Variable-width contouring for additive manufacturing

Hornus, Samuel; Kuipers, T.; Devillers, Olivier; Teillaud, Monique; Martínez, Jonàs; Glisse, Marc; Lazard, Sylvain; Lefebvre, Sylvain

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Variable-width contouring for Additive Manufacturing

SAMUEL HORNUS, Université de Lorraine, CNRS, Inria, LORIA
TIM KUIPERS, Ultimaker and Department of Sustainable Design Engineering, Delft University of Technology
OLIVIER DEVILLERS and MONIQUE TEILLAUD, Université de Lorraine, CNRS, Inria, LORIA
JONÀS MARTÍNEZ, Université de Lorraine, CNRS, Inria, LORIA
MARC GLISSE, Inria
SYLVAIN LAZARD and SYLVAIN LEFEBVRE, Université de Lorraine, CNRS, Inria, LORIA
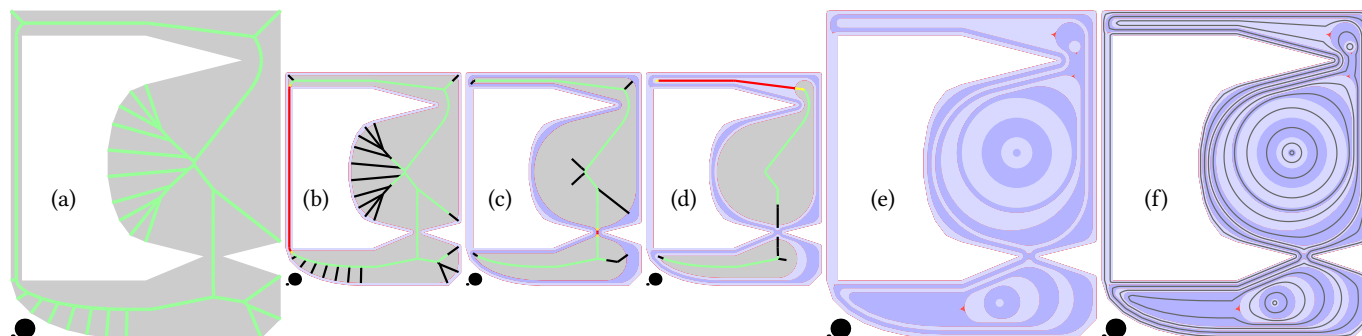
Fig. 1. In this example, we fill the input shape (a) with beads of width varying between 0.3 mm and 1.5 mm, as visualized as the diameter of the black disks. We trim the medial axis and label parts variously (colors) in order to model the (b) first, (c) second, (d) third bead, etc. All the beads are visible in (e) and the print trajectories in (f). In (e,f), the eight small triangular red areas are gaps not covered by any bead.

In most layered additive manufacturing processes, a tool solidifies or deposits material while following pre-planned trajectories to form solid *beads*. Many interesting problems arise in this context, among which one concerns the planning of trajectories for filling a planar shape as densely as possible. This is the problem we tackle in the present paper. Recent works have shown that allowing the bead width to vary along the trajectories helps increase the filling density. We present a novel technique that, given a deposition width range, constructs a set of closed beads whose width varies within the prescribed range and fill the input shape. The technique outperforms the state of the art in important metrics: filling density (while still guaranteeing the absence of bead overlap) and trajectories smoothness. We give a detailed geometric description of our algorithm, explore its behavior on example inputs and provide a statistical comparison with the state of the art. We show that it is possible to obtain high quality fabricated layers on commodity FDM printers.

CCS Concepts: • **Computing methodologies → Shape modeling**; • **Applied computing → Computer-aided design**.

Authors' addresses: Samuel Hornus, Université de Lorraine, CNRS, Inria, LORIA, F-54000, Nancy, France; Tim Kuipers, Ultimaker, Utrecht , Department of Sustainable Design Engineering, Delft University of Technology, The Netherlands; Olivier Devillers; Monique Teillaud, Université de Lorraine, CNRS, Inria, LORIA, F-54000, Nancy, France; Jonàs Martínez, Université de Lorraine, CNRS, Inria, LORIA, F-54000, Nancy, France; Marc Glisse, Inria, Palaiseau, France; Sylvain Lazard; Sylvain Lefebvre, Université de Lorraine, CNRS, Inria, LORIA, F-54000, Nancy, France.

Additional Key Words and Phrases: 3D printing, dense infill, medial axis

1 INTRODUCTION

In additive manufacturing (AM), *process planning* refers to the sequence of operations required to transform an input 3D model into a set of instructions for a target machine. These instructions define the strategies the machine will execute to solidify material and form a solid object. Our work is developed in the context of those strategies that rest on the ability to solidify material along planar trajectories or toolpaths. The solidified material along a trajectory is called a *bead*. On many technologies – such as selective laser sintering (SLS), stereolithography (SLA) and fused filament fabrication (FFF) – the instructions are vector trajectories enriched with information such as energy levels or material flow. When following the trajectories, the machine produces solidified beads that join together to form a planar solid layer. The layers are then sequentially solidified on top of each other to obtain a final solid object.

Process planning raises many interesting geometric problems, with direct practical implications. Amongst these, the question of how to solidify the area of each object layer has received much attention (we review prior works in Section 2).

In this context, we focus on the specific problem of densely covering a layer. A critical aspect of such *dense infills* is to avoid situations
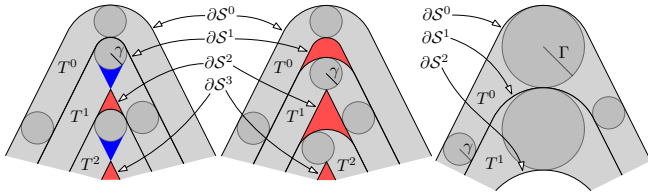
Fig. 2. *Left.* Uniform-width contouring. Beads have constant width 2γ and produce overfill as self-overlap in the blue areas. The inner red areas are not covered: we have underfill. *Middle.* Regularized parallel contouring avoids overfill. Beads have constant width 2γ. *Right.* Variable-width contouring (our technique) models beads with varying width in the range $[2\gamma, 2\Gamma]$.

where the produced beads overlap or self-overlap uncovered gaps between them (*underfill*). Such issues appear with common approaches, where beads are defined as the space between successive shapes, obtained by an iterated parallel offset (or morphological erosion). In *uniform-width contouring* (Figure 2 left), beads of width 2γ are obtained by applying a parallel offset of width 2γ to the shape, resulting in both overfill and underfill [Eiamsa-ard et al. 2003; Yang et al. 2002]. *Regularized parallel contouring* (Figure 2 middle) computes each nested shape by applying an offset of width 4γ (instead of 2γ) followed by a morphological dilation by a disk of diameter 2γ [MFX 2017]. The beads still follow boundary-parallel trajectories, which may create underfill but not overfill. Indeed, even in the favorable case of a feature that is not too sharp, note in the figure how the nested shapes become sharp after the first two beads are extracted, which creates a gap at each subsequent step.

While underfill threatens the structural strength, overfill may lead to fabrication failures [Kuipers et al. 2019; Wenbiao et al. 2002]. A dense, smooth, and precise fill leads to stronger parts and increased manufacturing reliability [Kao and Prinz 1998]. To achieve this, we exploit the ability of many AM processes to vary the solidification radius dynamically along a deposition trajectory [Ding et al. 2016; Wang et al. 2019]. However, bead width variations have to remain within minimum and maximum bounds for proper solidification. This makes the filling problem harder and of particular interest. Current processes are nevertheless not optimized for varying width deposition and there are practical difficulties that we discuss in Section 6.2.

We present an algorithm that leaves significantly fewer underfill (gaps) than existing techniques, guarantees the absence of overfill, while producing smooth, cyclic deposition trajectories. Its output is illustrated in Figure 2 right for a simple case, and in Figure 1 for a more complex case. We give a detailed overview of our algorithm in Section 3 and a complete geometric description in Section 5. We then provide in Section 6 an experimental assessment focused on fused filament processes (FDM) in order to demonstrate the applicability of the technique. We discuss limitations in the conclusion. In particular, our algorithm does not achieve a globally minimal underfill; Finding the global minimum remains an open problem.

The source code for this work is available at https://github.com/mfx-inria/Variable-width-contouring.

2 PREVIOUS WORK

Solidification inside the volume of an object is one of the main driving factors of fabrication time [Livesu et al. 2017]. Thus, a line of research explores how to make insides print faster, for instance with sparse patterns [McMains et al. 2000], thicker layers [Sabourin et al. 1997], or by maximally emptying the part while still ensuring fabricability under overhang and support constraints [Hornus and Lefebvre 2018; Lee and Lee 2016; Tricard et al. 2019; Wang et al. 2017b]. Under the same constraints, other approaches seek to optimize the internal properties to achieve specific mechanical objectives, such as balance [Prévost et al. 2013; Wu et al. 2016], rigidity [Li et al. 2015; Wang et al. 2017a], density [Kuipers et al. 2019] or elastic response orientation [Martínez et al. 2018].

There are, however, many situations where a dense cover is desired, for instance on the tops and bottoms of an object, but also to increase the strength and rigidity of a piece. A first approach for densely covering a contour with trajectories is to rely on zig-zag patterns [McMains et al. 2000]. This strategy, however, leads to underfill and overfill where paths connect to the enclosing contour, requiring trade-offs [Jin et al. 2013]. A second approach is to follow contour parallel trajectories [Eiamsa-ard et al. 2003; Yang et al. 2002]. These paths closely follow the contour outline and typically provide an improved surface finish. Unfortunately, as mentioned in the introduction, this approach also leaves many gaps where contours from both sides fail to meet exactly along the medial axis [Kao and Prinz 1998]. Note that in practice, a few contour parallel paths are often used first, followed by a zig-zag fill.

Another important concern is to avoid interrupting the solidification process, and to produce as-smooth-as-possible trajectories. This is especially important on extrusion-based systems, where pausing and resuming the flow is challenging due to pressure buildup. Thus, several works considered how to reconnect paths into continuous trajectories, for both zig-zag and contour parallel patterns [Ding et al. 2014; Jin et al. 2017b; Zhao et al. 2016].

To reduce underfills and overfills, Moesen et al. [2011] consider how to preserve key features of a part that would otherwise be removed by the initial offsetting. In particular, the algorithm widens narrow regions to accommodate a single bead, while preserving the initial layer topology.

Kao et al. [1998] considered exploiting a varying solidification width early on. Their proposed method optimizes and modifies the contour outline to ensure it can be fully filled – without gaps – with varying width beads between a minimum and maximum bound. However, to reproduce the original contour the smooth beads have to be clipped, thus producing many open paths. Ding et al. [2016] also exploit a varying solidification width to eliminate gaps. The proposed approach is to generate the same number of beads in between the outer contour and an inner skeleton of the slice. As there are the same number of beads across, their widths vary simultaneously to adapt to the local shape thickness around the skeleton: the bead widths shrink and expand. While this proves effective to generate beads of varying widths, the technique cannot bound *both* the smallest and largest obtained widths: one of the bounds is determined by the other, and can become arbitrarily small/large. Xiong et al. [2019] similarly produce varying width beads by contouring a distance field

computed from the input shape outline. The bounds also cannot be guaranteed, limiting the applicability to specific parts.

Jin et al. [2017a] reduce gaps and overlaps in narrow areas. After each successive parallel offset, a medial axis transform of the region sandwiched between trajectories is constructed. The radii along the medial axis are used to detect whether the region becomes smaller or larger than the desired deposition width. If it is too large an additional segment is inserted along the medial axis to avoid a gap. If it is too small, the trajectories are locally removed to avoid overflow. Compared to the methods presented above this method reduces the bead width variation to a more narrow range independent of the filled geometry. A similar approach is proposed in a concurrent work [Kuipers et al. 2020]. The resulting beads also have a preferred width in most locations, but a deviating width in the center. However, that deviation is distributed over several beads near the center of the shape, which further reduces the bead width variation. Moreover, the medial axis transform is only computed once, which results in a considerable computational performance boost. While both latter works offer clear improvements over the direct contouring approach, many uncovered areas remain.

We provide discussion and comparisons with respect to these approaches in Section 6.3.

3 KEY IDEAS

In this section, we give an overview of our approach and its underlying principles. Our algorithm for modeling beads has a standard structure (Figure 2): starting from a shape $\mathcal{S}^0$, we iteratively construct a sequence of nested shapes $\mathcal{S}^n \subset \cdots \subset \mathcal{S}^0$ so that the difference $\mathcal{S}^i \setminus \mathcal{S}^{i+1}$ between two successive shapes is, roughly, a set of topological annuli (Figures 1 (f) and 3 (h)). A bead is then modeled inside each annulus.

Common state-of-the-art approaches use beads of constant width, which, as mentioned in the introduction, leave many gaps (see Figure 2 left and middle). A key element of our approach for reducing the underfill is to use beads of variable width, say within 2γ and 2Γ.

Our idea is to determine beads with variable width so that the nested shapes $\mathcal{S}^0 \supset \cdots \supset \mathcal{S}^n$ become as "round" as possible. Somewhat counter-intuitively, our technique does so by *increasing* the bead width along sharp features, where the curvature of the boundary $\partial \mathcal{S}^i$ of $\mathcal{S}^i$ is locally maximal. Doing so aims at obtaining smaller local maxima of curvature along $\partial \mathcal{S}^{i+1}$, so that $\partial \mathcal{S}^{i+1}$ becomes as circular as possible.

