

Connected Fermat Spirals for Layered Fabrication

Haisen Zhao¹ Fanglin Gu¹ Qi-Xing Huang² Jorge Garcia³ Yong Chen⁴ Changhe Tu¹ Bedrich Benes³
Hao Zhang⁵ Daniel Cohen-Or⁶ Baoquan Chen¹

¹Shandong University ²TTI Chicago ³Purdue University ⁴USC ⁵Simon Fraser University ⁶Tel-Aviv University

Abstract

We develop a new kind of “space-filling” curves, *connected Fermat spirals*, and show their compelling properties as a tool path fill pattern for layered fabrication. Unlike classical space-filling curves such as the Peano or Hilbert curves, which constantly wind and bend to preserve locality, connected Fermat spirals are formed mostly by *long, low-curvature* paths. This geometric property, along with continuity, influences the quality and efficiency of layered fabrication. Given a connected 2D region, we first decompose it into a set of sub-regions, each of which can be filled with a single continuous Fermat spiral. We show that it is always possible to start and end a Fermat spiral fill at approximately the same location on the outer boundary of the filled region. This special property allows the Fermat spiral fills to be joined systematically along a graph traversal of the decomposed sub-regions. The result is a globally continuous curve. We demonstrate that printing 2D layers following tool paths as connected Fermat spirals leads to efficient and quality fabrication, compared to conventional fill patterns.

Keywords: connected Fermat spirals, space-filling curve, layered fabrication, tool path, continuous fill pattern

Concepts: •Computing methodologies → Parametric curve and surface models; Shape analysis;

1 Introduction

The emergence of additive manufacturing technologies [Gibson et al. 2015] has led to growing interests from the computer graphics community in geometric optimization for 3D fabrication. The focus of many recent attempts has been on *shape* optimization: how to best configure a 3D shape, e.g., via hollowing or strengthening, to achieve quality and cost-effective fabrication. In this work, we look at the problem from a different angle. Instead of addressing the higher-level question of *what* to print, we examine lower-level yet fundamental issues related to *how* to print a given object.

At the most elementary level, additive or layered fabrication operates by moving a print head which extrudes or fuses print material layer by layer. When printing each layer, the print head follows a prescribed *tool path* to fill the 2D region defined by the shape of the printed object. Topologically, *continuity* of a tool path is critical to fabrication. A tool path discontinuity or contour plurality forces an on-off switching of the print nozzle, negatively impacting build quality and precision [Dwivedi and Kovacevic 2004; Ding

et al. 2014]. Geometrically, sharp turns and corners are undesirable since they lead to discretization artifacts at layer boundaries and cause de-acceleration of the print head, both reducing print speed and degrading fill quality [Jin et al. 2014].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org. © 2016 Copyright held by the owner/author(s). Publication rights licensed to ACM.
SIGGRAPH '16 Technical Paper, July 24 - 28, 2016, Anaheim, CA
ISBN: 978-1-4503-4279-7/16/07
DOI: <http://dx.doi.org/10.1145/2897824.2925958>

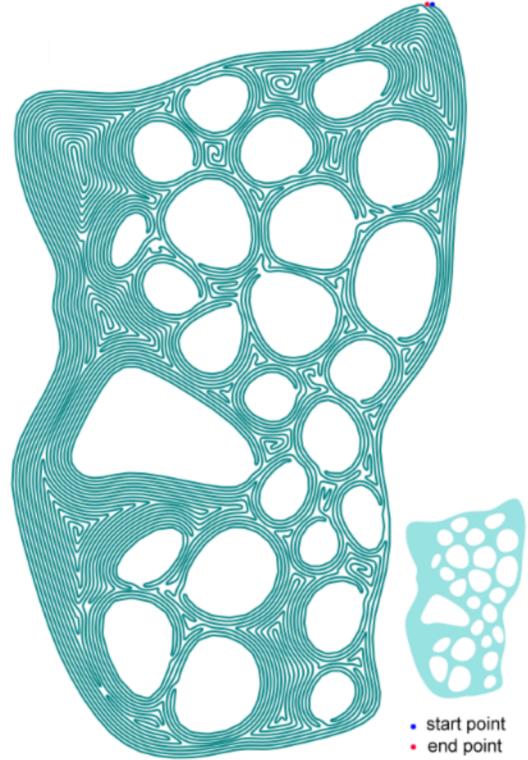


Figure 1: A new kind of “space-filling” curves called connected Fermat spirals. Unlike classical space-filling curves which wind and bend, the new curve is composed mostly of long, low-curvature paths, making it desirable as a tool path fill pattern for layered fabrication. The tool path shown is continuous with start and end points marked; the input 2D layer shape is displayed on the side.

