

Improving printing orientation for Fused Deposition Modeling printers by analyzing connected components

Jorge A. García Galicia*, Bedrich Benes

High Performance Computer Graphics Laboratory, Purdue University, 401 N. Grant St., West Lafayette, IN 47907, USA

ARTICLE INFO

Keywords:

Additive manufacturing
Optimization
Printing path planning

ABSTRACT

The spatial orientation of an object on a 3D printing plate is a significant contributor to its printing time. Thus, the speed of the 3D printing processes can generally be increased by using time-efficient object orientations. This paper presents a novel method for speeding-up printing processes that employs maximally efficient orientations. This method finds an orientation for the object that minimizes the number of disconnected components and the distance between the disconnected components that remain, thereby minimizing the time needed for the printer head to traverse empty areas. The method also considers the height of the printed object, its trapped volume, and the number of connected components in each layer. Our novel algorithm considers all four criteria, each weighted according to printer-specific and experimentally-obtained parameters. Preliminary trials demonstrate that this methodology can decrease printing times on fused deposition printers to 45% of that of current state of the art algorithms.

1. Introduction

While additive manufacturing technologies vary and 3D printers use a wide range of materials and processes, most contemporary printers rely on a printing head that deposits one layer of printing material on top of previous layers. While recent advances in the hardware infrastructure of this layer-depositing printing process have led to substantial gains in terms of efficiency and reliability, the software infrastructure used in this process has yet to be fully optimized. Improving 3D printing software is a field of intense research interest, which has led to a number of recent breakthroughs in 3D printing practices. These include algorithms that 3D models commonly contain structural faults that produce defective printed pieces [1–3] and that objects larger than the printing tray can be printed by splitting [4] and packing [5].

A central focus of contemporary 3D printing software research is printing time reduction. Recent research has revealed, for example, that printing time can be shortened by optimizing the printed object's support structures [6–10] or by varying the object's printing orientation [11]. The movement of the printing head itself during printing has also become the focus of improvement. The typical zig-zag printing head movement has been improved by using smooth, continuous Fermat spirals [12]. Moreover, a variety of new algorithms expand the possibilities of 3D printing systems by augmenting structure and printing efficiency in the design phase [13].

While advances in 3D printing technology promise greater precision

and faster printing, there is still significant room for improvement. This article describes a novel 3D printing technique that searches for an object orientations that minimizes print times. The work of Ahsan et al. [14], who observe that printing speed depends on the orientation of the object, on its height, and on the number of connected components in each sliced layer, is instrumental to our own contributions. Chief among these is our observation that the time the printing head spends traveling through the empty space between various filled areas of the object during printing represents a factor yet to be improved. While decreasing the number of connected components (areas) can be achieved by rotating the object (e.g., the letter E has one connected component but if it is rotated 90° to yield  it will have up to three), it is not always possible to decrease this number. If the number of connected components cannot be decreased, the speed of fabrication can nonetheless be increased by minimizing the travel of the printing head over empty areas. This is intuitive when printing multiple simple objects on the printing tray: in this case it is most efficient to simply position them as close to each other as possible. However, this is more complicated for topologically complex objects.

Our method searches for the printing orientation of these complex objects that improves their printing speed. Our algorithm looks for the printing orientation with respect to two global orientation metrics: object height and area required for support structures. While minimizing the void distance traveled by the printing head is key to the optimization process, previous work [14] shows that other factors,

* Corresponding author at: Purdue University, Knott (Maurice G.) Hall of Technology, 401 N. Grant St., Room 363, USA.
E-mail addresses: garcia191@purdue.edu (J.A. García Galicia), bbenes@purdue.edu (B. Benes).

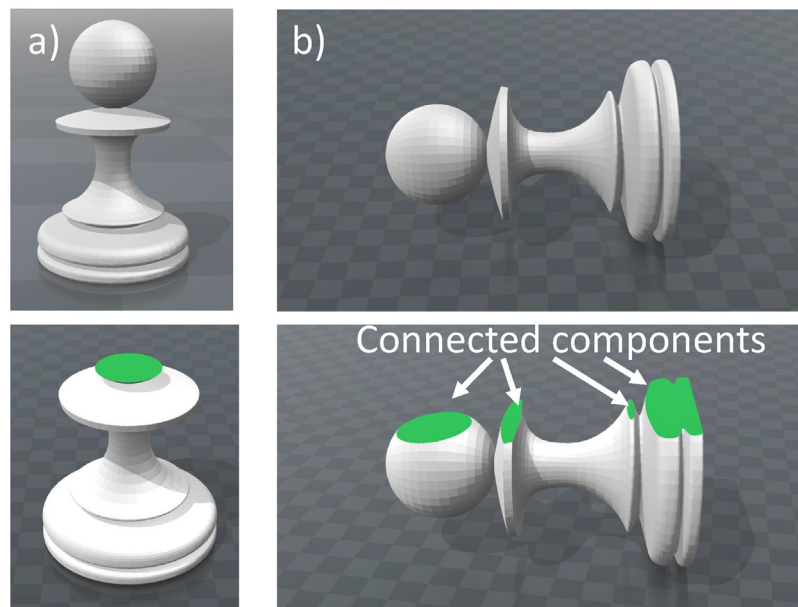


Fig. 1. An object in an orientation that minimizes the number of connected components (a) and in an orientation that considers the height of the object (b). The first orientation takes 69% of the second's printing time.

including the number of connected components, the object's height, and the trapped volume needed for support structures, also affect the printing time. In this study, we analyze the effect of each of these components while incorporating the new metric of traveled distance of the printing head. Moreover, we posit a methodology for finding the weight of the variables in the optimization function for a particular technology (Fused Deposition Modeling) and a model of printer (MakerBot Replicator+).

Our implementation yields a printing time 45% that of current state of the art optimization [14]. Fig. 1 shows an object in a default position (as specified by its creator) as well as in the orientation that maximizes the printing speed detected via current criteria. The orientation in Fig. 1(a) is less suitable because it gives the object a significantly greater height than the object has in Fig. 1(b). However, the reduction in wasted printing head movement makes the first orientation take only 69% of the time of the second. While we tested and verified our approach on the mass-market MakerBot Replicator+ 3D printer, the approach can likely be generalized to any printer that uses additive layering technology with a traveling extruder head.