Our approach for computing such sequences of nested shapes and the corresponding beads strongly relies on their representations as unions of disks using their medial axes. Recall that the medial axis of a shape is the (closure of the) set of points that are centers of so-called medial-axis disks tangent to the shape's boundary in at least two points. To each point on the medial axis corresponds the radius of this disk (see Section 4.2 for more formal definitions). We use the term *sub-axis* to denote a subset of the medial axis. The shape is equal to the union of its medial-axis disks. The nested shapes and the beads are constructed one from another by manipulating their medial axes (and their disk-radius functions) in three steps referred to as *Trimming, Collapsing* and *Shaving*. The print trajectories are defined as the medial axis of each bead, which is sampled and output.
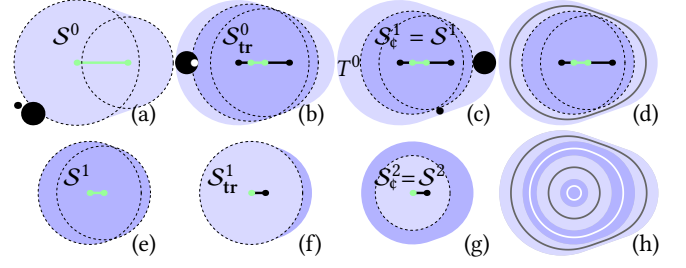

Fig. 3. A simple example. The diameters of the black disks illustrate the minimal and maximal bead widths. The medial axis is black where *Trimmed* and green otherwise (see Table 1). To help the reader, the medial-axis disks centered at *Normal* vertices (green) are stroked with dashes. *(a)* $\mathcal{S}^0$ and its medial axis. *(b)* Trimming. *(c)* Parallel offset of 2γ. The bead T^0 (light blue) is born. *(d)* The print trajectory traj(T^0) of the bead T^0. *(e)* $\mathcal{S}^1$ and its medial axis. *(f)* Trimming; a single point remains *Normal*. *(g)* Parallel offset larger than 2γ. *(h)* All four beads and print trajectories. There is no gap.
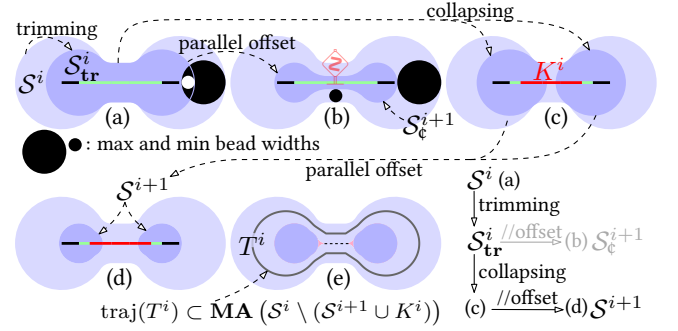

Fig. 4. The sequence of operations from $\mathcal{S}^i$ to $\mathcal{S}^{i+1}$. *(a)* Trimming $\mathcal{S}^i$. *(b)* A parallel offset of 2γ renders the dark blue shape $\mathcal{S}^{i+1}_{\mathfrak{C}}$ less than 2γ-fat, which violates our required invariant. *(c)* Instead, we identify the narrow features of $\mathcal{S}^i$ by the set K^i of centers of medial-axis disks of radius smaller than 4γ. *(d)* $\mathcal{S}^{i+1}$ is obtained by removing K^i from the medial axis of $\mathcal{S}^i_{tr}$ and applying a 2γ-parallel offset. *(e)* The bead T^i is modeled between $\partial \mathcal{S}^i$ and $\partial \mathcal{S}^{i+1} \cup K^i$. Its print trajectory is drawn in black. Gaps (pink pseudo-triangles) may remain between the bead T^i and $\mathcal{S}^{i+1}$.

Trimming. Trimming is applied first. It amounts to deleting parts of the medial axis of $\mathcal{S}^i$. The union of the medial-axis disks centered on the *remaining* medial axis is a new shape $\mathcal{S}^i_{tr}$, which is a strict subset of $\mathcal{S}^i$ (dark blue in Figures 3 (b) and 4 (a)). We trim as much as possible the extremities of the medial axis of $\mathcal{S}^i$ while ensuring that the *crescent shapes in* $\mathcal{S}^i \setminus \mathcal{S}^i_{tr}$ (light blue in Figures 3 (b) and 4 (a)) stay "narrow enough" for the future bead to not exceed 2Γ in width. (Roughly speaking, the crescent width, orthogonally to $\partial \mathcal{S}^i$, has to stay below $2\Gamma - 2\gamma$.)

We define $\mathcal{S}^{i+1}_{\mathfrak{C}}$ as the offset by 2γ of $\mathcal{S}^i_{tr}$ (Figures 3 (c) and 4 (b)). The shape $\mathcal{S}^{i+1}_{\mathfrak{C}}$ may not yet be a satisfactory candidate for $\mathcal{S}^{i+1}$ because it may contain regions that are too narrow for the smallest allowed deposition radius; this may for instance happen with a dog-bone ●—● shape as input (Figure 4 (b)).

Collapsing. To avoid these issues, we identify the narrow features of $\mathcal{S}^i$ as the subset K^i of its medial axis where the medial-axis disk

radius is smaller than 4γ (drawn in red in Figure 4 (c,d); recall that the medial axis of $\mathcal{S}^{i+1}_\varphi$ is included in that of $\mathcal{S}^i$). The narrow features of $\mathcal{S}^i$ are then *collapsed* onto K^i: The new shape $\mathcal{S}^{i+1}$ is defined as the parallel offset (of width 2γ) of the union of the disks centered on the medial axis of $\mathcal{S}^i_{\text{tr}}$ from which we remove the *Collapsed* sub-axis K^i (Figure 4 (d)) and the bead is locally widened in $\mathcal{S}^i \setminus (\mathcal{S}^{i+1} \cup K^i)$ to be in contact with $\partial\mathcal{S}^i$, $\partial\mathcal{S}^{i+1}$ and K^i (Figure 4 (e)).

Constraints on γ, Γ and the shapes. Our approach computes a sequence of nested shapes $\mathcal{S}^i$ such that each $\mathcal{S}^i \setminus \mathcal{S}^{i+1}$ is covered (possibly with gaps) by closed beads whose width varies between 2γ and 2Γ. To avoid gaps between $\partial\mathcal{S}^i$ and the bead(s) in $\mathcal{S}^i \setminus \mathcal{S}^{i+1}$, the shape $\mathcal{S}^i$ needs to be 2γ-fat (i.e., all medial-axis disks have radius at least 2γ). This is a constraint on $\mathcal{S}^0$, but the collapsing phases ensure that all other $\mathcal{S}^i$ are 2γ-fat. On the other hand, the collapsing phase defines, around *Collapsed* sub-axes, beads of width up to 4γ; since the largest possible width is 2Γ, we must have $2\Gamma \geq 4\gamma$. In summary, our technique has the following constraints: the shape $\mathcal{S}^0$ must be 2γ-fat and the bead width range must satisfy $\Gamma \geq 2\gamma$.

Bead trajectories. When K^i is empty (there is no narrow feature), $\mathcal{S}^i \setminus \mathcal{S}^{i+1}$ can be exactly covered, with no overfill nor underfill, by a closed bead $T^i = \mathcal{S}^i \setminus \mathcal{S}^{i+1}$ whose print trajectory is its medial axis (Figure 3). However, the presence of a *Collapsed* sub-axis calls for a more subtle definition of the beads wherein gaps (underfill) are unavoidable (see the small pink gaps in Figure 4 (e)). Details are given in Section 5.2 and Appendix A. We provide two optional strategies (Section 5.3) to minimize the area of these gaps. The first, *shaving*, "trims" the *Collapsed* sub-axis to widen the bead that wraps around it. The second, *medial axis simplification*, erases most of the problematic branches of K^i altogether, but at the expense of simplifying (thereby approximating) the shape $\mathcal{S}^i$.

Figure 4 bottom-left summarizes how the sequence $(\mathcal{S}^i)$ and the beads are computed. Figures 1, 3 and 17 show other examples of our technique in action.

4 PRELIMINARIES

We first discuss and emphasize our assumptions on our input shapes and on the fabrication constraints (Section 4.1). We then recall classical notions on medial axes (Section 4.2) and describe how we represent our nested shapes using sequences of medial axis transforms (Section 4.3). We then discuss our data structure, notation, and how we process our medial axes by labeling their arcs (Section 4.4). Some important definitions and notation are also illustrated in Figure 5 and summarized in Table 1.

4.1 Fabrication constraints and admissible input shapes

We assume that a given fabrication machine provides a range of achievable bead widths. For example, on a Ultimaker 3, we are able to vary the bead width within [0.3 mm, 1 mm] at 0.2 mm layer height, and [0.3 mm, 0.7 mm] at 0.1 mm layer height.

Since the achievable bead width range is hardware-dependent, we give it as input parameters $[2\gamma, 2\Gamma]$ to our technique. Our algorithm guarantees that all output beads have their width varying in that range. As a consequence of the way we define *closed* print trajectories, our initial shape $\mathcal{S}^0$ needs to be 2γ-fat. Furthermore,
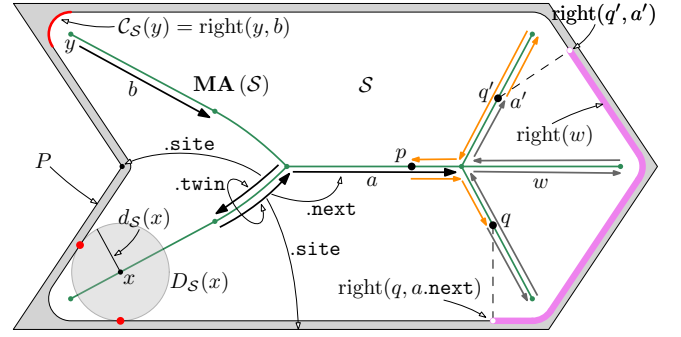

Fig. 5. Some notation. The wide black arrows along the medial axis are half-arcs. The bottom-left red dots are the contact points $C_\mathcal{S}(x)$ of x. The gray half-arcs form the walk w from point q on a.next to point q' on a'. The right-curve of w, right(w), is drawn in violet. Together, the gray and orange half-arcs form the w-tree $\mathcal{T}_{P,a}$ (defined in Section 5).

the way we define beads around *Collapsed* sub-axes (Section 5.2), requires that $\Gamma > 2\gamma$.

In practice, our implementation naturally considers as input a polygon P. We thus start by computing the largest 2γ-fat shape $\mathcal{S}^0$ in P. This is easily done by computing the medial axis of P and by deleting all the sub-arcs supporting medial-axis disks of radius less than 2γ. (As discussed below, $\mathcal{S}^i$ will be represented by its medial axis and the corresponding medial-axis radius function.)

Note that the shape $\mathcal{S}^0$ that we will print is 2γ-fat although the fabrication constraints allow to print shapes that are γ-fat inside P, instead of 2γ-fat. This is a slight limitation of our algorithm. The thin features of P, i.e., $P \setminus \mathcal{S}^0$, might be processed with a specialized path planner, which we leave outside the scope of the present work (see e.g. [Jin et al. 2017a]).
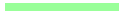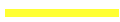
4.2 Medial axis

Let $\mathcal{S}$ be a planar shape. We denote its boundary as $\partial\mathcal{S}$. We write $D(x, r)$ for the closed disk centered at a point x with radius r. A disk centered at $x \in \mathcal{S}$ and contained in $\mathcal{S}$ is *maximal* if any larger disk centered at x has a point *not* in $\mathcal{S}$. There is a unique maximal disk centered at x. We call it $D_\mathcal{S}(x)$ and its radius is denoted $d_\mathcal{S}(x)$. We define the *contact points* $C_\mathcal{S}(x)$ *of x* as the points of $\partial D_\mathcal{S}(x) \cap \partial\mathcal{S}$. A maximal disk $D_\mathcal{S}(x)$ is also a *medial-axis* disk when x has at least two contact points. The medial axis of $\mathcal{S}$, written $\mathbf{MA}(\mathcal{S})$ is the closure of the set $\{x \in \mathcal{S} \mid |C_\mathcal{S}(x)| > 1\}$ of centers of the medial-axis disks. The medial axis $\mathbf{MA}(\mathcal{S})$ is a network of curves that we call *arcs*, connected at *vertices*.

The main object that we work with is the *medial axis transform*, or MAT, of $\mathcal{S}$ [Blum 1967]. It comprises (1) an explicit representation of the medial axis $\mathbf{MA}(\mathcal{S})$ of $\mathcal{S}$ together with (2) an explicit representation of the maximal radius function $d_\mathcal{S}$ restricted to $\mathbf{MA}(\mathcal{S})$.

The medial axis transform (MAT) of a shape is enough to recover it: Given a network of curves $\mathcal{M}$ and a non-negative function d defined over $\mathcal{M}$, we define the shape *generated* by $(\mathcal{M}, d)$ as $\mathcal{G}(\mathcal{M}, d) = \cup_{x \in \mathcal{M}} D(x, d(x))$. Of course, the MAT of $\mathcal{S}$ satisfies $\mathcal{S} = \mathcal{G}(\mathbf{MA}(\mathcal{S}), d_\mathcal{S})$. In this work, we do not store the boundary $\partial\mathcal{S}$ of $\mathcal{S}$. Instead, we store and modify its medial axis transform

Table 1. Notation and color code for the medial axis arcs.

$\mathcal{S}$	A round shape
$\partial\mathcal{S}$	The boundary of shape $\mathcal{S}$
$D(x, r)$	Closed disk centered at x, of radius r
$D_\mathcal{S}(x)$	Maximal disk centered at x included in $\mathcal{S}$
$d_\mathcal{S}(x)$	Radius of $D_\mathcal{S}(x)$; distance from x to $\partial\mathcal{S}$
$C_\mathcal{S}(x)$	The contact points: $C_\mathcal{S}(x) = \partial D_\mathcal{S}(x) \cap \partial\mathcal{S}$
$\mathbf{MA}(\mathcal{S})$	The medial axis
half-arc a	Stores a.site, a.twin, a.next and the label of a
right(x, a)	The contact point of $C_\mathcal{S}(x)$ to the right of a
right(w)	The right-curve of w: the sub-curve of $\partial\mathcal{S}$ to the right of the walk w
![green line]	*Normal* label (green)
![black line]	*Trimmed* label (black)
![red line]	*Collapsed* label (red)
![yellow line]	*Shaved* label (yellow)

explicitly. When needed, we walk around the medial axis in order to reconstruct the boundary of $\mathcal{S}$ (for display or sampling).

Throughout the paper, we assume that a shape $\mathcal{S}$ can have multiple connected components and holes, so that its boundary $\partial\mathcal{S}$ consists in a number of closed curves. We compute all medial axes using the CGAL library [Karavelas 2019].

4.3 Sequences of medial axis transforms

Starting from the MAT of an input polygonal shape P (possibly with holes and several connected components), our technique constructs a sequence of MATs wherein each shape $\mathcal{S}'$ is obtained from the previous one $\mathcal{S}$ by

(1) removing some arcs or sub-arcs from $\mathbf{MA}(\mathcal{S})$, which defines $\mathbf{MA}(\mathcal{S}')$, and

(2) *parallel offsetting*, i.e., subtracting a constant $r \geq 0$ from $d_\mathcal{S}$, provided that $d_\mathcal{S}(x) > r$ for all $x \in \mathbf{MA}(\mathcal{S}')$, which defines $d_{\mathcal{S}'}$.

Let $\mathcal{S}$ be a shape obtained from P by applying any sequence of the above transformation to P. We call $\mathcal{S}$ a *round shape*. From now on, we discuss only round shapes. The boundaries of a round shape are sequences of line segments and circular arcs. The arcs of $\mathbf{MA}(\mathcal{S})$ are the locus of the centers of the medial-axis disks in P, tangent to two edges of P, or an edge and a vertex of P, or two vertices. These arcs are then always linear or parabolic segments, and we call their type, respectively, EE, EV and VV. The value of $d_\mathcal{S}$ is equal to the value of d_P minus the *current offset* of $\mathcal{S}$, which we write offset$(\mathcal{S})$ and is the sum of all parallel offsets that have been applied from P to $\mathcal{S}$. The current offset is constant per connected component of $\mathcal{S}$. Each connected component is treated independently, so we can assume that $\mathcal{S}$ is connected and write offset$(\mathcal{S})$ for its current offset.

4.4 Data structure and notation

Let P be the input polygon and $\mathcal{S}^i$ be a round shapes obtained from P as described above (Section 4.3). For the medial axes of P and $\mathcal{S}^i$, we use a "half-arcs" data structure similar to the half-edge data structure [de Berg et al. 2008]. A half-arc a is an arc with a prescribed orientation that lies on the bisector between two sites s

and s' (vertex or edge) of polygon P. Let $\ell(a)$ be the oriented line or parabola that supports the half-arc a with the same orientation.