Zigzag has been the most widely adopted fill pattern by today’s 3D printers due to its simplicity [Gibson et al. 2015]; see Figure 2 for various fill patterns. However, a zigzag fill consists of many sharp turns, a problem that is amplified when printing shapes with complex boundaries or hollow structures. A contour-parallel tool path, formed by iso-contours of the Euclidean distance transform, provides a remedy, but it leads to high contour plurality since the iso-contours are disconnected from each other. A spiral fill pattern, for simple shapes such as a square, is continuous. However, for more complex shapes, both contour-parallel fills and spiral fills tend to leave isolated “pockets” corresponding to singularities of the distance transform, as shown in Figure 3(a). These pockets are disconnected and result in path plurality. An intriguing geometry question is whether a connected 2D region can always be filled by a *continuous* pattern formed by one or more spirals.

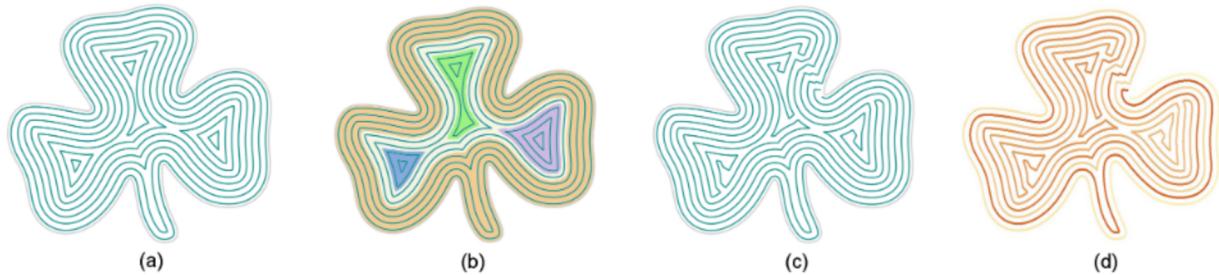


Figure 3: Overview of connected Fermat spiral algorithm. (a) Iso-contours via distance transform lead to four “pockets”. (b) Decomposition into four sub-regions, each to be filled using a single Fermat spiral. (c) Connecting single spirals into a globally continuous curve. (d) Visualization (lower resolution for ease of visualization) of the continuous curve through smooth color transition.

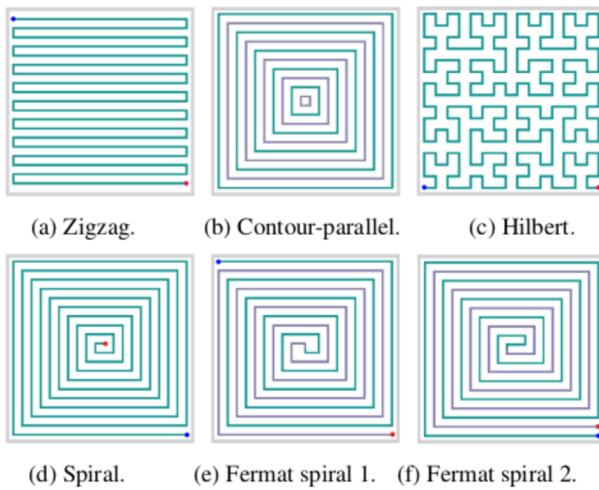


Figure 2: Various space-filling patterns. For a Fermat spiral, the start and exit points on the boundary can be chosen freely (e-f).

