Jaiswal, P., and Rai, R., 2019. “A geometric reasoning approach for additive manufacturing print quality assessment and automated model correction”. *Computer-Aided Design*, **109**, pp. 1 – 11.
- [35] Nelaturi, S., Kim, W., and Kurtoglu, T., 2015. “Manufacturability feedback and model correction for additive manufacturing”. *Journal of Manufacturing Science and Engineering*, **137**(2), p. 021015.
- [36] Lam, L., Lee, S.-W., and Suen, C. Y., 1992. “Thinning methodologies-a comprehensive survey”. *IEEE Trans. Pattern Anal. Mach. Intell.*, **14**(9), Sept., pp. 869–885.
- [37] Alexander, P., Allen, S., and Dutta, D., 1998. “Part orientation and build cost determination in layered manufacturing”. *Computer-Aided Design*, **30**(5), pp. 343 – 356.
- [38] Ahn, D., Kim, H., and Lee, S., 2007. “Fabrication direction optimization to minimize post-machining in layered manufacturing”. *International Journal of Machine Tools and Manufacture*, **47**(3), pp. 593 – 606.
- [39] Ezair, B., Massarwi, F., and Elber, G., 2015. “Orientation analysis of 3d objects toward minimal support volume in 3d-printing”. *Computers & Graphics*, **51**, pp. 117 – 124. International Conference Shape Modeling International.
- [40] Morgan, H. D., Cherry, J. A., Jonnalagadda, S., Ewing, D., and Sienz, J., 2016. “Part orientation optimization for the additive layer manufacture of metal components”. *The International Journal of Advanced Manufacturing Technology*, **86**(5), Sep, pp. 1679–1687.
- [41] Umetani, N., and Schmidt, R., 2013. “Cross-sectional structural analysis for 3d printing optimization”. In SIGGRAPH Asia 2013 Technical Briefs, SA '13, ACM, pp. 5:1–5:4.
- [42] Ulu, E., Korkmaz, E., Yay, K., Ozdoganlar, O. B., and Kara, L. B., 2015. “Enhancing the structural performance of additively manufactured objects through build orientation optimization”. *Journal of Mechanical Design*, **137**(11), p. 111410.
- [43] Zhang, X., Le, X., Panotopoulou, A., Whiting, E., and Wang, C. C. L., 2015. “Perceptual models of preference in 3d printing direction”. *ACM Trans. Graph.*, **34**(6), Oct., pp. 215:1–215:12.
- [44] Hildebrand, K., Bickel, B., and Alexa, M., 2013. “Orthogonal slicing for additive manufacturing”. *Computers & Graphics*, **37**(6), pp. 669 – 675. Shape Modeling International (SMI) Conference 2013.
- [45] Wang, W. M., Zanni, C., and Kobbelt, L., 2016. “Improved surface quality in 3d printing by optimizing the printing direction”. In Proceedings of the 37th Annual Conference of the European Association for Computer Graphics, EG '16, Eurographics Association, pp. 59–70.
- [46] Lefebvre, S., 2013. “Icesl: A gpu accelerated modeler and slicer”. In Proceedings of AEFA13, 18th European forum on additive manufacturing.
- [47] Guo, Z., and Hall, R. W., 1989. “Parallel thinning with two-subiteration algorithms”. *Commun. ACM*, **32**(3), Mar., pp. 359–373.
- [48] Hilitch, C. J., 1969. “Linear skeletons from square cupboards”. In *Machine Intelligence 4*, B. Meltzer and D. Michie, eds. Edinburgh University Press, p. 403.
- [49] Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P., 1983. “Optimization by simulated annealing”. *science*, **220**(4598), pp. 671–680.

- [50] Gradientspace, 2019. gsslicer. <http://www.gradientspace.com/opensource>. Accessed: 2019-04-18.
- [51] Museth, K., Lait, J., Johanson, J., Budsberg, J., Henderson, R., Alden, M., Cucka, P., Hill, D., and Pearce, A., 2013. “Opencvdb: An open-source data structure and toolkit for high-resolution volumes”. In *ACM SIGGRAPH 2013 Courses, SIGGRAPH '13*, ACM, pp. 19:1–19:1.
- [52] Behandish, M., Mirzendehtel, A. M., and Nelaturi, S., 2019. “Detection and correction of topological errors in additive manufacturing”. *Computer-Aided Design*. Special Issue on the 23rd ACM Symposium on Solid and Physical Modeling (SPM'2019).
- [53] Chen, Y., and Wang, C. C. L., 2013. “Regulating complex geometries using layered depth-normal images for rapid prototyping and manufacturing”. *Rapid Prototyping Journal*, **19**(4), pp. 253–268.