The half-arc a stores a reference to its "site to the right," a.site, which is the feature of $\{s, s'\}$ that, for any point x in the relative interior $\mathring{a}$ of a, supports the unique point of $C_P(x)$ that lies to the right of $\ell(a)$. Note that a.site is a feature of P, even for the half-arcs a of $\mathbf{MA}(\mathcal{S}^i)$. Each half-arc a also stores references to a.twin, the half-arc with the same geometry but opposite orientation, and to a.next, the next half-arc around the medial axis: If half-arc a ends at vertex p_1, then a.next is the half-arc that starts *from* p_1 and is geometrically immediately after a in the counter-clockwise radial ordering around p_1.

Given a point x in the relative interior $\mathring{a}$ of half-arc a of $\mathbf{MA}(\mathcal{S}^i)$, we call the *right contact* of x, written right(x, a), the point of $C_{\mathcal{S}^i}(x)$ that lies to the right of $\ell(a)$. When x is the common vertex of a and a.next, $C_{\mathcal{S}^i}(x)$ may have several connected components, each of which being a point or a convex circular arc. If $C_{\mathcal{S}^i}(x)$ is connected, we set right$(x, a$.next$) = C_{\mathcal{S}^i}(x)$ and, in general, we define right$(x, a$.next$)$ as the connected component of $C_{\mathcal{S}^i}(x)$ that connects $\cup_{y \in \mathring{a}}$right$(y, a)$ and $\cup_{y \in a.\mathring{\text{next}}}$right$(y, a$.next$)$. We also set right$(x, a) = \emptyset$ for conveniency.

We define a *walk along* $\mathbf{MA}(\mathcal{S}^i)$ as a sequence of half-arcs $w = a_0, a_1, \ldots, a_n$ such that $a_{i+1} = a_i$.next. We extend the notion to allow the start and end points of a walk to lie inside an half-arc. We then talk about a walk from $p \in a_0$ to $q \in a_n$. A walk w on $\mathbf{MA}(\mathcal{S}^i)$ defines its *right curve* on $\partial\mathcal{S}^i$ as right$(w) = \cup_{(x,a) \in w}$ right(x, a).

Working with labels. Our technique makes essential use of a labeling of a subdivision of the medial axis. At the beginning of each iteration, $\mathbf{MA}(\mathcal{S}^i)$ is labeled *Normal* everywhere. After trimming, collapsing and shaving, $\mathbf{MA}(\mathcal{S}^i)$ will have been subdivided into sub-axes with four different labels: *Normal*, *Trimmed*, *Collapsed* and *Shaved* (See Table 1). The labeling is used both to guide the sampling of the print trajectory traj(T) of each bead T and to define the next shape $\mathcal{S}^{i+1}$ as the shape generated by the subset of $\mathbf{MA}(\mathcal{S}^i)$ labeled *Normal*. We write *Normal*$(\mathbf{MA}(\mathcal{S}))$ (resp. *Collapsed*$(\mathbf{MA}(\mathcal{S}))$) for the subset of $\mathbf{MA}(\mathcal{S})$ labeled *Normal* (resp. *Collapsed*).

5 VARIABLE-WIDTH CONTOURING

Given a round shape $\mathcal{S}^i$, we start by applying trimming to its medial axis. Trimming labels as *Trimmed* the extremities of $\mathbf{MA}(\mathcal{S}^i)$ as described in Section 5.1. After trimming, collapsing (Section 5.2) labels as *Collapsed* the sub-axes of $\mathbf{MA}(\mathcal{S}^i)$ that are too narrow (roughly, where $d_{\mathcal{S}^i} < 4\gamma$). After collapsing, the axis $\mathbf{MA}(\mathcal{S}^i)$ will have been subdivided into *Collapsed*, *Trimmed* and *Normal* sub-axes. See, e.g. Figure 17 (4). The MAT of $\mathcal{S}^{i+1}$ is then obtained as follows:

$$\mathbf{MA}\left(\mathcal{S}^{i+1}\right) = Normal\left(\mathbf{MA}\left(\mathcal{S}^i\right)\right) \tag{1}$$

$$d_{\mathcal{S}^{i+1}} = d_{\mathcal{S}^i} - 2\gamma \tag{2}$$

$$\mathcal{S}^{i+1} = \mathcal{G}\left(\mathbf{MA}\left(\mathcal{S}^{i+1}\right), d_{\mathcal{S}^{i+1}}\right). \tag{3}$$

Each bead is modeled between the boundaries of two successive shapes $\mathcal{S}^i$ and $\mathcal{S}^{i+1}$. We denote this difference by $\widetilde{T}^i = \mathcal{S}^i \setminus \mathcal{S}^{i+1}$ and the bead by $T^i \subset \widetilde{T}^i$. If trimming produces a 4γ-fat shape $\mathcal{S}^i_{\mathbf{tr}}$ (i.e., $\mathcal{S}^{i+1}_{\text{⊄}}$ is 2γ-fat and $\mathcal{S}^{i+1} = \mathcal{S}^{i+1}_{\text{⊄}}$), then collapsing has no

effect and $\widetilde{T}^i$ can be fully covered by bead loops with width varying within the prescribed bounds. In that case, it is easy to sample the print trajectory of the bead, $\text{traj}(T^i)$ (see e.g. Figure 3 (h)). This is detailed in Appendix A. Otherwise, the *Collapsed* sub-axis of $\mathcal{S}^i$ is not empty and the bead cannot fully cover the area $\widetilde{T}^i$ and leaves gaps: $\widetilde{T}^i \setminus T^i \neq \emptyset$ (Section 5.2). In a way, trimming tries to minimize the occurrence of collapsing. When a collapse occurs anyway, we strive to minimize the area of the gaps with *shaving* (Section 5.3.1) and *medial axis simplification* (Section 5.3.2).

In the next sections, we describe trimming, collapsing, shaving and medial axis simplification. The sampling of the print trajectories of the beads is detailed in Appendix A. The proof that the beads generated with variable-width contouring are free of overfill is presented in Appendix B.

5.1 Trimming

In the trimming phase, we cut the branches of $\mathbf{MA}\left(\mathcal{S}^i\right)$ as much as possible so as to make the shape "rounder." This is a very effective strategy: often, after some iterations the remaining shape becomes exactly a disk, after which gap-free bead generation becomes trivial (Figure 3 (g-h)).

We introduce trimming with a simple example (Figure 3), then generalize trimming to arbitrary input. In (a) we see the input shape $\mathcal{S}^0$ which, for the sake of this simple example, is r-fat for some r larger than 2Γ. $\mathbf{MA}\left(\mathcal{S}^i\right)$ reduces to a single line segment initially labeled *Normal* (colored green). Trimming determines a splitting point on the medial axis at both ends and labels the two extreme segments as *Trimmed* (colored black). From (b) to (e), the dark-blue shape is generated by the remaining *Normal* sub-axis. The splitting points (the endpoints of the green, or *Normal*, segment in (b)) are computed so that the maximal width of the crescent shapes (light blue on either side) is $2\Gamma - 2\gamma$. The subtraction of 2γ is in anticipation of the parallel offset of 2γ that happens in (c). There is no thin feature so collapsing does not apply to this example. The parallel offset ensures that the bead (the light-blue shape in (c)) has width larger than 2γ and the bound on the crescents' widths ensures that the bead is no wider than 2Γ. The bead $T^0 = \mathcal{S}^1 \setminus \mathcal{S}^0$ is complete and (d) shows its print trajectory as a dark line. In (e), we see the shape $\mathcal{S}^1$ from which we start the computation of the next bead. The medial axis is very short now and trimming is able to label it entirely *Trimmed*, but for a single point which stays *Normal* as needed to provide an inner boundary for the bead T^1. In (f) we see the result of trimming. Since $\mathbf{MA}\left(\mathcal{S}^2\right)$ is a point, all subsequent beads are annuli (g-h).

Before describing the algorithm for trimming, we give a detailed geometric description of the general case, illustrated in Figure 6. The trees that we consider for trimming are of a particular kind. We call them *walk-trees* or *w-trees* for short. A *w-tree* $\mathcal{T} \subset \mathbf{MA}\left(\mathcal{S}^i\right)$ is obtained by choosing a *root* point p on $\mathbf{MA}\left(\mathcal{S}^i\right)$ and one half-arc a that contains p. The *w-tree* $\mathcal{T}_{p,a}$ consists of all the points of $\mathbf{MA}\left(\mathcal{S}^i\right)$ on the walk from $p \in a$ to $p \in a.\texttt{twin}$ (see Figure 5), if, of course, that *combinatorial* walk exists and its corresponding subset of $\mathbf{MA}\left(\mathcal{S}^i\right)$ in the plane does not contain any loop. Note that this forbids to choose the half-arc a on a cycle of $\mathbf{MA}\left(\mathcal{S}^i\right)$ since in that case, $a.\texttt{twin}$ is not reachable from a. The *w-trees* have two important
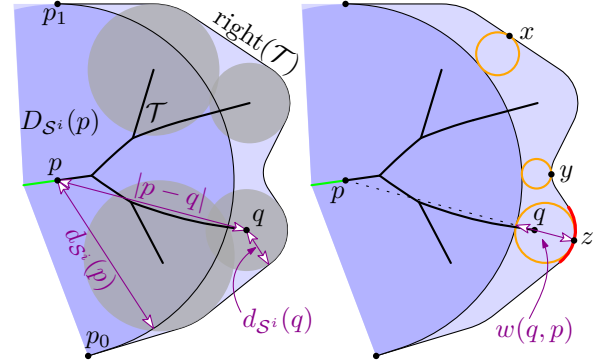

Fig. 6. To avoid clutter, the diagram is drawn twice. *Left.* The crescent $\mathbb{D}$ is delineated with a thin black curve. The medial axis is trimmed at p. The medial-axis disks at the leaves of the axis are drawn gray. *Right.* The width of the crescent at x, y and z are the diameter of the respective orange circles. The width at z is a local maximum.

properties: First the requirement for a *w-tree* to be a walk from $p \in a$ to $p \in a.\texttt{twin}$ enforces the property that if one cuts the medial axis at p, then $\mathcal{T}_{p,a}$ is a simply connected component, detached from the rest. Second and more importantly, the right-curve $\text{right}(\mathcal{T})$ of a *w-tree* $\mathcal{T}$ (see Section 4.4) is a single connected subset of $\partial \mathcal{S}^i$, whereas the right-curve of a random tree in $\mathbf{MA}\left(\mathcal{S}^i\right)$ may consist of several disconnected curves along $\partial \mathcal{S}^i$.

Let $\mathcal{T}$ be a *w-tree* and p its root: $p = \text{root}(\mathcal{T})$. The right-curve of $\mathcal{T}$, $\text{right}(\mathcal{T})$, is a (connected) sub-curve of $\partial \mathcal{S}^i$ whose endpoints, p_0 and p_1 are contact points of the root p: $\{p0, p1\} \subset C_{\mathcal{S}^i}(p)$. Labeling $\mathcal{T}$ as *Trimmed* has the effect of replacing the sub-curve $\text{right}(\mathcal{T})$ of $\partial \mathcal{S}^i$ by the circular arc on $\partial D_{\mathcal{S}^i}(p)$ from p_0 to p_1; See Figure 6.

Together, the curve $\text{right}(\mathcal{T})$ and that circular arc bound a "crescent" $\mathbb{D}$ that forms part of the bead T^i. (Figure 3 (b-c) shows how crescents are integrated into the bead.) To make sure the bead is printable, we need to control its maximal width: At a point $x \in \text{right}(\mathcal{T})$, the width of the crescent $\mathbb{D}$ is the diameter of the largest disk tangent to $\text{right}(\mathcal{T})$ at x and to $D_{\mathcal{S}^i}(p)$ (orange color in Figure 6). (The constraints that we enforce on the *w-trees* that we work with (see below) ensure that the width of the crescent is always well defined.) That width may have a local minimum anywhere, along line segments (e.g. in the neighborhood of x in Figure 6) or convex or concave circular arcs (e.g. at y in Figure 6). But it can have a local maximum only along *convex circular arcs* (e.g. at z in Figure 6).

5.1.1 Constraints from the convex circular arcs.
The convex circular arcs along $\text{right}(\mathcal{T})$ are generated by the vertices v of $\mathcal{T}$ for which $\partial D_{\mathcal{S}^i}(v) \cap \text{right}(\mathcal{T})$ is a circular arc. These vertices include the leaves of $\mathcal{T}$ and some internal vertices. We need to focus only on these vertices, which we gather in the set $\text{conv}(\mathcal{T})$.

Let q be any vertex in $\text{conv}(\mathcal{T})$. By definition, vertex q generates a convex circular arc on $\text{right}(\mathcal{T}) \subset \partial \mathcal{S}^i$ (see the red circular arc in Figure 6). The maximum width of $\mathbb{D}$ along that arc is denoted $w(q, p)$ and we have $w(q, p) = |p - q| + d_{\mathcal{S}^i}(q) - d_{\mathcal{S}^i}(p)$. The width $w(q, p)$ is the diameter of the disk (orange color) tangent to $\text{right}(\mathcal{T})$ at the intersection point z of line pq with the circular arc centered at q (red color). This orange disk is tangent to $\text{right}(\mathcal{T})$ at z and
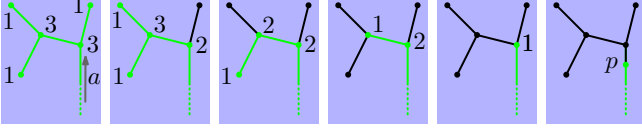
Fig. 7. Maximal trimmable w-trees are built from the leaves to the root. From left to right, each step processes one half-arc and a new set of a trimmable w-trees (black) is obtained. At the last step (right), a *maximal* trimmable w-tree $\mathcal{T}_{p,a}$ is found. The number next to each vertex indicates its *Normal*-degree.

must be internally tangent, so that its diameter $w(q, p)$ cannot be larger than the diameter of the osculating disk as z, namely $2d_{\mathcal{S}^i}(q)$. In order to guarantee that the bead width stays within the fabrication constraints, we put further constraints on $w(q, p)$: taking into account the subsequent parallel offset of 2γ and the upper bound 2Γ on any bead's width, we conclude that the root p of $\mathcal{T}$ must respect $w(q, p) \leq \min(2d_{\mathcal{S}^i}(q), 2\Gamma) - 2\gamma$ for all vertices $q \in \mathrm{conv}(\mathcal{T})$. At a vertex q in $\mathrm{conv}(\mathcal{T})$ we define $W(q) = 2(\min(d_{\mathcal{S}^i}(q), \Gamma) - \gamma)$, the *maximal bead width allowed at q*. We say that *the* w*-tree $\mathcal{T}$ is trimmable* if

$$\forall q \in \mathrm{conv}(\mathcal{T}), w(q, \mathrm{root}(\mathcal{T})) \leq W(q) \qquad (4)$$

5.1.2 Algorithm for trimming. Trimming computes all *maximal* trimmable w-trees and labels them as *Trimmed*. When we extend the w-tree $\mathcal{T}$ by moving its root p away from its leaves, the value $w(q, p)$ increases for each vertex $q \in \mathrm{conv}(\mathcal{T})$.[*] This implies that if a w-tree $\mathcal{T}$ is trimmable, then any sub-w-tree $\mathcal{T}'$ of $\mathcal{T}$ ($\mathcal{T}' \subset \mathcal{T}$) is trimmable too, so that the notion of a maximal (by inclusion) trimmable w-tree is well-defined. To compute all maximal trimmable w-trees we start from the leaves of $\mathbf{MA}(\mathcal{S}^i)$, which are the *smallest* trimmable w-trees, and grow them while labeling the corresponding parts of the medial axis as *Trimmed*, until $w(q, \mathrm{root}(\mathcal{T})) = W(q)$ for some $q \in \mathrm{conv}(\mathcal{T})$ or the growth is stopped by other constraints, as we detail below.