2. Previous work

This section reviews literature relating to 3D print optimization. This includes research aimed at improving print speed as well as studies of 3D printed objects' visual and structural properties. Additionally, work that examines the process of segmenting the input and packing the pieces is reviewed.

2.1. Structural properties

Volumetric representations of the input models have been used to make voxel-based analyses and detect parts of the model that are not printable [1]. Finite element method (FEM) has also been used to detect areas of structural instability as well as to propose alterations to the model to correct the defects [2]. Similarly, geometric analyses are able to create a map of weaknesses in a model quickly and accurately [3]. Various solutions for these weaknesses have been developed. For instance, introducing honeycomb-like internal structures to a hollow model can increase the model's structural strength [15]. Since the supporting honeycomb structures are contained within the model, the

outer appearance of the model is not altered. One recent study describes a method for calculating internal support structures for the models by using the medial axis transform [10]. Implementing the structural analysis in the design process instead of performing it separately after the piece is finished can result in greater structural integrity [16]. A parametric representation of the surface can also be used to design surfaces that are guaranteed to be self supporting is discussed in [17]. Another recent study describes a topological optimization that is both light and also capable of supporting a predefined load. The study presents a method to fabricate such pieces, minimizing the support structures needed for manufacturing [18]. Rigid body analysis of several plausible loads can be used to solve several FEM problems; the analysis is used to create a probability of failure map [19].

Chen [20] considers the deformation of the model after fabrication. The study describes the fabrication of a modified model that, once deformed by gravity, assumes the original desired shape. Another workaround involves fabricating a structure that approximates the exterior surface of the model in a similar fashion to produce a wireframe of the model. Taking an exemplar pattern and reproducing it on the model's surface ensures that the resulting surface is structurally sound [21].

It is also possible to segment an object so that the separate pieces can be fabricated individually, then assembled to form the original shape. This permits the printing of an object larger than the original printer chamber. Luo et al. [22] are the first to propose an automatic method for segmenting large objects into pieces so that they can be assembled post-print. Vanek [5] builds on this methodology by demonstrating a way to pack the resulting pieces in order to minimize the number of batches to print. Chen et al. [4] contribute further by describing techniques for packing printed pieces more efficiently, thus optimizing the final packing result.

2.2. Supports generation

Structural support has been identified as one of several criteria that can be altered by changing model orientation [23]. Majhi et al. present a method to optimize support for convex models and a mathematical framework to minimize the support volume in 2D (or to be more precise the support area), as well as an optimization algorithm for minimizing the support volume for convex polyhedrons [24].

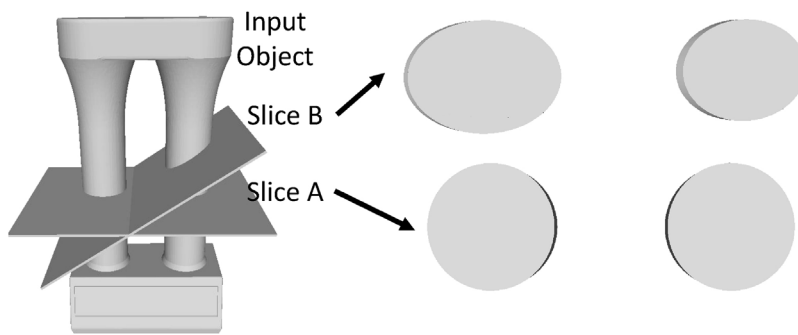


Fig. 2. Illustration of our approach. The pair of columns are sliced in two different ways. Slice A causes the connected areas to be smaller and farther away than the connected areas in Slice B. Although the amount of printed material for the entire object does not change, the printing head spends less time traveling through empty space in Slice B, decreasing the overall printing time.

Ilinkin et al. [25] minimize the surface requiring support rather than the volume because the latter has greater impact in the final quality of the model. A decomposition of the model in pieces in order to avoid the need of support material is presented in Hu [26]. The study employs pyramidal shapes because they do not need extra support to print and they are easy to assemble.

Support structure generation is typically addressed via one of two strategies. These are (1) based on the scaffolding used for creating buildings [27] and (2) based on a tree-like structure that minimizes the material used in the support [8]. Schmidt also proposes a tree-like structure generation technique [9].

When using the techniques of tree-like support generation, the input model can be altered to reduce the overhang area (i.e., the part of the model requiring support during manufacturing) and thus reduce the need for structural support [28]. Another part of the process that can be optimized is the placement of the supports in the overhang area. Yu-xin et al. [6] propose a Poisson sampling to reduce the need of such points. Wang et al. [7] study the creation of an inner support structure not for the manufacturing process itself, but instead for adding structural soundness to a hollow model. Mirzendehtdel and Suresh [18] offer an optimization method that takes into account the need for supporting structures in the final model, rather than during the printing process alone.

2.3. Improving the building process

The early work of Schwerdt et al. [29] recognizes that the orientation problem has multiple facets. In their study, the orientation is constrained to protect certain important parts of the model by disallowing them from being overhangs in the result. Another multi-criteria optimization that considers building time, part quality, and building material is that provided by Phatak and Pande [30]. The optimization method uses a genetic algorithm approach to perform the optimization. This is due to the high computational cost of computing these factors. The quality of the surface is addressed by Delfs et al. [31], who assign a roughness factor to each facet of the object, then optimize the object's orientation with the help of an already-existing database of surface qualities. The work of Ahsan et al. [14] contributes greatly to this article. Ahsan et al. use a two-step optimization process to optimize first the building orientation, then the toolpath direction. Our work differs from that of Ahsan et al. insofar as the latter does not consider the distance between connected components. Ahsan et al. also assume that the toolpath for fabrication lies always in parallel lines. The work

of Wang et al. [11] uses concepts from several of the aforementioned studies to optimize printing by segmenting the print model in several pieces. Because the outer surface of the piece does not require support and is almost aligned with the building direction, connectors can be added to the pieces and the assembly order can be computed. Alexa and Hildebrand [32] assume that the machine is able to perform an adaptive slicing. Thus, the thickness of each layer can vary. The orientation that can produce a slicing of minimum surface error and maximum speed is optimized.