In this paper, we introduce the use of *Fermat spirals* [Wikipedia 2015] as a fundamental 2D fill pattern and develop a tool path planning algorithm based on *connected Fermat spirals* or *CFS* to continuously fill a connected 2D region. A Fermat spiral is an interesting space-filling pattern with two interleaving sub-spirals, one inward and one outward; see Figure 2(e). Fermat spirals had not been exploited for tool path planning before and they possess three key properties to make them attractive as a fill pattern:

1. Like contour-parallel paths, a Fermat spiral conforms to the region boundary, with one sharp turn in the center.
2. Several Fermat spirals covering a 2D region can be continuously connected. While a conventional spiral travels either inward *or* outward, a Fermat spiral goes in and *then* out, allowing several of them to be joined at their boundaries.
3. The start and exit points of a Fermat spiral can be chosen arbitrarily over its boundary; see Figures 2(e-f). This special property facilitates connections between Fermat spirals.

We develop an algorithm to construct a CFS curve to fill a singly-connected 2D region. First, the algorithm decomposes the input region into a set of sub-regions each of which admits a continuous fill by a single Fermat spiral; we call these sub-regions *spirallable*. The start and exit points for each Fermat spiral are placed next to each other along the spiral boundary. We then obtain a continuous

traversal of the spirallable sub-regions and connect their respective Fermat spirals through the start/exit points to form a globally continuous curve. Further optimization can be applied to improve fairness and spacing. The resulting curve has fewer sharp turns than a zigzag fill and composed mostly of long, low-curvature paths.

We show CFS curves constructed for input shapes with varying exterior and interior complexity. We fabricate 2D layers and 3D prints using CFS patterns and compare the results visually and numerically to those provided by conventional fills via zigzag and contour-parallel tool paths. The new fill pattern appears to excel at handling shapes with much concavity and many interior holes.

In retrospect, the desirable properties we seek from connected Fermat spirals are almost completely opposite to those possessed by classic space-filling curves such as Hilbert curves; see Figure 2(c). Hilbert or Peano curves are designed to wind and bend to preserve locality of the space traversal. CFS curves are meant to avoid bending as much as possible to attain a higher degree of fairness. In general, CFS curves are not guaranteed to completely cover an arbitrary 2D region even at infinitely high resolution. As well, our new curves do not possess recursive properties as the classical space-filling curves. For layered fabrication however, the new curve is clearly more attractive than fractal-like fill patterns.

2 Background and Related Work

Recently, there has been a flourishing of works in computer graphics on optimizing 3D shapes or their configurations for efficient and effective fabrication, e.g., to ensure or improve physical stability [Prévost et al. 2013], structural strength [Stava et al. 2012; Hildebrand et al. 2013; Lu et al. 2014], or appearance [Zhang et al. 2015] of the print, to save material [Vanek et al. 2014b; Hu et al. 2014], and to adapt to the limited print volume [Luo et al. 2012; Vanek et al. 2014a; Chen et al. 2015; Yao et al. 2015].

In this section, we focus on the tool path planning problem in the context of additive manufacturing (AM). First, to explain the importance of continuity and fairness for the tool paths, we present some background material on nozzle mechanisms that control the viscoelastic material that is extruded along the path, as well as the mechanics of the motors that control the tool paths. Then we discuss related geometric and engineering approaches for optimizing tool paths. Our coverage is not meant to be exhaustive. Interested readers should refer to the book by Gibson et al. [2015], short surveys from [Kulkarni et al. 2000; Ding et al. 2014], as well as the recent SIGGRAPH course by Dinh et al. [2015].

Tool path continuity. Fused Deposition Modeling or FDM is the most widely applied AM technology. During the FDM process,

filament is melted into viscoelastic material and extruded from a small opening of the print nozzle. Due to liquid compressibility, it is generally hard to predict the amount of viscoelastic material to emit in order to create a continuous extrusion control. Consequently, the first portion of the filament from the nozzle usually under- or over-fills. Such uneven fills cause visual artifacts when they occur near the surfaces of the printed object. When they occur between fill paths, attachment between filaments can be weakened, lowering the strength of the print. When turning off the extrusion, the temporal gap between the stopping of the feed motor and that of the filament extrusion is difficult to control, which again leads uneven fills. Similar situations also arise when the print material is fused by the print head, e.g., for powder-based printing. As well, any discontinuity along a tool path necessitates a nozzle movement which does not contribute to the print. Thus, the primary objective in designing tool paths is to minimize the on/off switching along the path, or in other words, to *maximize its continuity*.