While growing the trimmable w-trees, we maintain the invariant that each vertex v of $\mathbf{MA}(\mathcal{S}^i)$ with *Normal*-degree 1 [†] is the root of a trimmable (and *Trimmed*) w-tree. That invariant is of course true at the very beginning since we have seen that each leaf of $\mathbf{MA}(\mathcal{S}^i)$ is a (small!) trimmable w-tree. See Figure 7.

For each *Normal*-degree 1 vertex p of $\mathbf{MA}(\mathcal{S}^i)$ let $e = p-p'$ be the *Normal* half-arc incident to p. Let $\mathcal{T}_p = \mathcal{T}_{p,e.\mathtt{twin}}$ be the trimmable w-tree rooted at p. $\mathcal{T}_p$ is just a point when p is a leaf of $\mathbf{MA}(\mathcal{S}^i)$ and a larger w-tree when p is the root of a non-maximal trimmable w-tree being grown. We consider the half-arc e, a point $u \in e$ and the w-tree $\mathcal{T}_{u,e.\mathtt{twin}}$ (the walk from $u \in e.\mathtt{twin}$ to $u \in e$). We have $\mathcal{T}_{p,e.\mathtt{twin}} \subset \mathcal{T}_{u,e.\mathtt{twin}}$.

(1) If e has already been labeled as *Trimmed*, then stop. If there exists a point u in half-arc e that makes $\mathcal{T}_{u,e.\mathtt{twin}}$ a maximal trimmable w-tree (see Appendix C) then we split the arc e at u and label the sub-arc $p-u$ as *Trimmed*.

---

[*] Indeed, while p moves along its arc to grow $\mathcal{T}$, the crescent area strictly increases for the subset relation $\subset$. The orange disks in Figure 6 must grow as well.

[†] The *Normal*-degree of vertex v is the number of *Normal* arcs incident to v.

(2) Otherwise, by definition, $\mathcal{T}_{p',e.\mathtt{twin}}$ is trimmable as well: we mark e as *Trimmed* and examine the *Normal*-degree of p'.
   (a) If it is larger than 1, we stop here, since growing the w-tree along any incident *Normal* arc would not yield a w-tree (i.e., a walk from a half-arc to its twin). Later, trimming from other leaves might reach p', making its *Normal*-degree 1, and continue the trimming along the only *Normal* half-arc incident to p'.
   (b) If it is 1, then all arcs incident to p' but one, say e', have been trimmed, so we propagate trimming along e', back to step (1) above.

The computation of $u \in e$ at step (1) amounts to solving a quadratic polynomial for each vertex in $\mathrm{conv}(\mathcal{T})$. It is detailed in Appendix C.

The set $\mathrm{conv}(\mathcal{T})$ does not depend on the point in e at which $\mathcal{T}$ is rooted. We compute it efficiently from the leaves upward: We give to each vertex v of $\mathbf{MA}(\mathcal{S}^i)$ an attribute "$v.\mathrm{bag}$" that is a dynamic set of references to other vertices of $\mathbf{MA}(\mathcal{S}^i)$. At the beginning of trimming, we set $v.\mathrm{bag} \leftarrow \{v\}$ if v generates a convex circular arc on $\partial \mathcal{S}^i$, and $v.\mathrm{bag} \leftarrow \emptyset$ otherwise. When processing p, $p.\mathrm{bag}$ contains the union of $\mathrm{conv}(\mathcal{T})$ for all the trimmable w-trees $\mathcal{T}$ rooted at p. At the beginning of step (2), we set $p'.\mathrm{bag} \leftarrow p'.\mathrm{bag} \cup p.\mathrm{bag}$.

Non-determinism. There is one source of non-determinism in the above algorithm. When a connected component of $\mathbf{MA}(\mathcal{S}^i)$ does not contain any loop, it is possible that the maximal trimmable w-trees in that component have a non-empty intersection $\mathcal{I}$. In that case, any point k in the intersection $\mathcal{I}$ is an acceptable cut point for trimming. The whole medial axis is trimmed away except for that point $k \in \mathcal{I}$, whose label remains *Normal*. Our algorithm for trimming picks any leaf of $\mathcal{I}$, and the growth of the other trimmable w-tree is stopped at this vertex by checking the label of e in step (1). This is the only source of non-determinism in our technique. If desired, the intersection $\mathcal{I}$ could be computed and the cut-point chosen deterministically according to some heuristic.

5.2 Collapsing

Motivation for collapsing. After trimming, $\mathbf{MA}(\mathcal{S}^i)$ is labeled *Normal* everywhere except along the maximal trimmable w-trees that are labeled *Trimmed*. In order to understand why collapsing is necessary, let us look at the shape $\mathcal{S}_{\mathrm{\cent}}^{i+1}$ obtained after trimming and parallel offset, but forgoing collapse: $\mathcal{S}_{\mathrm{\cent}}^{i+1} = \mathcal{G}(Normal(\mathbf{MA}(\mathcal{S}^i)), d_{\mathcal{S}^i} - 2\gamma)$; see e.g. the dark blue shape in Figure 3 (c) or in Figure 4 (b). (The $\mathrm{\cent}$ symbol reminds the reader that collapsing is not applied when modeling this shape.)

The set $T = \mathcal{S}^i \setminus \mathcal{S}_{\mathrm{\cent}}^{i+1}$ is a printable bead. This comes from the way maximal trimmable w-trees "leave room" for a subsequent 2γ parallel offset, and the fact that $\mathcal{S}^i$ is 2γ-fat, so that the medial axis of T is a simple closed curve. However, $\mathcal{S}_{\mathrm{\cent}}^{i+1}$ may exhibit features too narrow for accommodating a bead of minimal width 2γ. If $\mathcal{S}_{\mathrm{\cent}}^{i+1}$ is 2γ-fat, then $\mathcal{S}^{i+1} = \mathcal{S}_{\mathrm{\cent}}^{i+1}$, $T^i = T$ and collapsing will have no effect. If $\mathcal{S}_{\mathrm{\cent}}^{i+1}$ is not 2γ-fat, we need to collapse the narrow features of $\mathcal{S}_{\mathrm{\cent}}^{i+1}$ on the medial axis to obtain a suitable shape $\mathcal{S}^{i+1}$ for the successful continuation of the process.

We reexamine the dog-bone example in more details: see Figure 4. In (a) we see a narrow shape such that $d_{\mathcal{S}^i}(p) < 4\gamma$ at some *Normal* points p around the center of the shape. In (b), the dark blue shape is
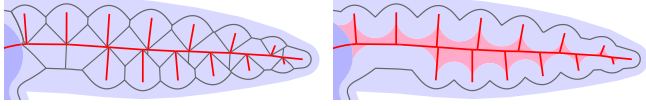
Fig. 8. *Left*. In red, the *Collapsed* sub-axis K^0. In dark gray, the medial axis of $\widetilde{T}^0 \setminus K^0$. *Right*. The print trajectory traj(T^0) comprises the cycles of $\mathbf{MA}\left(\widetilde{T}^0 \setminus K^0\right)$ without its branches. The bead T^0 is shaded light-blue. The gaps (pink) between T^0 and K^0 are very prominent in this example.

Fig. 9. The EE arc is drawn blue. *Left*. Case 1: $L \le \Gamma$. *Right*. Case 2: $L > \Gamma$.

$\mathcal{S}_\ell^{i+1}$. Its middle part is too narrow: $d_{\mathcal{S}_\ell^i}(p) < 2\gamma$ which is equivalent to $d_{\mathcal{S}^i}(p) < 4\gamma$. In (c), we see the result of the collapsing: the sub-axis K^i of $\mathbf{MA}\left(\mathcal{S}^i\right)$ where $d_{\mathcal{S}^i} < 4\gamma$ is marked as *Collapsed* (red color). The parallel offset is done only after collapsing (d). The idea of collapsing is to collapse these narrow parts of $\mathcal{S}_\ell^{i+1}$ onto the medial axis. To do so, the bead $T = \mathcal{S}^i \setminus \mathcal{S}_\ell^{i+1}$ around narrow passages is enlarged until it touches the medial axis in (e).

Collapsing is applied after trimming and before the parallel offset. Below, we describe collapsing in full generality.

5.2.1 Base collapse. First, the sub-axis of $\mathbf{MA}\left(\mathcal{S}^i\right)$ along which $d_{\mathcal{S}^i}(x) \le 4\gamma$ is identified. Then, only the connected components of this sub-axis that overlap with a *Normal* arc are labeled *Collapsed* (drawn in red in the figures). In other words, component that are already fully *Trimmed* are ignored, and a component labeled *Collapsed* might overwrite both *Normal* and *Trimmed* labels.

If a *Trimmed* maximal w-tree is partially *Collapsed*, then its root p is necessarily *Collapsed*. The disappearance of p from the *Normal* sub-axis results in the disappearance of the local inner boundary of the bead supported by $\partial D_{\mathcal{S}^i}(p)$ (see Figure 6). So, we relabel as *Normal* each vertex incident to both a *Trimmed* and a *Collapsed* arc. Such a *Normal* vertex v contributes a disk in $\mathcal{S}^{i+1}$ that is suitable for supporting the inner side of the crescent associated with the smaller (non-maximal) trimmable w-tree rooted at v.

We write K^i for the collapsed sub-axis of $\mathbf{MA}\left(\mathcal{S}^i\right)$: $K^i = \{x \in \mathbf{MA}\left(\mathcal{S}^i\right) \mid x$ is labeled *Collapsed*$\}$. To account for the *Collapsed* sub-axis, we generalize the modeling of the bead T^i as follows. We model T^i *indirectly* by defining its medial axis, which is also its print trajectory traj(T^i). We consider the medial axis $M = \mathbf{MA}\left(\mathcal{S}^i \setminus (\mathcal{S}^{i+1} \cup K^i)\right)$ and define the print trajectory for T^i as the cycles in M, i.e., we remove all the branches from M, see Figure 8. The width of the T^i is naturally defined by the radius function associated to the medial axis M.

This definition, however, possibly creates large regions in $\widetilde{T}^i$ that are not covered by the bead (Figures 4 and 8 right). In Section 5.3 we propose two strategies to reduce such underfill.

5.2.2 Collapsing more to help subsequent trimming. The base collapse operation, described above, leaves many *Normal* vertices in $\mathcal{S}^{i+1}$ with the minimal radius $d_{\mathcal{S}^{i+1}} = 2\gamma$. Each of these has a higher chance to lead to another collapse immediately or a few steps later if trimming cannot perform well. This happens when the radius at these points increases slowly. During collapsing however, we have the freedom to collapse further, up to a radius of 2Γ. In this section, we examine the case of an EE arc of the medial axis[‡] and give the

‡ Refer to Section 4.3 for the definition of EE, EV and VV.

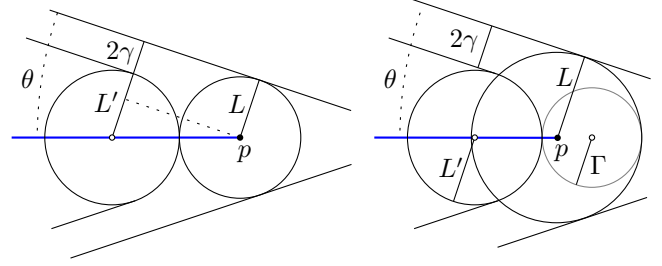conditions for trimming to actually increase the local radius of the shape. In turn, this gives us a heuristic way to extend the collapse just enough beyond 4γ to keep small gaps but guarantee that the subsequent trimming actually increases the local radius. The EE case is also a conservative approximation of the EV and VV cases, so that the heuristic works for these types of arc.

We consider an (infinitely long) EE arc e that starts at leaf p and generated by two line segments that form and angle of 2θ. See Figure 9. Then, along the arc e, the radius $d_{\mathcal{S}^i}$ has gradient $d'_{\mathcal{S}^i} = \pm \sin \theta$.

Case 1: If $L \equiv d_{\mathcal{S}^i}(p) \le \Gamma$ (Figure 9 left). Then, trimming finds a disk of radius L' centered on e and touching $D(p)$. We have $L' + 2\gamma = L + (L + L') \sin \theta$. Which implies that $L' > L \Leftrightarrow L \sin \theta > \gamma$.

Case 2: If $L \equiv d_{\mathcal{S}^i}(p) > \Gamma$ (Figure 9 right). The equation changes because $W(p) = \Gamma$. ($W(p)$ is the maximal bead width allowed at p, see Section 5.1.1.) We obtain $L' = L + \frac{2(\Gamma \sin \theta - \gamma)}{1 - \sin \theta}$ and $L' > L \Leftrightarrow \Gamma \sin \theta > \gamma$.

Let us now extend collapse beyond 4γ at point p. If $\Gamma \sin \theta \le \gamma$ then, whatever the value of $d_{\mathcal{S}^i}(p)$, trimming cannot increase the radius and collapse must reappear. In that case, we want to extend the collapsed axis as much as possible, that is, up to the point x on arc e where $d_{\mathcal{S}^i}(x) = 2\Gamma$. If $\Gamma \sin \theta > \gamma$ then we extend the collapse up to $x \in e$ where

$$d_{\mathcal{S}^i}(x) = \gamma/\sin \theta + 2\gamma. \tag{5}$$

(The addition of 2γ accounts for the subsequent parallel offset.)

We now transpose this idea to a general path along $\mathbf{MA}\left(\mathcal{S}^i\right)$. At any point p on an arc of $\mathbf{MA}\left(\mathcal{S}^i\right)$, the gradient of $d_{\mathcal{S}^i}$ along that arc is $\sin \theta$ where 2θ is the angle between the tangents to $\partial \mathcal{S}^i$ at the two contact points of $C_{\mathcal{S}^i}(p)$. When propagating the *Collapsed* label along an arc p–q, we first evaluate the gradient at p and see if there is a point x along the arc that satisfies Equation (5). If there is, we stop the collapse at x. Otherwise we propagate the *Collapsed* label beyond q, or at a point on the arc that reaches the maximal radius of 2Γ. That method is exact along EE arcs and conservative on other arcs along which the gradient of $d_{\mathcal{S}^i}$ always increases.

Because it is exact on EE arcs, trimming will actually keep the radius, at the next iterations, exactly the same as long as trimming stays on that same arc. To actually make the shape rounder along EE arcs, we add 5 % to the right member of Equation (5).