Hon et al. [33] provide one of the first works that recognizes that toolpath generation should be optimized. The study assumes that such toolpaths are always in parallel lines and attempts to optimize a major deposition direction by rotating the object around the building direction. A strategy of creating contour parallel toolpaths near the boundary of a layer and then zig-zag pattern in the interior is discussed by Jin et al. [34]. Later work [35] develops another strategy that calculates an optimal direction for creating parallel toolpaths by using different weights for the speed and the quality of the layer. This latter study also takes into account the time spent in traveling between connected components. Recent research favors toolpath generation techniques that use Fermat spirals, which have the advantage of preserving the object's surface quality as well as improving the speed of manufacturing due to the smooth curvature of the spirals [12]. The method is powerful enough to fill a topologically-complicated region (for example, a region containing many random holes) with a single continuous curve. An analysis of how to plan the toolpath given a topologically-complicated surface already exists, although the toolpath in the analysis is based on parallel infills [36].

However, to the best of our knowledge, no studies have coupled the number of connected components with the distance between them in order to improve the printing time. This paper introduces a method to calculate the optimization parameters for given printing settings.

3. Overview

Our method is based on the observation that an input object can be oriented in such a way that the path traveled by the 3D printing head can be minimized. While the printed areas need to be filled and the printing head must visit every location where the material is deposited, the object orientation affects the amount of time needed for the printing head to travel through empty areas. Fig. 2 demonstrates this approach and Algorithm 1 shows the pseudocode.

Algorithm 1. Pseudocode of our method

Data: 3D mesh as input geometry
Result: Optimal orientation $(\alpha_x, \alpha_z)_{min}$

```

1  $O$  = an array of equidistant spherical orientations  $(\alpha_x, \alpha_z)$ ;
2 for  $j = 1$  to  $|O|$  ; // Traverse all orientations
3 do
4    $(\alpha_x, \alpha_z) \leftarrow O_j$ ;
5    $H_j = H(\alpha_x, \alpha_z)$  ; // Global (height of the orientation)
6    $V_j = V(\alpha_x, \alpha_z)$  ; // Global (trapped volume)
7   Get critical points array  $P$  for  $(\alpha_x, \alpha_z)$ ;
8   Get representative slices array  $R$  for  $(\alpha_x, \alpha_z)$ ;
9    $S_d = 0$  ; // Sum of connected component distances
10   $S_c = 0$  ; // Sum of the number of connected components (CCs)
11  for  $i = 1$  to  $|R|$  ; // For each rep. slice  $r_i$  in this orientation
12  do
13    Calculate the contribution of the slice  $r_i$ ;
14     $S_d += D(r_i, \alpha_x, \alpha_z)$  ; // Local (dist. of the CCs)
15     $S_c += C(r_i, \alpha_x, \alpha_z)$ ; // Local (number of the CCs)
16  end
17   $D_j = S_d$ ;
18   $C_j = S_c$ 
19 end
20 for  $j = 1$  to  $|O|$  ; // Feature scaling
21 do
22    $(H_i^s, V_i^s, D_i^s, C_i^s) = \text{featureScale}(H_i, V_i, D_i, C_i)$ 
23    $f_i = w_h \cdot H_i^s + w_v \cdot V_i^s + w_d \cdot D_i^s + w_c \cdot C_i^s$ 
24 end
25  $(\alpha_x, \alpha_z)_{min} = \min(f)$  ; //  $f$  contains the score of each orientation

```

Previous work [14] used the number of connected components, object height, and the trapped volume needed for support structures to orient the object. This paper builds on this by adding the void distance traveled by the printing head to this existing set of metrics. The focus of this paper on Fused Deposition Modeling and we provide an experimental methodology to find optimal values for the weights of the algorithm for one particular model of printer: the MakerBot Replicator+.

An overview of our system is shown in Fig. 3. The input to our algorithm is a 3D model represented as a triangular watertight mesh

such as an obj or a stl file. The output is its optimal orientation for printing represented as a pair of angles $(\alpha_x, \alpha_z)_{min}$ that correspond to the rotation that needs to be applied to the input model to position it in the optimal orientation.

In the first step, a set of *evaluated orientations* is precomputed. Because it would be unfeasible to evaluate the hypothetically infinite number of possible orientations, it is assumed that printing time does not change significantly for small changes in the object's rotation. Therefore, the continuous domain is discretized into a set of fixed

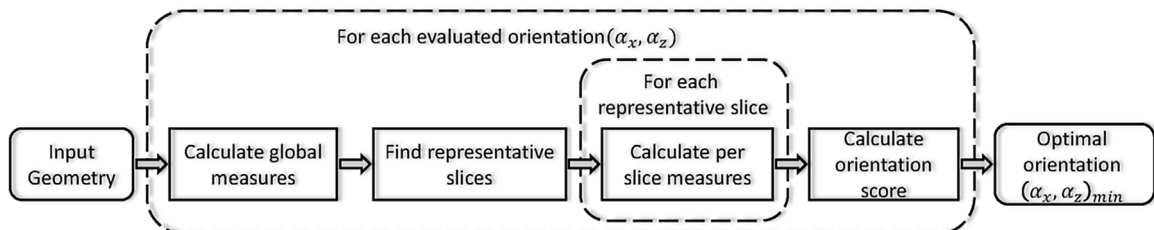


Fig. 3. The input to our method is a 3D geometry and the output is the orientation that minimizes the printing time.

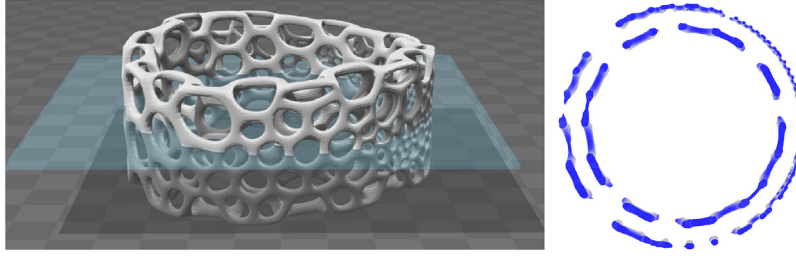


Fig. 4. A slice is generated by the intersection of a plane orthogonal to the printing direction with the mesh. The slice is composed of closed regions that are filled by the material (bracelet courtesy of Nervoussystem 2016).

orientations with a user-defined number of samples. The computation is then treated as a combinatorial problem, which allows us to perform an exhaustive search on all orientations (Section 4).

Our algorithm considers four different criteria during the optimization. We use two *global criteria* for each orientation of the object: the height of the object (denoted by H) and the trapped volume that requiring support structures (denoted by V). In addition, it also considers two local criteria *per layer*: the number of connected components C and their average distance D .