Finally, note that one could also directly solve $d_{\mathcal{S}^i}(x) = \frac{\gamma}{d'_{\mathcal{S}^i}(x)} + 2\gamma$ on EV and VV arc. This amounts to finding the roots of a cubic
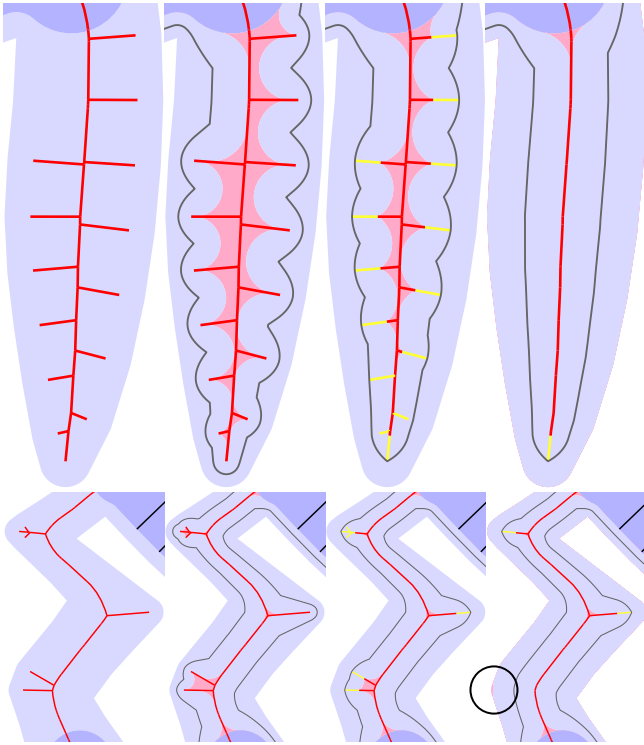
Fig. 10. Two parts of the funny figure of Figure 17. *From left to right:* The labeled $\mathbf{MA}\left(\mathcal{S}^0\right)$; Gaps without shaving nor medial axis simplification; With shaving (the yellow color indicates *Shaved* arcs); With both. The circle highlights gaps due to a local loss of shape fidelity.

polynomial for EV arcs and a quartic polynomial for VV arcs. (We have not implemented this alternative.)

5.3 Improving the bead shape around collapsed sub-axes

When the input is a polygonal approximation of a smooth shape, the medial axis contains many branches that come close to $\partial \mathcal{S}^0$. During the computation of the very first bead, these branches might appear in K^0 (Figure 10, first column). Their closeness to the boundary of $\mathcal{S}^0$ makes it difficult for the bead T^0 to completely fill $\widetilde{T}^0$. More generally, branches in K^i are responsible for large gaps not covered by T^i, inside each corner of K^i (Figure 10, second column). The problem exists mostly for the first bead T^0 inside narrow features, and almost never for latter beads.

We use two strategies to mitigate the problem. The first, *Shaving* (Figure 10, third column) reduces the gaps while still respecting $\partial \mathcal{S}^i$ exactly. Shaving is very similar to trimming but in different geometric setting. Shaving labels a sub-axis of the *Collapsed* sub-axis as *Shaved*. The second, *Medial axis simplification* (Figure 10, last column) erases (as opposed to simply giving some label) most of the problematic branches altogether, but at the expense of simplifying the shape $\mathcal{S}^i$. The loss of shape fidelity is well compensated by a large reduction of gaps area.

5.3.1 Shaving. Shaving is applied only to the *Collapsed* sub-axes K^i. Intuitively, shaving removes from every arc a that is incident to a
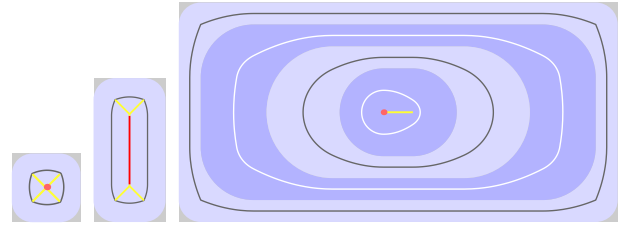

Fig. 11. Rectangles. $2\gamma = 0.3\,\text{mm}$, $2\Gamma = 1.0\,\text{mm}$. The rectangles have size 1x1, 2x1, 6x3 (in millimeters). The last state of the medial axis is shown.

leaf q of K^i the part of a that lies in $D_{\mathcal{S}^i}(q)$ (Figure 10 third column-top). Shaving is similar to trimming and allows for wider beads at local maxima of curvature along $\partial \mathcal{S}^i$.

For the precise definition, we refer to *sub-trees* of K^i as to *w*-trees of $\mathbf{MA}\left(\mathcal{S}^i\right)$ that lie in K^i (a leaf of a sub-tree of K^i is thus a leaf of $\mathbf{MA}\left(\mathcal{S}^i\right)$ and it cannot be an inner point of $\mathbf{MA}\left(\mathcal{S}^i\right)$ where a connected component K^i "starts"). Shaving removes from K^i every maximal sub-tree $\mathcal{T}$ rooted at a point p that lies in all the medial-axis disks centered at the leaves of $\mathcal{T}$ (Figure 10 third column).

These maximal sub-trees are computed with the same algorithm as trimming, but with different equations to solve for the cutting position along one arc. Their arcs are labeled as *Shaved*. When sampling $\text{traj}(T^i)$, the *Shaved* arcs are simply ignored (see Appendix A). Shaving reduces the *Collapsed* sub-axis K^i and therefore reduces underfill around it. The effect of shaving is visible in Figure 10, third column. Figure 11 illustrates on simple rectangles how collapsing and shaving help form the last and innermost bead.

5.3.2 Medial-axis simplification. More radically, another way to reduce the gaps around corners of K^i is to remove the corners. We can do so by removing branches of K^i from $\mathbf{MA}\left(\mathcal{S}^i\right)$. This amounts to using an approximation of the input shape $\mathcal{S}^i$, so we trade a gap around K^i for a gap outside of T^i, due to the approximation error. We take two precautions to avoid unnecessary simplification of the medial axis. First, we simplify *only* the *Collapsed* sub-axis of $\mathbf{MA}\left(\mathcal{S}^i\right)$, and not the whole medial axis. Second, our implementation considers for removal only the *Collapsed* arcs immediately adjacent to a *Collapsed* leaf of K^i and do not propagate the simplification further along K^i.

Let c_0 be the leaf and c_1 the other endpoint of an arc that we consider for removal. If $|c_0 - c_1| + d_{\mathcal{S}^i}(c_1) < \sigma d_{\mathcal{S}^i}(c_0)$ for some tuning-parameter $\sigma \geq 1$, then the arc c_0–c_1 is removed. As seen in Figure 10 (fourth column), the trade-off is favorable. Setting $\sigma = 1$ disables the simplification. Typically, $\sigma \approx 1.05$ gives good results. When used, medial-axis simplification is applied *before* shaving. The results presented in the paper all use $\sigma = 1.05$.

5.4 Taking larger parallel offsets

In some particular case, the trimming process stops early, either because the medial axis is reduced to a point, or because all the roots of the maximal trimmable *w*-trees are on a loop of $\mathbf{MA}\left(\mathcal{S}^i\right)$. Then, the equality in Equation (4) is not reached: For any maximal
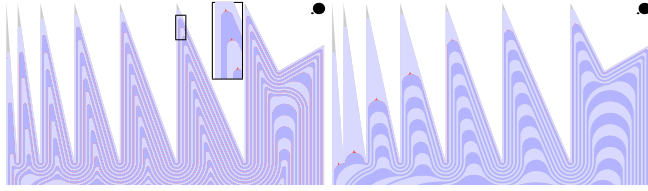
Fig. 12. Illustrating the collapse heuristics. The input shape shows 8 spikes ordered from left to right. *Left.* Base collapse only, up to 4γ, with $2\gamma = 0.08$ and $2\Gamma = 0.5$. *Right.* The full collapse heuristics.

Fig. 13. Varying Γ with $2\gamma = 0.08$ mm. *Left.* The value $2\Gamma = 0.161$ is barely above 4γ. *Right.* $2\Gamma = 2$.

trimmable w-tree $\mathcal{T}$, we have

$$\forall q \in \mathrm{conv}(\mathcal{T}), w(q, \mathrm{root}(\mathcal{T})) < W(q) \tag{6}$$

The minimal residual, $\min_{\mathcal{T}} \min_{q \in \mathrm{conv}(\mathcal{T})} (W(q) - w(q, p))$, represents a length that we can add to 2γ to make the bead wider. The residual is seldom positive, but when it is, we take the opportunity to widen the parallel offset. This is mostly useful to reduce the number of beads needed as soon as some connected component of $\mathcal{S}^i$ reduces to a disk. See e.g. the third bead in Figure 3 (h) which has a width of 2Γ and the second and fourth whose width is larger than 2γ. Note that we do compute a different minimal residual for each connected component of $\mathcal{S}^i$. Finally, the processing of collapsed subaxes depends on the parallel offset be precisely 2γ so the residual is discounted when at least one point of $\mathbf{MA}\left(\mathcal{S}^i\right)$ is labeled *Collapsed*.

6 RESULTS

Section 6.1 illustrates various aspects of our variable-width contouring technique with synthetic results. Section 6.2 discusses the challenges in printing variable-width beads on commodity hardware and show some layers that we have obtained on Ultimaker printers. Section 6.3 discusses statistics obtained for different dense infill methods on a dataset of 300 planar shapes.

6.1 In silico

6.1.1 Modeling a bead, step by step. Figure 17 illustrates the modeling of the first bead on a relatively complex shape. The input polygon P (1) is a linearization of a smooth shape and we see the typical numerous medial axis branches due to the sampling of the boundary. We prepare the 2γ-fat input shape $\mathcal{S}^0$ (2) by cutting the parts of $\mathbf{MA}(P)$ where $d_P < 2\gamma$. In (3) we see the trimming step. We observe that each trimmed sub-axis is indeed a tree. There is a nice bushy tree on the left. Zooming in, we can see how each trimmed tree creates a crescent shape between $\partial \mathcal{S}^0$ and $\mathcal{S}^0_{\mathbf{tr}}$. In (4) we see the collapsing step. Five sub-axes are collapsed. Four are typical, in the ears and feet on the figure. The one on the tail is less typical as it separates $\mathbf{MA}(\mathcal{S}^0)$ into two *Normal* connected components. After collapsing, in (5), a parallel offset is applied. We see $\widetilde{T}^0$ in light blue. In (6), we see that the simplification of the collapsed sub-axes has removed several branches. Is is followed by shaving (the yellow labeling). The medial axis is ready for the sampling of the print trajectory traj(T^0), seen in (7). By construction, the next shape $\mathcal{S}^1$, seen in (8), is 2γ-fat.

6.1.2 On the collapse heuristic. We examine Figure 12 to see the behavior of the base collapse and its extension heuristic.
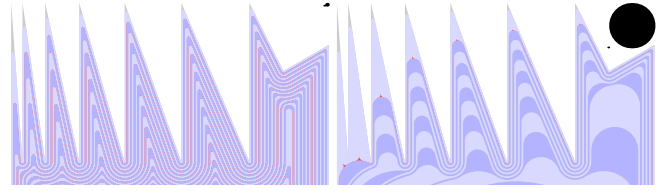
In Figure 12 left, we use the base collapse only (below 4γ, Section 5.2.1). Zooming in (see inset) reveals gaps (from collapsing) everywhere: at every step in every spike but the last, which is open enough for trimming to work well immediately. In order to limit the number of collapses, our heuristics (Figure 12 right, see Section 5.2.2) extends the collapsed axis up to a point where subsequent trimming should work well, or up to 2Γ whichever comes first. We see that we have traded having several collapses (and gaps) with having a single collapse and a larger gap in each spike. In the first spike, collapsing goes all the way out of the spike and stops at 2Γ. In spikes 2 to 4, the same happens. The spikes are too narrow so that trimming fails at increasing the radius at each step along these spikes. The spikes 5 to 7 open up more and the heuristics is able to find a point, high in the spike, where trimming can work well enough to increase the radius in subsequent steps. Indeed we see that radius increases down each spike.

6.1.3 Varying Γ. We examine Figure 13 to see the effect of changing the ratio γ/Γ.

On the left, we see the effect of a small Γ barely above 2γ. In the spikes, the effect is quite similar to using only the base collapse. In open spaces, trimming has very limited freedom.

On the right, we see the effect of a very large Γ. Collapse is able to find good cut points in spikes 4 to 7. And trimming is able to simplify the medial axis much faster.

6.1.4 A deformed ring. We see in Figure 14 an input shape from [Jin et al. 2017a]. We give a statistical comparison to their work in Section 6.3. In the top-left, we try to reproduce roughly the width-range use by Jin et al.. Top-right: a larger Γ helps reduce the number of gaps. Bottom-left: an even larger Γ give beads with extreme variation in width. Bottom-right shows that the bottom-left result is not optimal in the sense that the bottom-right beads are within the bottom-left width range, but were not found by our method with $2\gamma = 0.3$ mm. That's because we cautiously use a parallel offset of 2γ. With $2\gamma = 0.65$ mm we force the collapse of the complete medial axis. Then medial axis simplification and shaving let the two beads (almost) completely cover the input shape. The bottom-right result shows only a tiny amount of underfill, at the cost of having (very subtly) simplified the input shape, while the bottom-left result fully respects $\mathcal{S}^0$. By construction, there is no overfill.

6.1.5 A complex collapsed axis. Figure 15 shows an input polygon that exhibits only narrow features. The bead width range is $[0.3, 0.7]$ mm, so that the lower right of the input shape cannot be covered with our technique and is removed. In the top-right picture, accordingly, we see that the lower-right part of the medial
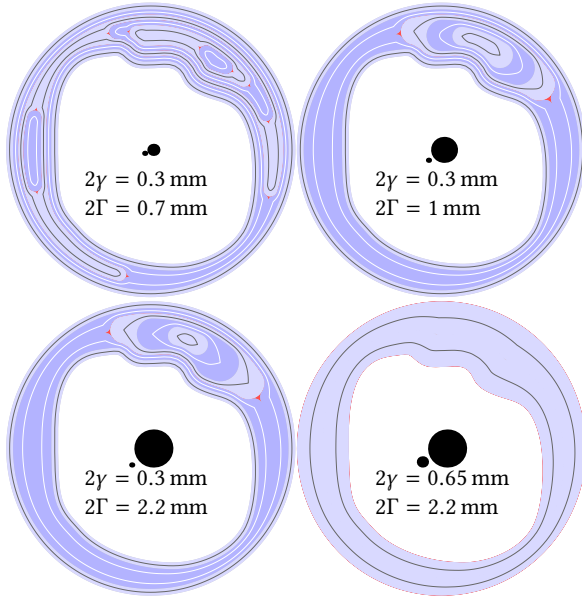
Fig. 14. Variable-width contouring on Jin's ring [Jin et al. 2017a] with various values of γ and Γ. Underfill in red. There is no overfill.
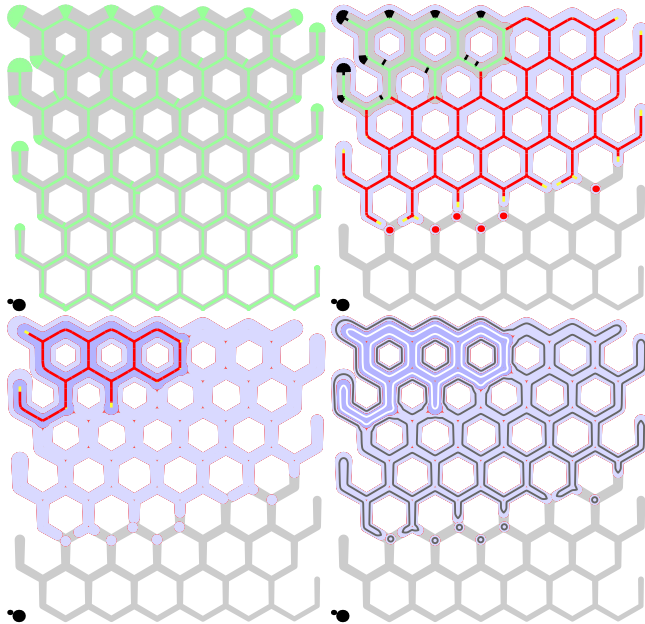

Fig. 15. A complex collapsed axis. In reading order: The input polygon and its medial axis. The labeled medial axis $\mathbf{MA}\left(\mathcal{S}^0\right)$. The labeled medial axis $\mathbf{MA}\left(\mathcal{S}^1\right)$. The resulting two beads and their trajectories (black for T^0, white for T^1).

axis has been removed. We also see that remaining thin features have been collapsed while the upper-left part is wide enough to afford a regular trimming, during the modeling of the first bead T^0. On the bottom-left: we see that the remaining medial axis is
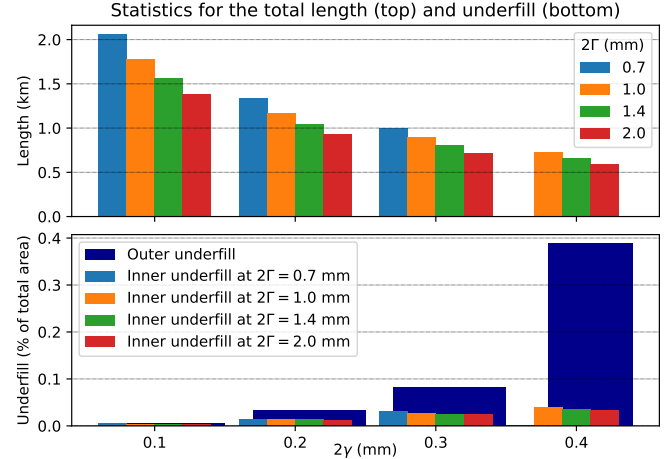

Fig. 16. 2γ varies from 0.1 to 0.4 and 2Γ from 0.7 to 2 (see the legend).

completely collapsed for the modeling of the second bead T^1. The print trajectories of both beads are shown on the bottom-right.

6.1.6 Varying the bead width range. In Figure 16 we look at the total length of the print trajectories, and inner and outer underfill under fifteen different bead width ranges, computed over the 300 input polygons of our dataset. On the top bar plot, we see that increasing Γ reduces the total length of the print trajectories. On the bottom bar plot, we see how the outer underfill depends only on γ and the inner underfill is slightly improved with increasing Γ.

6.2 In plastico

We have performed test prints on an Ultimaker S5 and an Ultimaker 2+ system using PLA filament in a 0.4 mm nozzle.

Varying the bead widths requires the variation of the extrusion process, which depends on the movement speed and the pressure in the system. Changing the pressure inside the system by feeding an excess or lack of material is a strategy commonly employed by *pressure advance algorithms* in order to accomplish constant width beads under varying movement speeds [Arntsønn 2019]. However, our printing system has a large distance between the filament feeder and the nozzle, which means that the amount of material needed to change the pressure inside the system is prohibitively large. We therefore employ the *back-pressure compensation* method proposed by Kuipers et al. [2020]. The bead width variation is accomplished by varying the movement speed according to the following formula:

$$v(w) = \frac{f(w)}{hw} \tag{7}$$

$$f(w) = f_0 - k\left(w/w_0 - 1\right) \tag{8}$$

$$f_0 = v_0 w_0 h \tag{9}$$

where $v(w)$ is the movement speed as a function of the requested bead width w, $f(w)$ is the filament outflow, h is the layer thickness, k is the amount of back pressure compensation and f_0 is a constant reference flow based on the reference speed v_0 and the reference width w_0. The bead width variation is accomplished by changing the movement speed while keeping the internal pressure in between the
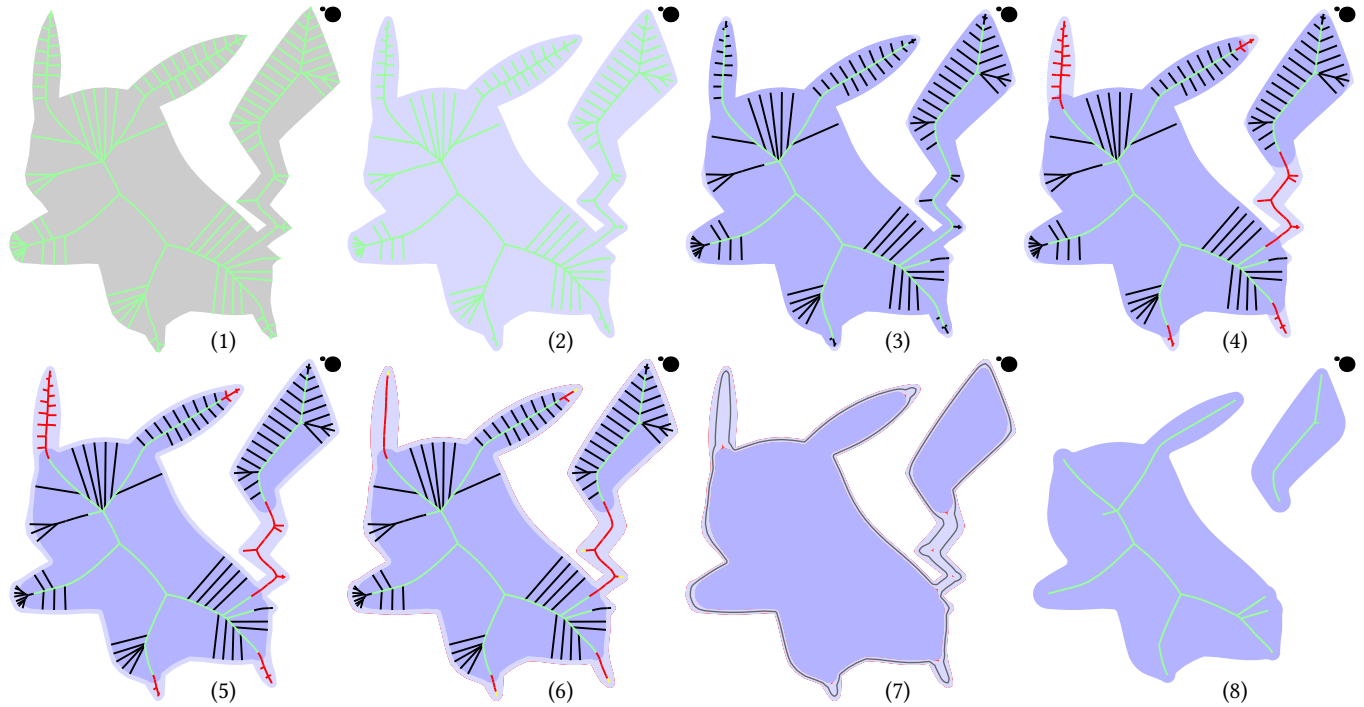
Fig. 17. The bead width range is [0.3, 1.0]. The dark blue shape is generated by $Normal(\mathbf{MA}(\mathcal{S}^0))$. In reading order. *1:* The input polygon and its medial axis. *2:* Shape $\mathcal{S}^0$ and its medial axis. *3:* Trimming. *4:* Collapsing. *5:* Parallel offset (only $d_{\mathcal{S}^0}$ is modified: the dark blue shape is $\mathcal{S}^1$). *6:* Medial axis simplification and shaving. *7:* Sampling of $\mathrm{traj}(T^0)$. *8:* The medial axis of $\mathcal{S}^1$ is much simpler.

filament feeder and the nozzle constant. The increased back pressure from the layer below induced by wider beads is compensated for in order to actualize the bead width variation accurately. We set the reference speed v_0 to 30 mm/s and the reference width w_0 to 0.4 mm.

On the Ultimaker S5 system, we tested prints with a layer thickness $h = 0.1$ mm and bead width range [0.3, 0.7]mm. A back pressure compensation of $k = 1.1$ yields satisfactory results. In order to compensate for the micro-gaps underneath the sides of each bead (whose extrusion profile is not a perfect rectangle) we reduced the material flow to 80 %.

Figure 18 bottom-left shows the capabilities and limitations of our hardware setup by printing only the beads with even index. Figures 18 bottom and 19 show scans of various tool paths generated using our method and printed using the back pressure compensation approach. Note the loss of detail in the SIGGRAPH logo because these shapes are not 2γ-fat. Due to several inaccuracies in the deposition system the extrusion rate fluctuates slightly, resulting in some thin long gaps in between the bead. Most of the printed geometry seems slightly wider than the intended geometry. This might be due to the fact that if a printed line is unconstrained on one side it can bulge more in that direction. On the other hand the back pressure compensation approach might be less accurate for thin lines, which generally occur more frequently on the outside of the print. More accurate dynamic extrusion control is required.

On the Ultimaker 2+ system, we tested thicker prints with $h = 0.2$ mm and bead width range [0.3, 1.0]mm. We found that a value of $k = 1.44$ yields good results provided we further clamp the speed $v(w)$ to the interval [4.5, 30]mm/s. The lower bound 4.5 mm/s is used to avoid too long print time and has no effect on the quality. The upper bound 30 mm/s is to improve the quality of thin beads; It is rather conservative and good results are also obtained at higher speeds.

Figure 18 top shows 4 samples printed on the Ultimaker 2+ system. Here, we have further reduced the flow by some constant factor until the gaps were clearly visible. The computed beads are shown next to each print for comparing the position and size of the gaps.

6.3 In simulacra

Data set. We have tested two different setups: one with a narrow range of $[2\gamma, 2\Gamma] = [0.3\,\mathrm{mm}, 0.7\,\mathrm{mm}]$ and one with a wide range of $[0.3\,\mathrm{mm}, 1.0\,\mathrm{mm}]$, corresponding to feasible printed ranges for a layer thickness of 0.1 mm and 0.2 mm respectively. We compare against the techniques presented in the existing literature as emulated by [Kuipers et al. 2020] using a preferred bead width equal to the middle of the narrow range, i.e., $w^* = 0.5$ mm. We use the same dataset as that latter work. It consists of 300 polygonal slices of publicly available 3D shapes. The centered approach was emulated using the values $r_{\min} = 0.8w^*$ and $r_{\max} = 1.25w^*$, while the inward distributed was configured to distribute the deviation from the preferred width over the innermost $N = 2$ print trajectories.
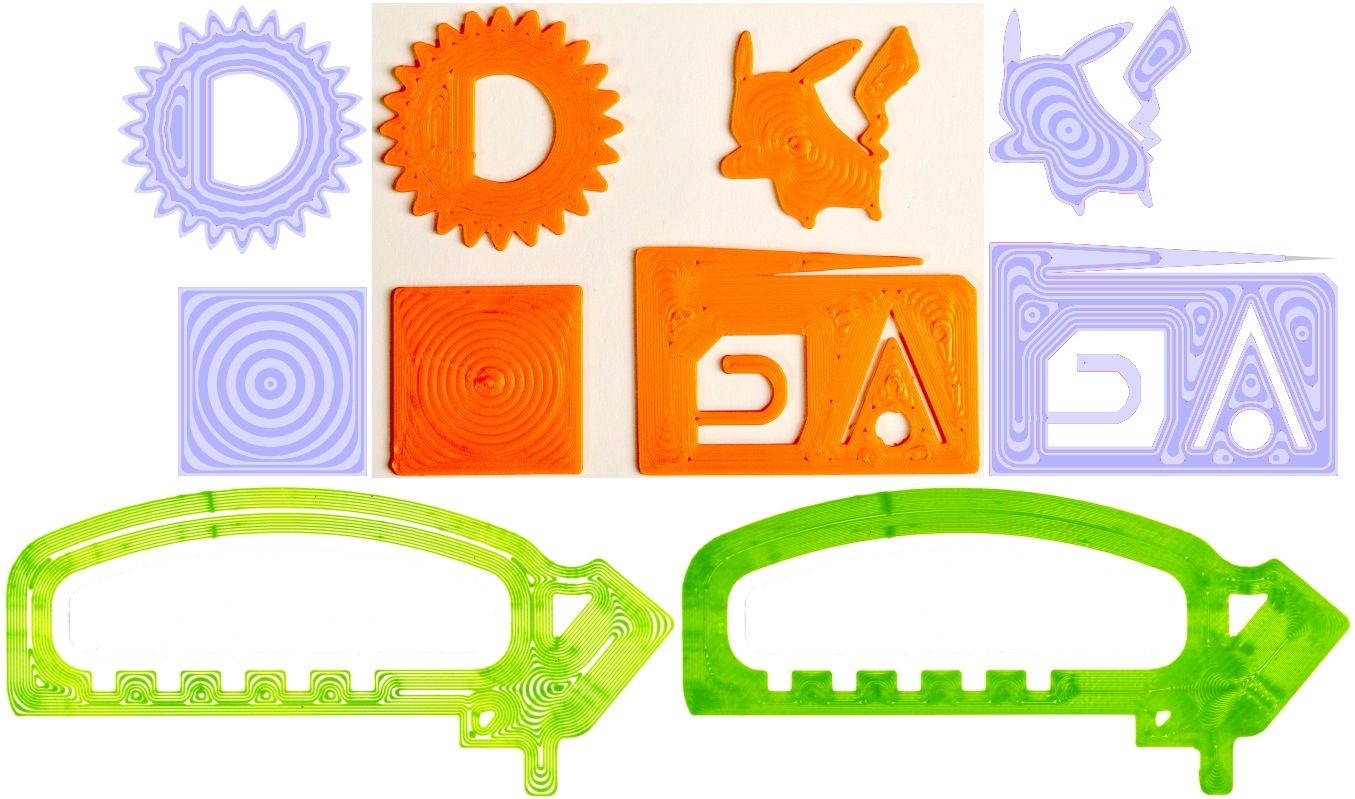
Fig. 18. *Top.* Photographs of prints obtained on an Ultimaker 2+ system (processed to bring out details). Each sample is 4 layers high. Observe how the synthetic gaps (sides, red) correspond to the real ones (middle). *Bottom.* Scans of prints obtained on the Ultimaker S5 system. Each sample is a single layer (green) printed on a white "raft" (a flat printed substratum). On the left, only even beads have been printed. On the right, the strong backlit scan better delineates each bead and reveals a slight uneven thickness across the surface.

Fig. 19. Photographs of prints obtained on the Ultimaker S5. Because $\mathcal{S}^0$ is 2γ-fat rather than γ-fat, a large part of the hexagonal grid is missing; See Figure 15. The second print from the left is $2/3$ the scale of the one on Figure 18. This renders the hook ʔ less that 2γ-fat in the stem, so that a small part of it has been removed.

The five compared techniques are described in the caption of Figure 20. A comparison of the inward approach to [Xiong et al. 2019] can be found in [Kuipers et al. 2020].