Typically, the height of the orientation H should be minimized, as printing in the vertical direction y is usually significantly slower than printing in the direction parallel to the printing plane. The trapped volume V should also be minimized because it requires additional support structures, which require time and material to build. Note that we do not examine the way these supports are constructed. Various methods exist for optimizing support structures, all of which can significantly affect the printing time [37,27,8,38,18].

Having large numbers of disconnected components C is not beneficial because the printer needs to stop printing and move the printing head between them. If having disconnected components is inevitable, it is beneficial to minimize their distance D so that idle head movements are minimized.

Once the orientations of the input object are precomputed, the *printing score* for each of them is calculated. The score is a two-component value. Let's denote the first *the orientation score*. This includes the trapped volume V and the total height of the mesh H . The second component of the orientation score varies for each slice and it is denoted as *the slice score*. It includes the number of connected components C and the distance between them D (Section 5). In order to calculate the slice score for a given orientation score, it is necessary to virtually slice the object. However, the overall score per orientation can be effectively estimated by a smaller set of *representative slices*. This improves the efficiency of the score calculation, as slicing is generally a time-consuming procedure. The representative slices impersonate the slices that have the same number of connected components per slice (Section 4.2). The *slice score* is the score of the representative slice weighted by the number of layers this representative slice represents.

The weight of each element of the printing score has only been approximated in previous research. We show that it can be accurately estimated for a particular 3D printer by performing simple tests. Specific weights to the MakerBot Replicator+ 3D are provided.

The detected orientation is the pair of angles $(\alpha_x, \alpha_z)_{\min}$ that minimizes the overall printing time.

4. Score function

The input object is located in the origin of the coordinate system and the printing axis is aligned with y direction. The object can be rotated around two angles (α_x, α_z) to a new orientation. The angle $\alpha_x \in [0, 2\pi]$ represents the amount of the rotation around the x axis and the angle $\alpha_z \in [-\pi, \pi]$ represents rotation around the z axis.

Once the orientation is defined, a plane orthogonal to the building direction moves by a small Δy and the model is sliced. A slice consists of

a set of closed regions for watertight meshes (see an example in Fig. 4).

The *toolpath* plans the nozzle movement that deposits the material to fill the slice. Most printers plan the toolpath as a set of parallel lines (i.e., a zig-zag pattern) but, newer printers provide more control over the nozzle. These printers can plan paths concentric to the boundary of the regions (*contour parallel* paths), which improve the quality of the final results. However, if the region is not convex, the printing path can be discontinuous. Recent work [12] shows that continuous spiral fill can be provided for any region.

4.1. Sampling the orientations

The spherical domain of pairs (α_x, α_z) , where $\alpha_x \in [0, 2\pi]$ and $\alpha_z \in [-\pi, \pi]$ is sampled by using regular recursive subdivision of the sphere to find the sampling directions.

The goal of our search algorithm is to find a minimum of the score function $f(\alpha_x, \alpha_z)$:

$$\min f(\alpha_x, \alpha_z) = w_h \cdot H + w_v \cdot V + \sum_i r_i (w_d \cdot D + w_c \cdot C), \quad (1)$$

where $H(\alpha_x, \alpha_z)$ is the height of the object, $V(\alpha_x, \alpha_z)$ is the trapped volume, $D(\alpha_x, \alpha_z)$ is the connected component distance, $C(\alpha_x, \alpha_z)$ is the connected distance number, and w represents the corresponding weights for each criterion (henceforth, the (α_x, α_z) notation is omitted). The two global criteria H and V are calculated for the entire object. The two local criteria D and C that are calculated for the i th representative slice. Section 5 describes details of the calculations of each component.

Since the criteria H , V , D and C have different units and ranges, a feature scaling is performed

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)},$$

before they are used in Eq. (1).

4.2. Representative slices

Slicing the object is time consuming, as is evaluating per each individual slice. Instead of evaluating each slice, we introduce the concept of *representative slices*. These are slices representing a set of slices with the same topology. By sweeping the object in the printing direction, a Reeb graph of the input object is constructed. This is a graph with critical points where the topology of the slice changes (i.e., the number of connected components) increases or decreases. For a given orientation, the plane sweeps from the tray up to the building direction. A new region appears when the plane reaches a valley in the surface defined by the mesh. Similarly, a region disappears when the plane reaches a peak of the surface. These points are called *critical points*, as shown in Fig. 5. The critical points can be detected by comparing the surface normal to the printing direction without actually slicing the object.

The critical points p_i are sorted according to their projection $h(p_i)$. Two successive critical points define a *slab*. The heights $h(p_i)$ are the slab boundaries (Fig. 5 middle). Each slab has a constant number of

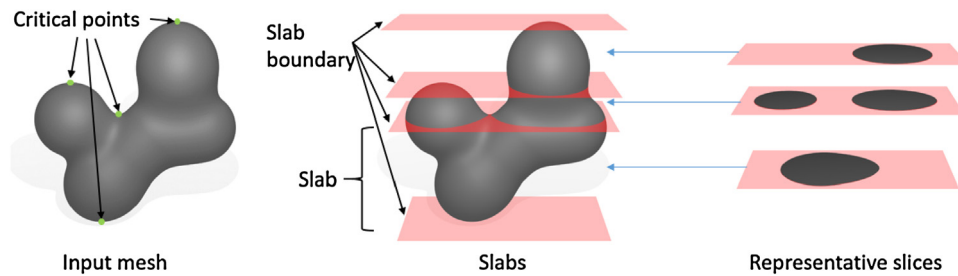


Fig. 5. The critical points (right), define a set of boundaries that also define a slab (middle). Only one representative slice per slab is taken (left).

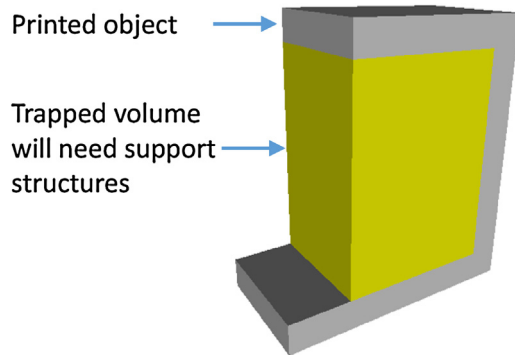


Fig. 6. Trapped volume requires support structures for printing.

components, as shown in Fig. 5. The *representative slice* is then the slice taken from the slab center.