Computation time. We evaluated the computation time on an Intel Core i7-7500U CPU @ 2.70 GHz (a single core is used) and 16.3 GB memory. Figure 20a shows how the computation time is related to the vertex count of the input polygon and that each of the methods roughly follows $10^{-5}n \log n$. It seems that for some subset of the data the uniform method performs considerably worse than the other methods. The total computation times over the whole data set are 14.9 s, 5.7 s and 10.7 s respectively for the uniform, the inward distributed and our approach using the narrow bead width range, which shows that all methods perform in the same order of magnitude and take up a negligible amount of time for most practical purposes.

Filling accuracy. In order to estimate the filling accuracy of our method we evaluate the size of underfill and overfill areas of the resulting beads.

We sample the trajectories using a sampling distance of 0.02 mm and generate polygonal areas covering the extruded segment - including a circle with a diameter corresponding to the destination bead width, but excluding the circle corresponding to the start location. We can then estimate the amount of overfill and underfill by performing boolean operations on all segment polygons and the target polygon. We separate the underfill areas which are on the outside of the polygon from the internal underfill areas, because the outer underfill tells us something about how accurately the beads represent the target polygon, while the internal underfills provide information on filling density. These areas are calculated over the whole data set and their total area is compared against the total target area; see Figure 20f
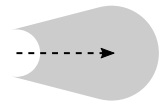
Because of the discretization involved in sampling our evaluation falsely reports a background level of overfill, even though we have proven that no overfill exists in an idealized setting. Even though the reported overfill is inaccurate, we can safely assume that the amount of overfill is considerably lower than the overfill of existing methods. While the error in reported overfill is approximately 0.02 % the smallest reported overfill on existing techniques is an order of magnitude larger: 0.3 %.

Because the presented method does not inherently deal with target areas narrower than 2γ, we see that the outer underfill is larger than for the existing methods. The areas within $[\gamma, 2\gamma]$ could be filled using a different approach which does produce open beads along the medial axis.

The remaining inner underfill is smaller than the current state of the art. While the inward distributed method has a total underfill of 0.210 % our method employing a corresponding narrow range only exhibits 0.050 % underfill. We therefore conclude that our method for generating densely filling beads is ¼ as porous as the existing state of the art.

Extrusion width. Figure 20b shows a histogram of the total length of the print trajectories where the bead has a width within a bin of 0.01 mm over the whole data set. It also reports on the average and standard deviation of the width data. We observe that even though the middle of the range of possible bead widths corresponds with the average bead width of the data from the existing techniques, i.e. 0.5 mm, the average bead width of our technique lies below the middle of the range: 0.36 mm for the narrow bead width range and 0.39 mm for the wide range. This is explained by the fact that beads of width 2γ occur two orders of magnitude more frequently than the other bead widths. The variance in bead width in our method is a lot higher than for the existing methods, which puts extra strain on the manufacturing system requirements.

Manufacturing time. Because the back pressure compensation approach adjusts the movement speeds in order to accommodate the bead width variation, it influences the total time it takes to print the beads. We therefore report both on the manufacturing time when using back pressure compensation and in the ideal case where the (maximum) speeds are unaffected by extrusion width in Figure 20d. While in the ideal case a wider line width causes a higher extrusion flow rate in mm³/s, the back pressure compensation approach reduces the flow rate for beads wider than the nominal bead widths. This means that for the ideal case the manufacturing time is mostly determined by the total trajectories length (Figure 20e), while the Gcode produced using back pressure compensation is dominated by the occurrence of wide lines (Figure 20b). While the manufacturing time for the narrow bead width range is smallest when using back pressure compensation, it is largest in the idealized case. For hardware systems where the back pressure compensation can be realized using some form of pressure advance, the manufacturing time of our method is longer compared to the state of the art. For our current setup, however, our technique employing the narrow bead width range outperforms the state of the art.

In order to accurately manufacture beads which exhibit sharp corners, the print head must decelerate before it comes to a sharp corner. Our print time estimation simulates the acceleration behavior exhibited by the Ultimaker S5 system. Toolpaths that contain a large amount of sharp corners therefore take longer to print. We can see in Figure 20c that our approach produces significantly lower amounts of sharp corners compared to the existing approaches - especially around 90°. Our method therefore features to less stringent requirements on the hardware setup.

Because every separate bead requires a non-extruding travel move, the number of beads influences the manufacturing time as well. Moreover, because open polyline trajectories can only be printed starting from either end they may take longer travel moves than reaching a closed trajectory, which can be started anywhere. While the total path counts in Figure 20e of our method are similar to the state of the art, all of the trajectories are closed paths, indicating that travel time is reduced.

7 CONCLUSION

The technique presented here for variable-width contouring computes a precise mathematical boundary representation of the beads using only line segments and circular arcs. The print trajectories are computed as the medial axis of each bead. These two characteristics allow our technique to guarantee the absence of any inter-bead overfill or bead self-overlap. The technique also makes extensive use of the assumption that the width of a bead can vary within bounds in order to minimize underfill. We believe that the absence of overfill and the minimal underfill makes our technique attractive for additive manufacturing.

In practice, we have verified that by carefully adapting the nozzle speed to their local width, the variable-width beads can be printed on commodity 3D printers. But the variable-width fused filament deposition is very sensitive to tiny mechanical or dimensional variations in the machine. We have also found that the buffers used by the printers firmware to store and analyse the trajectory are undersized for our purpose: a very fine sampling (5 μm) would give smoother bead width variation but the printer chokes on it.

Our technique might then give its fullest on future machines developed with variable-width solidification in mind.

On the algorithmic side, some limitations and future work should be noted:

- Currently the technique requires an initial filtering of the shape so that $\mathcal{S}^0$ is 2γ-fat, while a nozzle capable of a solidification radius of γ can, by definition, fully cover a γ-fat shape. This limitation comes from the way our collapsing operation is not permitted to model a bead sitting over the medial axis $\mathbf{MA}\left(\mathcal{S}^i\right)$. A subtler collapsing might be able to remove that constraint. In the meantime, other existing techniques designed specifically for thin features can help cover $P \setminus \mathcal{S}^0$.
- The geometric modeling of the beads has some limitations as well. Clearly, our technique does not find the set of beads that minimizes underfill. One example is detailed in Section 6.1.4. Two other examples are shown in Figure 21.
- In considering a 2D dense filling problem, we have ignored possible gaps between horizontal layers due to bead alignments. Can we take these gaps into account or devise a fully 3D contouring scheme that minimizes volumetric underfill?
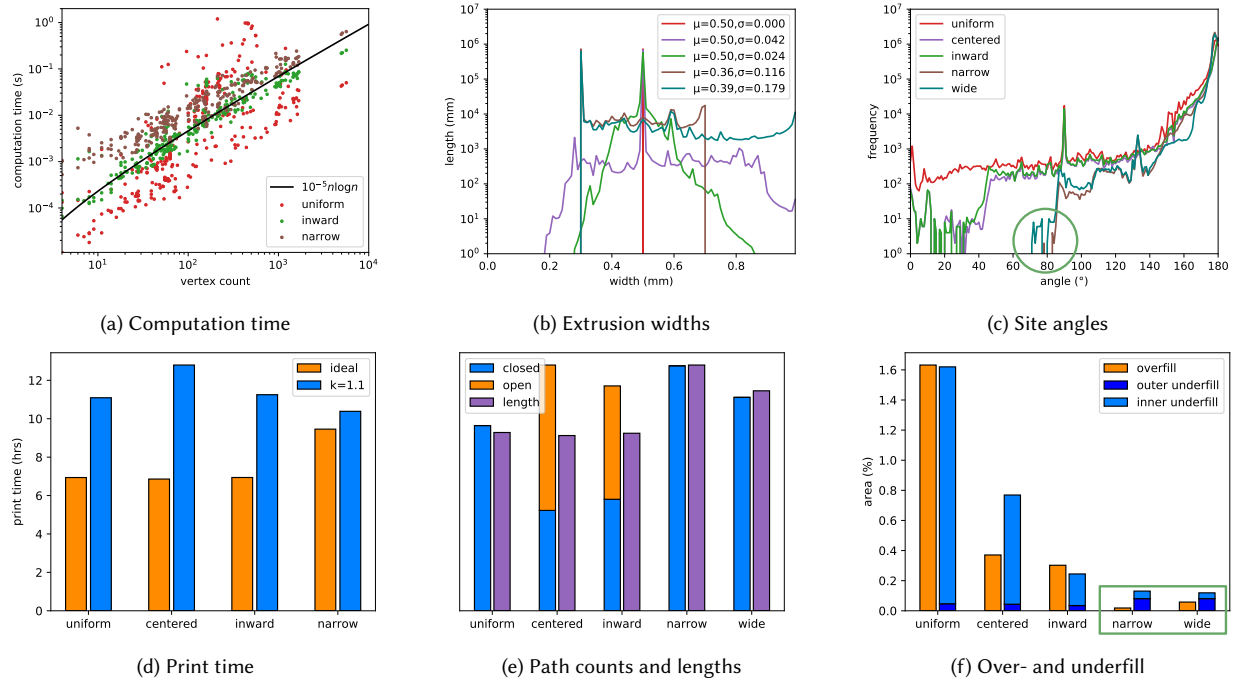
(a) Computation time

(b) Extrusion widths

(c) Site angles

(d) Print time

(e) Path counts and lengths

(f) Over- and underfill

Fig. 20. Statistical analysis of bead generation methods. Our technique employing a range of $[2\gamma, 2\Gamma] = [0.3\,\text{mm}, 0.7\,\text{mm}]$ is referred to as **narrow**, while **wide** refers to $[0.3\,\text{mm}, 1.0\,\text{mm}]$; **inward** is the inward distributed technique proposed by [Kuipers et al. 2020]; **centered** is the method proposed by [Jin et al. 2017a] and **uniform** refers to the uniform-width contouring. Note the use of a logarithmic scale on the top y-axes and on the top-left x-axis.
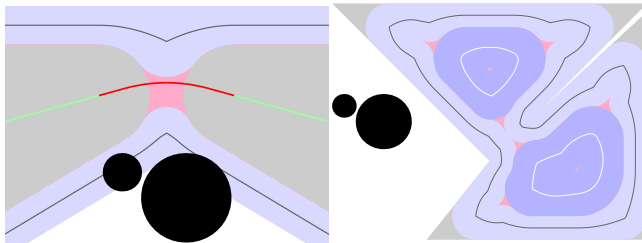

Fig. 21. Limitations. *Left.* A short *Collapsed* axis (middle) prevents the expanded bead to reach it. *Right.* The two gaps in the center could be replaced by a single one by cutting the input shape between those gaps, into two triangular shapes.

- On the theoretical front, some natural questions are raised: What other approach could get less underfill or even find a global minimum? Can we prove a lower bound on the number of gaps or their total area that would depend on γ, Γ and in the input shape? Can we characterize the family of shapes that can be covered without underfill by beads whose width varies within $[2\gamma, 2\Gamma]$?

ACKNOWLEDGMENTS

REFERENCES

Tronvoll Sigmund Arntsønn. 2019. Investigating pressure advance algorithms for filament-based melt extrusion additive manufacturing: theory, practice and simulations. 25, 5 (01 Jan 2019), 830–839. https://doi.org/10.1108/RPJ-10-2018-0275

H. Blum. 1967. A transformation for extracting new descriptors of shape. In *Models for the Perception of Speech and Visual Form*, Weinant Wathen Dunn (Ed.). MIT Press, 362–381.

Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. 2008. *Computational Geometry: Algorithms and Applications* (2 ed.). Springer-Verlag.

Donghong Ding, Zengxi Pan, Dominic Cuiuri, Huijun Li, and Nathan Larkin. 2016. Adaptive path planning for wire-feed additive manufacturing using medial axis transformation. *Journal of Cleaner Production* 133 (2016), 942–952. https://doi.org/10.1016/j.jclepro.2016.06.036

Donghong Ding, Zengxi Stephen Pan, Dominic Cuiuri, and Huijun Li. 2014. A tool-path generation strategy for wire and arc additive manufacturing. *The international journal of advanced manufacturing technology* 73, 1-4 (2014), 173–183.

Kunnayut Eiamsa-ard, Frank W Liou, Robert G Landers, and Howie Choset. 2003. Toward automatic process planning of a multi-axis hybrid laser aided manufacturing system: skeleton-based offset edge generation. In *ASME 2003 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 227–235.

Samuel Hornus and Sylvain Lefebvre. 2018. Iterative carving for self-supporting 3D printed cavities. In *Eurographics 2018 - Short Papers*. Delft, Netherlands. https://hal.inria.fr/hal-01764291

Yuan Jin, Jianke Du, and Yong He. 2017a. Optimization of process planning for reducing material consumption in additive manufacturing. *Journal of Manufacturing Systems* 44 (2017), 65–78. https://doi.org/10.1016/j.jmsy.2017.05.003

Yuan Jin, Yong He, Guoqiang Fu, Aibing Zhang, and Jianke Du. 2017b. A non-retraction path planning approach for extrusion-based additive manufacturing. *Robotics and Computer-Integrated Manufacturing* 48 (2017), 132 – 144.

Yu An Jin, Yong He, and Jian Zhong Fu. 2013. An adaptive tool path generation for fused deposition modeling. In *Advanced Materials Research*, Vol. 819. Trans Tech Publ, 7–12.

Ju-Hsien Kao and Fritz B Prinz. 1998. Optimal motion planning for deposition in layered manufacturing. In *Proceedings of DETC*, Vol. 98. 13–16.

Menelaos Karavelas. 2019. 2D Segment Delaunay Graphs. In *CGAL User and Reference Manual* (5.0 ed.). CGAL Editorial Board. https://doc.cgal.org/5.0/Manual/packages.html#PkgSegmentDelaunayGraph2

Tim Kuipers, Eugeni L. Doubrovski, Jun Wu, and Charlie C. L. Wang. 2020. A framework for adaptive width control of dense contour-parallel toolpaths in fused deposition modeling. arXiv:2004.13497 [cs.GR] In submission.

Tim Kuipers, Jun Wu, and Charlie C.L. Wang. 2019. CrossFill: Foam Structures with Graded Density for Continuous Material Extrusion. *Computer-Aided Design* 114 (2019), 37–50.

Jusung Lee and Kunwoo Lee. 2016. Block-Based Inner Support Structure Generation Algorithm for 3D Printing Using Fused Deposition Modeling. *The International Journal of Advanced Manufacturing Technology* (2016). https://doi.org/10.1007/s00170-016-9239-3

Dawei Li, Ning Dai, Xiaotong Jiang, and Xiaosheng Chen. 2015. Interior Structural Optimization Based on the Density-Variable Shape Modeling of 3D Printed Objects. *The International Journal of Advanced Manufacturing Technology* 83, 9 (2015), 1627–1635. https://doi.org/10.1007/s00170-015-7704-z

André Lieutier. 2004. Any open bounded subset of Rn has the same homotopy type as its medial axis. *Computer-Aided Design* 36, 11 (2004), 1029 – 1046. https://doi.org/10.1016/j.cad.2004.01.011 Solid Modeling Theory and Applications.

Marco Livesu, Stefano Ellero, Jonàs Martínez, Sylvain Lefebvre, and Marco Attene. 2017. From 3D Models to 3D Prints: An Overview of the Processing Pipeline. *Computer Graphics Forum* 36 (2017), 537–564.