Since the various slabs have different heights and volumes, a weighting factor r_i for each representative slice is calculated as:

$$r_i = \frac{h(p_{i+1}) - h(p_i)}{h(p_n) - h(p_0)}, \quad (2)$$

where $h(p_i)$ and $h(p_{i+1})$ are the heights of the critical points and $h(p_0)$, $h(p_n)$ are the first and the last critical points.

5. Orientation score

The search algorithm uses two global criteria (the height of the object H and its trapped volume V) as well as two local criteria that are calculated per slice (the distance of the connected components D and

the number of connected components C).

5.1. Global criteria

For each orientation (α_x, α_z) , its global score is first calculated. The height H of the object in the given orientation is simply the difference between the maximum and the minimum of its vertex y coordinates. The algorithm attempts to find the orientation with the minimal the object height because the printer moves faster in the horizontal direction, which makes it generally beneficial to have a low height for the printed object. However, this maximum is not absolute, as compounding variables can make high-height objects desirable (see Fig. 1).

Most 3D printers need to use support structures while printing. The amount of support required is a factor that must be considered in the print time. Since the algorithm used to build the supports can vary, we use the strategy developed by Majhi et al. [37], whose methodology minimizes the trapped volume V (Fig. 6). It should be noted that minimizing the support may not be beneficial for certain technologies. For example in selective laser melting (SLM) the support structures help heat diffusion during printing process. In this work, FDM and printing speed optimization is used. It is assumed that the printing time of the support material is proportional to the trapped volume. The trapped volume is calculated as the volume under all areas that require support and a GPU-optimized algorithm from [8] to calculate this volume is used.

5.2. Per slice criteria

The intersection of the input polygonal object with the cutting plane defines a slice. Because the object is a polygonal watertight triangular mesh, a slice is composed of a series of line segments that form closed

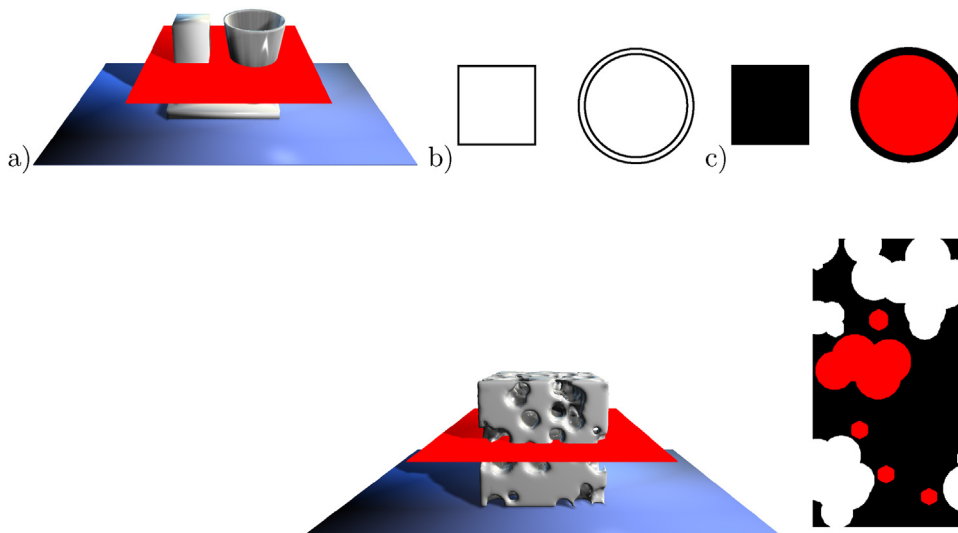


Fig. 7. Two objects are sliced (a) and a set of contours is generated (b). The contours define positive regions that require filling (black) and negative regions (red) that are to be left empty (c). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

Fig. 8. An object (left) and a slice (right) with positive regions in black and negative regions in red. Note the white regions are actually outside the object.

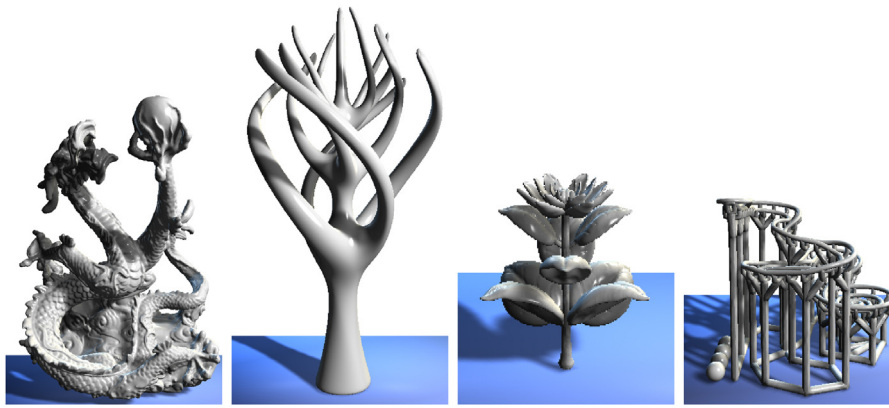


Fig. 9. Some of the test objects used to estimate the weights.

contours (see Fig. 7). The area inside each contour, then, defines a region that must be filled with material. Depending on the topology of the original mesh, one contour can be contained inside another. If a region needs to be filled during the printing, we refer to it as a *positive region*. We refer to regions that do not require filling (holes) as *negative regions* (Fig. 7).

From this step, the number of positive regions C in the representative slices is calculated (these are called *contour plurality* in Ahsan et al. [14]).

The distance of two positive regions is defined as the minimum distance between a point in one region to a point in the other. The last per slice component D is the average distance between all positive regions of the slice (Fig. 8).

6. Determining the weights

The objective function equation (1) has a set of user weights $w = (w_h, w_v, w_d, w_c)$ that must be specified before the search algorithm is run. These parameters depend on the printer and other computational factors including the slicer and the path planning for the printing. The interaction of these parameters is not yet fully understood. Previous work [14] used a set of user-defined fixed values, while in this work weights are determined experimentally for one particular printer model (the MakerBot Replicator+), but the way it has been determined could be used for different printer or the same printer with different firmware.

A total of $k = 20$ objects with widely varying shapes and topologies were evaluated (Fig. 9). To do this, each parameter's space was sampled with sampling distance $\Delta = 0.1$. Because there are four parameters to determine, we created $n = 1/\Delta + 1$ samples and generated all possible n^4 tuples of (w_h, w_v, w_v, w_c) ; these led to $\approx 14k$ ordinations to be evaluated.

For each model, we found the optimal orientation for each set of weights. However, there were some duplicates. For example, all tuples (w_h, w_v, w_v, w_c) and $(\lambda w_h, \lambda w_v, \lambda w_v, \lambda w_c)$ for some λ resulted in the same orientation.

It would not have been feasible to print several thousands of objects to measure their printing times. Thus, after the weights have been found, the printer simulator Printron¹ was used. The program allowed us emulate the settings of a particular printer to estimate the time that model would take to print for each orientation. The orientation with the minimal time reported by the software were recorder. These were the optimal parameters for that particular model.

The process was repeated for each model, taking the arithmetic median of all the optimal tuples to obtain the printer parameters.

The resulting values for the normalized weights in our settings are: $w_h = 0.34$, $w_v = 0.24$, $w_d = 0.11$, and $w_c = 0.31$.

7. Results and evaluation

One central finding of our work was the collection of weights for optimal orientation on the MakerBot Replicator+ 3D printer. Our results were compared to those of Ahsan et al. [14]; however, the authors do not report optimal weights for their algorithm. Thus, to make the comparison fair, we repeated the optimal weight evaluation from Section 6 using data from Ahsan et al. This led to the set of weights of $w_h = 0.39$, $w_v = 0.26$, and $w_c = 0.35$. Our algorithm was then compared with the previous work [14] by using optimal weights for both.

We simulated printing 953 models from the SHREC 2015 dataset.² The repository contains 1, 200 models, but those that were not printable were not used.

Our evaluation also simulated the printing in the orientations obtained by the two weight sets: one for our algorithm and one for the Ahsan et al. algorithm [14] (the printing time ratio of both methods is reported in Fig. 10). If the ratio was equal to one, both algorithms converged to a solution that required the same printing time. Ratios greater than one suggested that the distance of the connected components in the slice should be considered because this converges to a faster solution. In other words, ratios greater than one suggested our method improved printing speed. For some objects, the speed increase was 45%. Ratios less than one suggested the opposite: that our method decreased performance. The worst case for our algorithm was a 50% performance hit.

Overall, 62% of the cases of both algorithms arrived at the same orientation. The results also showed that considering the traveling distance can lead to worse performance in 23% of the cases. However, for the remaining 15%, considering the distance of the connected components led to a faster solution.

After all tests were performed, the three best and three worst objects from both sets were printed and the printing time for both cases was measured. Table 1 shows the actual printing time and the printed objects in their orientations. These printings confirm our findings. The third object is an example of a case where D becomes more important than both H and C , as discussed in Fig. 1.

The cases that do not show speed increases for our method occurred most likely from underestimating V in our experiments. This could suggest that thin, elongated objects that need a support and do not have a clear difference in C were not accurately represented in the determination of the optimal weights.

Although the actual printing and simulated time varied significantly (probably because of some systematic error in the simulation software), the ratios of the measured times agreed for the simulated and the measured printing times. Thus, we believe the simulation can be used to approximate the actual printing processes with accuracy.

¹ www.pronterface.com.

² <http://www.itl.nist.gov/iad/vug/sharp/contest/2015/Range/data.html>.

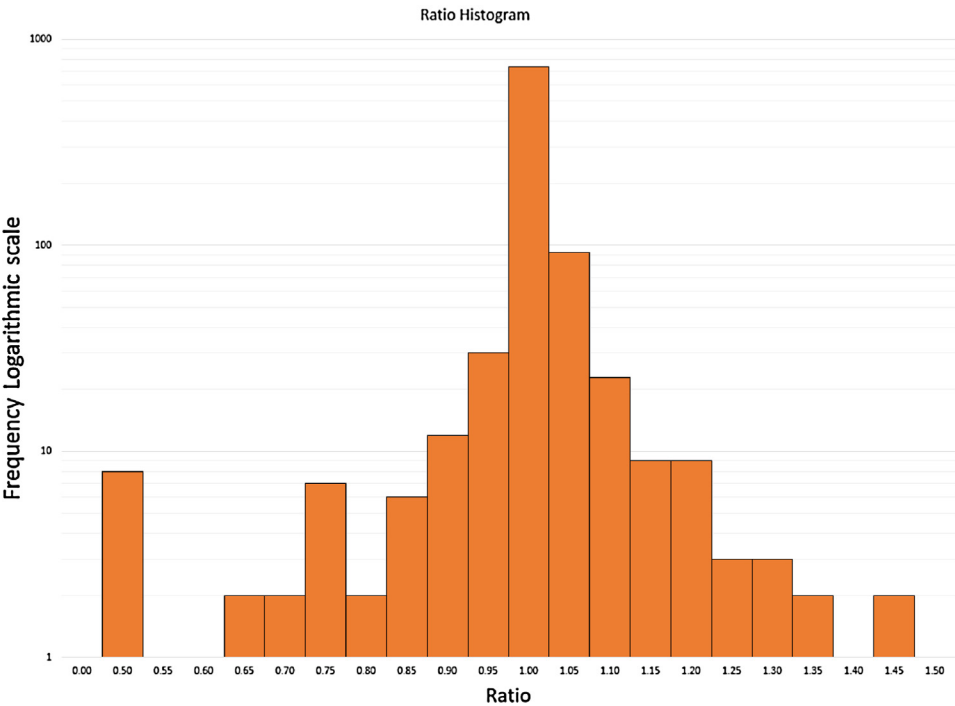
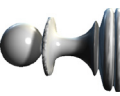


Fig. 10. Histogram of the ratios of printing times for 954 objects. A ratio greater than one suggests that using the four criteria is better; smaller than one suggests that using only three criteria (the method suggested by Ahsan et al. [14]) is better.

Table 1
Printing times for the cases that considered or did not consider the distance of connected components.