Jonàs Martínez, Samuel Hornus, Haichuan Song, and Sylvain Lefebvre. 2018. Polyhedral Voronoi diagrams for additive manufacturing. *ACM Transactions on Graphics* 37, 4 (Aug. 2018), 15.

Sara McMains, Jordan Smith, Jianlin Wang, and Carlo Sequin. 2000. Layered manufacturing of thin-walled parts. In *ASME Design Engineering Technical Conferenced*.

Inria Team MFX. 2017. IceSL, advanced modeling and slicing for 3D printing. https://icesl.loria.fr. Accessed: 2020-04-23.

Maarten Moesen, Tom Craeghs, Jean Pierre Kruth, and Jan Schrooten. 2011. Robust beam compensation for laser-based additive manufacturing. *CAD Computer Aided Design* 43, 8 (2011), 876–888. https://doi.org/10.1016/j.cad.2011.03.004

Romain Prévost, Emily Whiting, Sylvain Lefebvre, and Olga Sorkine-Hornung. 2013. Make It Stand: Balancing Shapes for 3D Fabrication. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH)* 32, 4 (2013), 81:1–81:10.

Emmanuel Sabourin, Scott A. Houser, and Jan Helge Bøhn. 1997. Accurate Exterior, Fast Interior Layered Manufacturing. *Rapid Prototyping Journal* 3, 2 (jun 1997), 44–52. https://doi.org/10.1108/13552549710176662

Thibault Tricard, Frédéric Claux, and Sylvain Lefebvre. 2019. Ribbed support vaults for 3D printing of hollowed objects. *Computer Graphics Forum* (June 2019).

Jun Wang, Ting-wei Chen, Yu-an Jin, and Yong He. 2019. Variable bead width of material extrusion-based additive manufacturing. *Journal of Zhejiang University-SCIENCE A* 20, 1 (01 Jan 2019), 73–82. https://doi.org/10.1631/jzus.A1700236

Weiming Wang, Baojun Li, Sicheng Qian, Yong-Jin Liu, Charlie C. L. Wang, Ligang Liu, Baocai Yin, and Xiuping Liu. 2017a. Cross section-based hollowing and structural enhancement. *The Visual Computer* 33, 6 (2017), 949–960. https://doi.org/10.1007/s00371-017-1386-5

W. Wang, Y. J. Liu, J. Wu, S. Tian, C. C. L. Wang, L. Liu, and X. Liu. 2017b. Support-Free Hollowing. *IEEE Transactions on Visualization and Computer Graphics* (2017). https://doi.org/10.1109/TVCG.2017.2764462

Han Wenbiao, Mohsen Jafari, S.C. Danforth, and Ahmad Safari. 2002. Tool Path-Based Deposition Planning in Fused Deposition Processes. *Journal of Manufacturing Science and Engineering-transactions of The Asme - J MANUF SCI ENG* 124 (05 2002). https://doi.org/10.1115/1.1455026

Jun Wu, Charlie C.L. Wang, Xiaoting Zhang, and Rüdiger Westermann. 2016. Self-supporting rhombic infill structures for additive manufacturing. *Computer-Aided Design* 80 (2016), 32–42. https://doi.org/10.1016/j.cad.2016.07.006

Yi Xiong, Sang-in Park, Suhasini Padmanathan, Audelia Dharmawan, Shaohui Foong, David Rosen, and Gim Soh. 2019. Process planning for adaptive contour parallel toolpath in additive manufacturing with variable bead width. *International Journal of Advanced Manufacturing Technology* 105 (12 2019). https://doi.org/10.1007/s00170-019-03954-1

Y Yang, HT Loh, JYH Fuh, and YG Wang. 2002. Equidistant path generation for improving scanning efficiency in layered manufacturing. *Rapid Prototyping Journal* 8, 1 (2002), 30–37.

Haisen Zhao, Fanglin Gu, Qi-Xing Huang, Jorge Garcia, Yong Chen, Changhe Tu, Bedrich Benes, Hao Zhang, Daniel Cohen-Or, and Baoquan Chen. 2016. Connected Fermat Spirals for Layered Fabrication. *ACM Trans. Graph.* 35, 4, Article 100 (July 2016), 10 pages. https://doi.org/10.1145/2897824.2925958

Appendices

A SAMPLING THE PRINT TRAJECTORIES

This section explains how we sample the medial axis of T^i, which is the trajectory traj(T^i) to be followed by the printer nozzle. traj(T^i) is, by construction, a set of closed simple curves that corresponds to the medial axis of $\widetilde{T}^i$ from which the branches have been removed. Branches may appear only around the *Collapsed* sub-axis K^i.

We sample traj(T^i) by projecting an adaptive sampling of its outer boundary ∂S^i onto it. Since all the branches of $\mathbf{MA}\left(\widetilde{T}^i\right)$ do lie "on the other side of ∂S^i," that is, between traj(T^i) and ∂S^{i+1}, there is in fact no need to take them into account and they can simply be ignored. (They are never actually computed.)

The sampling of ∂S^i proceeds by walking around each cycle of $\mathbf{MA}\left(S^i\right)$, from a half-arc a to the next by following the a.next reference. For each half-arc a, we sample its right-curve right(a) $\subset \partial S^i$. A sample is defined as a pair (p, t) where p is a point and t is a unit vector tangent to the sampled curve. Each sample (p, t) of ∂S^i is projected onto traj(T^i) along the direction $\bar{t}$ which is the vector t rotated 90 degrees to the left. The projection itself depends on the label of the half-arc a.

Normal projection. If half-arc a is labeled *Normal*, then the bead was locally formed only with parallel offset. The projection of (s, t) is then a translation by $\alpha\bar{t}$, where $\alpha = \gamma$ or some larger value specific to the component S^i, see Section 5.4.

Trimmed projection. Before walking along a *Trimmed* w-tree, we look at its root v, which is the last *Normal* vertex visited. While walking around the w-tree, traj(T) is locally the bisector between ∂S^i and $D_{S^{i+1}}(v)$. We project the samples (p, t) onto traj(T) by solving for x in $|p + x\bar{t} - v| = x + d_{S^{i+1}}(v)$. Squaring both sides and simplifying, we obtain a linear equation. The tangent at the projected sample point p' is the normalized average of t and the tangent to $D_{S^{i+1}}(v)$ at its point closest to p'.

Collapsed/Shaved projection. Along a walk-cycle in $\mathbf{MA}\left(S^i\right)$ a connected *Collapsed*/*Shaved* sub-walk w of that cycle, by construction, either forms the complete cycle or is bounded by *Normal* vertices. Let K be the set of *Collapsed* arcs in w (ignoring the *Shaved* ones). Along the walk w, the print trajectory traj(T^i) is the bisector between ∂S^i and $K \cup S^{i+1}$. Before projecting the samples from right(w) onto traj(T^i), we gather the relevant parts of $w \cup S^{i+1}$ in a set $B(w)$. The set $B(w)$ includes *(i)* all the arcs of K that are outside of S^{i+1} as well as *(ii)* the medial-axis disks in S^{i+1} whose boundaries are crossed by w as the sub-axis exits and enters S^{i+1}. For example, in Figure 10 (top-left) the set B contains only one disk, the one at the top. In Figure 17 (6), the two *Collapsed* walks in the tail exit and enter S^1 in two different disks. The projection itself is brute force: We project each sample s onto the bisector with each element of $B(w)$ and pick the projected sample closest to s.

Samples on traj(T^i) are added adaptively as long as the angle between the tangents at each end of a line segment is too large, or the ratio of the radii $d_{\widetilde{T}^i}$ at each end of the segment is outside $[\frac{1}{1.1}, 1.1]$, except if the segment is too small.

B ON THE OVERLAP

Definition of overlap. Let $c : [0, 1] \mapsto \mathbb{R}^2$ be a print trajectory and $R_c : [0, 1] \mapsto \mathbb{R}$ the bead radius at point $c(t)$. Let $D_c(t) = D(c(t), R_c(t))$ be the extrusion disk at parameter t. Let p be a point in the plane. Let $I(p, c)$ be the set of parameters at which p is contained in $D_c(t)$: $I(p, c) = \{t \in [0, 1] \mid D_c(t) \ni p\}$. The set I is a set of intervals in $[0, 1]$. We define the *overlap number* $O(c, p)$ of c over p as the number of maximal intervals in $I(p, c)$ (i.e., the number of connected components).§ Let T_c be the bead $T_c = \cup_{t \in [0,1]} D_c(t)$. We say that T_c has no overlap if $\forall p \in T_c, O(c, p) = 1$. Conversely, a bead T_c exhibits overlap if it contains one point p such that $O(c, p) > 1$.

Proof of no overlap. We now prove that variable-width contouring produces only beads with no overlap. The main ingredient is

LEMMA 1. *Let $\mathcal{S}$ be an open set in $\mathbb{R}^2$ and p a point in $\mathcal{S}$. Let M be the set of points $x \in \mathbf{MA}(\mathcal{S})$ such that $D_{\mathcal{S}}(x) \ni p$. Then M has the homotopy type of a point. (That is, M is a geometric tree.)*

PROOF. We consider $H = \cup_{x \in M} D_{\mathcal{S}}(x) \subset \mathcal{S}$. We prove that $\mathbf{MA}(H) = M$ (Lemma 2). Since H is an open set, it has the homotopy type of its medial axis [Lieutier 2004]. On the other hand, H is star shaped (Lemma 3), so it has the homotopy type of a point, and so has $M = \mathbf{MA}(H)$. □

Lemmas 2 and 3 are proved below. Let T_c be a bead such that the image of c is the medial axis of T_c. (This is the case for all the beads that our algorithm constructs. This is also the case for the *regular* parallel contouring but *not* for *uniform* parallel contouring.) We also take c continuous and injective (no back-and-forth along the medial axis). Let p be a point in T_c. Lemma 1 tells us that $I(p, c)$ has the homotopy type of a point. Since $\mathbf{MA}(T_c)$ is a simple curve, then $I(p, c)$ must be a single interval, which proves that we have no overlap.

LEMMA 2. $\mathbf{MA}(H) = M$.

PROOF. The easier half is $M \subset \mathbf{MA}(H)$: Let x be a point in M. $D_{\mathcal{S}}(x) \subset H \subset \mathcal{S}$ has at least two contact points on $\partial \mathcal{S}$. These two contact points are also on the boundary of H, so $x \in \mathbf{MA}(H)$.

Now we prove that $\mathbf{MA}(H) \subset M$: Let x be a point in $\mathbf{MA}(H)$. The disk $D_H(x)$ is a medial-axis disk in H. It also contains p. Indeed, let y be the point of $\partial D_H(x)$ farthest from p. Point y belongs to H so it lies in a disk $D' \subset H$ that contains p. By construction of y, $p \notin D_H(x) \Rightarrow D' \supset D_H(x) \Rightarrow D_H(x)$ is not a medial-axis disk in H, which is false. So, $p \in D_H(x)$. Now, let $C = \partial H \cap \partial D_H(x)$ be the contact points of x w.r.t. H. Suppose that C contains at most one point on $\partial \mathcal{S}$. Then the points of C not in $\partial \mathcal{S}$ lie in the interior of $\mathcal{S}$. So we can find a disk D'' in $\mathcal{S}$ that contains $D_H(x)$ and also contains p, so that: $D_H(x) \subset D'' \subset H$ so $D_H(x)$ is not a medial-axis disk in H, a contradiction. Therefore, C has at least two contact points on $\partial \mathcal{S}$ which means that $x \in M$. □

LEMMA 3. *H is star shaped.*

PROOF. Let q be a point in H. There exists $x \in M$ such that $q \in D_{\mathcal{S}}(x) \subset H$ and, by definition of M, p also is in $D_{\mathcal{S}}(x) \subset H$. So the segment pq is in H. □

C TRIMMING FORMULAS

Given a half-arc $e = p{-}p'$, we want to find if there exists a point $u \in e$ where $w(q, u)$ is equal to $W(q)$ for at least one vertex $q \in p.$bag and $w(q', u) \leq W(q')$ for the other vertices q' of the bag.

For each $q \in p.$bag, the equation $w(q, u \in e) = W(q)$ can be transformed into a quadratic equation as we calculate below. We solve the equation for all vertices $q \in p.$bag. If there is at least one solution point inside e, we pick the solution point u closest to p and declare the w-tree rooted at u as maximal. Otherwise, trimming continues beyond p'. The Euclidean coordinates of u and q are written, respectively, (u_x, u_y) and (q_x, q_y). The frame of reference varies depending on the arc e. Recall that $\mathbf{MA}(\mathcal{S}^i)$ is a subset of the medial axis of an input polygon. As such, the arcs of $\mathbf{MA}(\mathcal{S}^i)$ are defined as bisectors between vertices and edges of that input polygon.

Case EE. Arc e is a bisector of two line segments on $\partial \mathcal{S}^i$ (these segments are parallel offsets of two edges of the input polygon. Let 2θ be the angle they form. We use the line supporting e as the x-axis and put the origin at the intersection of the supporting lines of the two segments. Then $d_{\mathcal{S}^i}(u) = u_x \sin \theta$ and the equation becomes $(u_x \sin \theta + W(q) - d_{\mathcal{S}^i}(q))^2 = (u_x - q_x)^2 + q_y^2$.

Case EV. Arc e is a bisector of an input (concave) vertex Q and an input edge, the arc is parabolic. We take the directrix as the x-axis, in such a way that the focus Q has coordinates $(0, h)$. $d_{\mathcal{S}^i}(u)$ is the distance from u to the disk $D(Q, \text{offset}(\mathcal{S}^i))$ (see Section 4.3 for the definition of $\text{offset}(\mathcal{S}^i)$). Then, $d_{\mathcal{S}^i}(u) = \frac{u_x^2}{2h} + \frac{h}{2} - \text{offset}(\mathcal{S}^i)$. We again obtain (after simplification) a quadratic equation $(u_x - q_x)^2 + (\frac{u_x^2}{2h} + \frac{h}{2} - q_y)^2 = \left(\frac{u_x^2}{2h} + \frac{h}{2} - \text{offset}(\mathcal{S}^i) + W(q) - d_{\mathcal{S}}(q)\right)^2$.

Case VV. Arc e is a bisector between two input vertices P and Q, the arc is a straight line segment. We take their perpendicular bisector as the x-axis, in such a way that P and Q have coordinates $(0, h)$ and $(0, -h)$ respectively. Using $Y = -\text{offset}(\mathcal{S}^i) + W(q) - d_{\mathcal{S}^i}(q)$, we obtain

$$(u_x - q_x)^2 + q_y^2 = (\sqrt{u_x^2 + h^2} + Y)^2 \tag{10}$$

$$\Leftrightarrow \quad Z - 2q_x u_x = 2Y\sqrt{u_x^2 + h^2}, \text{ with } Z = q_x^2 + q_y^2 - h^2 - Y^2 \tag{11}$$

$$\Rightarrow \quad 0 = u_x^2(q_x^2 - Y^2) - u_x q_x Z + \left(\frac{Z^2}{2} - Y^2 h^2\right). \tag{12}$$

The last step, from Equation (11) to Equation (12), is an implication, not an equivalence, so we consider a root of the quadratic polynomial in Equation (12) only if it is also a solution of Equation (10).

---

§ If the curve c is a loop, as our algorithm produces, if an interval contains 0 and another contains 1, we count them as a single interval.