Our method Time in seconds			Method [14] Time in seconds			Comparison Ratio	
Orientation	Actual	Simulated	Orientation	Actual	Simulated	Measured	Simulated
<i>Prefers distance of connected components</i>							
	131	809		288	1434	2.19	1.77
	953	5847		1391	8418	1.45	1.44
	2609	10,116		3804	16,492	1.46	1.63
<i>Does not prefer distances of connected components</i>							
	671	1824		321	423	0.48	0.23
	1180	1327		737	224	0.62	0.17
	904	2263		605	782	0.67	0.35

8. Conclusions and future work

This paper introduced a novel algorithm for finding a printing orientation for a 3D object that minimizes its printing time. The key observation is that the time spent traveling over empty areas of the printed slice can be further minimized. This criterion has been incorporated into an algorithm that also considers the object height, the trapped volume, and the number of connected components. Furthermore, this paper also provided a systematic method for finding optimal weights for a given printer. The new method was evaluated by comparing it to current state of the art practices. In this comparison, our method demonstrates up to a 45% decrease in printing time.

The main limitation of our work derives from the way the weights have been found. The new algorithm used simulation software that needs to be carefully calibrated for an existing printer, but the calibration as well as the measured values can vary depending on many unknown variables. It would be preferable to actually print all the objects and measure their printing time, but this is time- and cost-prohibitive. Our algorithm also uses several space discretizations (the orientations and the slicing process) that may have an effect on the final results. Moreover, the choice of the representative slice from the slab could also affect the results.

One avenue for the future work would be to provide finer levels of granulation for the trapped volume. We consider its minimization to be an important factor. However, various approaches for building support structures already exist [37,27,8,38,18]; their time can significantly vary, affecting the printing time. Nonetheless, including them in the optimization could produce richer data. The authors are not aware of any full object path travel head optimization. Moreover, existing work focuses only on one slice [12]. Future work could consider slice coherence. Our main criterion is the speed of the printing. However, the speed may not always be the most important factor. For example, some applications require high quality of the surface of the 3D printed object as addressed for example in [4,11,14,31]. This depends, among other aspects, on the printing orientation, because the horizontal resolution is often different from the printing bed elevation step. This could be also considered as the part of our algorithm. For some printers, it is preferable to have object located horizontally, because of the structural properties that may also contradict the printing speed criterion. Moreover, the weights $w = (w_h, w_v, w_d, w_c)$ in Eq. (1) are determined experimentally. If their effect, and their mutual relation would be known, it would be interesting to allow the user to modify their values to affect the printing.

Acknowledgements

This research was funded in part by National Science Foundation grants “MRI: Development of a Next-Generation 3-D Printer for Smart Product Design - Purdue PolymerMakers” IIS 1726865 and “EAGER: Collaborative Research: Inverse Procedural Material Modeling for Battery Design” IIS 1608762.

References

- [1] A. Telea, A. Jalba, Voxel-based assessment of printability of 3D, *Mathematical Morphology and Its Applications to Image and Signal Processing*, Springer Berlin Heidelberg, 2011, pp. 393–404, http://dx.doi.org/10.1007/978-3-642-21569-8_34.
- [2] O. Stava, J. Vanek, B. Benes, N. Carr, R. Měch, Stress relief: improving structural strength of 3D printable objects, *ACM Trans. Graph.* 31 (4) (2012), <http://dx.doi.org/10.1145/2185520.2185544> 48:1–48:11.
- [3] Q. Zhou, J. Panetta, D. Zorin, Worst-case structural analysis, *ACM Trans. Graph.* 32 (4) (2013), <http://dx.doi.org/10.1145/2461912.2461967> 137:1–137:12.
- [4] X. Chen, L. Lu, R. Hu, D. Cohen-or, B. Chen, Dapper: decompose-and-pack for 3D printing, *ACM Trans. Graph.* 34 (6) (2015) 213, <http://dx.doi.org/10.1111/cgf.12811>.
- [5] J. Vanek, J.A.G. Galicia, B. Benes, R. Měch, N. Carr, O. Stava, G.S. Miller, PackMerger: a 3D print volume optimizer, *Comput. Graph. Forum* (2014), <http://dx.doi.org/10.1111/cgf.12353>.
- [6] M. Yu-xin, W. Li-fang, Q. Jian-kang, W. Runyu, An optimized scheme to generating support structure for 3D printing, *Image and Graphics vol. 1*, Springer International Publishing, 2015, pp. 571–578, <http://dx.doi.org/10.1007/978-3-319-21969-1>.
- [7] W. Wang, T.Y. Wang, Z. Yang, L. Liu, X. Tong, W. Tong, J. Deng, F. Chen, X. Liu, Cost-effective printing of 3D objects with skin-frame structures, *ACM Trans. Graph.* 32 (6) (2013) 1–10, <http://dx.doi.org/10.1145/2508363.2508382>.

- [8] J. Vanek, J.A.G. Galicia, B. Benes, Clever support: efficient support structure generation for digital fabrication, *Symposium on Geometry Processing*, vol. 33 (5) (2014), <http://dx.doi.org/10.1111/cgf.12437>.
- [9] R. Schmidt, N. Umetani, Branching support structures for 3D printing, *ACM SIGGRAPH 2014 Conference Proceedings*, 10–14th August 9, 2014, p. 1, <http://dx.doi.org/10.1145/2619195.2656293>.
- [10] X. Zhang, Y. Xia, J. Wang, Z. Yang, C. Tu, W. Wang, Medial axis tree: an internal supporting structure for 3D printing, *Comput. Aided Geom. Des.* 35–36 (2015) 149–162, <http://dx.doi.org/10.1016/j.cagd.2015.03.012>.
- [11] W. Wang, C. Zanni, L. Kobbelt, Improved surface quality in 3D printing by optimizing the printing direction, *Comput. Graph. Forum* 35 (2) (2016).
- [12] H. Zhao, Y. Chen, C. Tu, B. Chen, Connected Fermat spirals for layered fabrication, *ACM Trans. Graph. (Proc. ACM SIGGRAPH)* 35 (4) (2016), <http://dx.doi.org/10.1145/2897824.2925958> 100:1–100:10.
- [13] H. Lipson, M. Kurman, *Fabricated: The New World of 3D Printing*, John Wiley & Sons, 2013.
- [14] A.N. Ahsan, A. Habib, B. Khoda, Resource based process planning for additive manufacturing, *Comput. Aided Des.* (2015), <http://dx.doi.org/10.1016/j.cad.2015.03.006>.
- [15] L. Lu, A. Sharf, H. Zhao, Y. Wei, Q. Fan, X. Chen, Y. Savoye, C. Tu, D.C.-o. Baoquan, Build-to-last: strength to weight 3D printed objects, *ACM Trans. Graph.* 33 (4) (2014) 1–10, <http://dx.doi.org/10.1145/2601097.2601168>.
- [16] Y. Xie, W. Xu, Y. Yang, X. Guo, K. Zhou, Agile structural analysis for fabrication-aware shape editing, *Comput. Aided Geom. Des.* 35–36 (2015) 163–179, <http://dx.doi.org/10.1016/j.cagd.2015.03.019>.
- [17] M. Miki, P. Block, Parametric self-supporting surfaces via direct computation of airy stress functions, *ACM Trans. Graph.* 34 (4) (2015), <http://dx.doi.org/10.1145/2766888> 89:1–89:12.
- [18] A.M. Mirzendehehdel, K. Suresh, Support structure constrained topology optimization for additive manufacturing, *Comput. Aided Des.* (2016), <http://dx.doi.org/10.1016/j.cad.2016.08.006>.
- [19] T. Langlois, A. Shamir, D. Dror, W. Matusik, D.I.W. Levin, Stochastic structural analysis for context-aware design and fabrication, *ACM Trans. Graph.* 35 (6) (2016), <http://dx.doi.org/10.1145/2980179.2982436> 226:1–226:13.
- [20] X. Chen, An asymptotic numerical method for inverse elastic shape design, *ACM Trans. Graph.* 33 (4) (2014), <http://dx.doi.org/10.1145/2601097.2601189>.
- [21] D. Lorraine, C. Dick, By-example synthesis of structurally sound patterns, *ACM Trans. Graph.* 34 (4) (2015) 1–12.
- [22] L. Luo, I. Baran, S. Rusinkiewicz, W. Matusik, Chopper, *ACM Trans. Graph.* 31 (6) (2012) 1, <http://dx.doi.org/10.1145/2366145.2366148>.
- [23] J. Majhi, R. Janardan, M. Smid, J. Schwerdt, Multi-criteria geometric optimization problems in layered manufacturing, *Proceedings of the Fourteenth Annual Symposium on Computational Geometry – SCG’98* (1998) 19–28, <http://dx.doi.org/10.1145/276884.276887>.
- [24] J. Majhi, R. Janardan, M. Smid, P. Gupta, On some geometric optimization problems in layered manufacturing, *Comput. Geom.* 12 (3–4) (1999) 219–239, [http://dx.doi.org/10.1016/S0925-7721\(99\)00002-4](http://dx.doi.org/10.1016/S0925-7721(99)00002-4).
- [25] I. Ilinkin, R. Janardan, M. Smid, E. Johnson, P. Castillo, J. Schwerdt, Heuristics for estimating contact area of supports in layered manufacturing, *J. Exp. Algorithm.* 11 (1) (2007) 1–6, <http://dx.doi.org/10.1145/1187436.1210589>.
- [26] R. Hu, Approximate pyramidal shape decomposition, *ACM Trans. Graph.* 33 (6) (2014) 213–225.
- [27] J. Dumas, U.D. Lorraine, J. Hergel, S. Lefebvre, Bridging the gap: automated steady scaffolds for 3D printing, *SIGGRAPH 2014*, vol. 33 (4) (2014) 98–108, <http://dx.doi.org/10.1145/2601097.2601153>.
- [28] K. Hu, X. Zhang, C.C.L. Wang, Direct computation of minimal rotation for support slimming, *2015 IEEE International Conference on Automation Science and Engineering (CASE)* (2015) 936–941.
- [29] J. Schwerdt, M. Smid, R. Janardan, E. Johnson, Protecting critical facets in layered manufacturing: implementation and experimental results, *Comput. Aided Des.* 35 (7) (2003) 647–657, [http://dx.doi.org/10.1016/S0010-4485\(02\)00090-8](http://dx.doi.org/10.1016/S0010-4485(02)00090-8).
- [30] A.M. Phatak, S.S. Pande, Optimum part orientation in Rapid Prototyping using genetic algorithm, *J. Manuf. Syst.* 31 (4) (2012) 395–402, <http://dx.doi.org/10.1016/j.jmsy.2012.07.001>.
- [31] P. Delfs, M. Tows, H.J. Schmid, Optimized build orientation of additive manufactured parts for improved surface quality and build time, *Addit. Manuf.* 12 (2015) 314–320, <http://dx.doi.org/10.1016/j.addma.2016.06.003>.
- [32] M. Alexa, K. Hildebrand, Optimal discrete slicing, *ACM Trans. Graph.* 36 (1) (2017).
- [33] M.C. Hon, R. Janardan, J. Schwerdt, M. Smid, Minimizing the total projection of a set of vectors, with applications to layered manufacturing, *Comput. Aided Des.* 35 (1) (2003) 57–68, [http://dx.doi.org/10.1016/S0010-4485\(01\)00175-0](http://dx.doi.org/10.1016/S0010-4485(01)00175-0).
- [34] G.Q. Jin, W.D. Li, L. Gao, An adaptive process planning approach of rapid prototyping and manufacturing, *Robot. Comput. Integr. Manuf.* 29 (1) (2013) 23–38, <http://dx.doi.org/10.1016/j.rcim.2012.07.001>.
- [35] Y. Jin, Y. He, J. Zhong Fu, W. Feng Gan, Z. wei Lin, Optimization of tool-path generation for material extrusion-based additive manufacturing technology, *Addit. Manuf.* 14 (2014) 32–47, <http://dx.doi.org/10.1016/j.addma.2014.08.004> (inaugural issue).
- [36] Z. Lin, J. Fu, Y. Sun, Q. Gao, G. Xu, Z. Wang, Non-retraction toolpath generation for irregular compound freeform surfaces with the LKH TSP solver, *Int. J. Adv. Manuf. Technol.* (2017) 1–15, <http://dx.doi.org/10.1007/s00170-017-0247-8>.
- [37] J. Majhi, R. Janardan, J. Schwerdt, M. Smid, P. Gupta, Minimizing support structures and trapped area in two-dimensional layered manufacturing, *Comput. Geom.* 12 (3–4) (1999) 241–267, [http://dx.doi.org/10.1016/S0925-7721\(99\)00003-6](http://dx.doi.org/10.1016/S0925-7721(99)00003-6).
- [38] S.-s. Huang, H. Fu, L. Wei, S.-M. Hu, Support substructures: support-induced part-level structural representation, *IEEE Trans. Vis. Comput. Graph.* 2626 (XX) (2015) 1, <http://dx.doi.org/10.1109/TVCG.2015.2473845>.