Tool path geometry. The geometry of tool paths, in particular their curvature, influences fabrication time and quality. As a tool path rounds itself about a sharp turn, more de-acceleration and acceleration times are required, causing more of a slow-down of the extrusion head, as compared to the case of a soft turn. As well, acute turn angles also lead to more over-fill or under-fill of the filament [Jin et al. 2014]. Hence, a long and continuous tool path without sharp turns may enable the extrusion head to move along the whole tool path at a speed that is close to the highest allowed with small changes, leading to efficient and quality fabrication.

Direction-parallel vs. contour-parallel fills. The most popular fill method in commercial AM systems follows the zigzagging pattern [Ding et al. 2014]. Along with raster scans, zigzagging belongs to the class of direction-parallel fills. In contrast, contour-parallel paths are comprised of a set of closed contours parallel to the outline of a given slice [Yang et al. 2002]. Over simple 2D regions, such paths lead to smoother turns and object boundaries compared zigzagging [El-Midany et al. 1993], but they always have a high contour plurality. Hybrid fills have also been proposed [Jin et al. 2013]; they generate a few contours inward before filling the remaining interior area with a zigzag, but attachment between the two fill patterns can become suspect. When the 2D slice to be filled has a complex shape with many concavities, standard implementations of both fill patterns are prone to discontinuity issues.

Spiral tool paths. Spiral tool paths have been widely applied for pocket machining [Ren et al. 2009]. Held and Spielberger [2014] decompose a 2D layer into spirallable pockets and machine each pocket following a *separate*, classical spiral pattern; no globally continuous path was constructed. Spiral tool paths are less common for AM and one major reason (also applicable to contour-parallel fills) is that due to a lack of direction bias, spiral patterns for adjacent slices replicate each other and cannot be “cross-weaved” at an angle; this could compromise fabrication strength for FDM printers [Gibson et al. 2015]. This problem may be fixed by hybrid fills, e.g., alternating between spiral and zigzagging layers. Our work focuses on how to optimize the continuity of spiral fills.

Space-fill curves. A continuous tool path that fills a 2D region is a space-filling curve (SFC). SFCs have been adopted for various applications, e.g., image encoding [Dafner et al. 2000] and maze design [Pedersen and Singh 2006]. Fractal-like SFCs have been suggested as fill patterns for AM [Wasser et al. 1999]. However, they are complex to realize and full of sharp turns. The tool path fill problem has some resemblance to lawn mowing [Arkina et al. 2000], which is formulated under a rather different setting: it seeks the shortest path for a “cutter” with a prescribed shape to cover all points (possibly multiple times) in a 2D region.

Labyrinths. Mazes and many famous labyrinth patterns are also space-filling [Wikipedia 2016]. A unicursal labyrinth curve is continuous and starts and ends at the same point, just like our connected Fermat spirals. Pedersen and Singh [2006] developed a stochastic curve evolution algorithm to produce expressive, space-filling labyrinths where the Brownian motion of the curve particles are subject to attraction-repulsion forces, as well as *local* fairness and field alignment constraints. In contrast, our algorithm is top-down and with more global structural control in the construction. The resulting curves have smoother boundaries and less turns.

Domain decomposition. One interesting way to obtain a continuous tool path is to decompose a 2D region into several sub-regions each of which admits a continuous fill. Then these regional fills are connected to achieve global continuity. Along these lines, [Dwivedi and Kovacevic 2004] decompose a polygon into monotone sub-polygons and fill each sub-polygon using a *closed* zigzagging curve, along which the start/entry point can be chosen arbitrarily. Ding et al. [2014] execute convex decomposition and for each convex polygon, an optimal zigzagging direction is found to facilitate continuous connection between the polygon fills. However, both methods were designed to deal with polygon inputs and cannot properly handle shapes with smooth concave boundaries.

Our work also relies on a region decomposition while it can deal with arbitrary 2D shapes as input. Instead of using zigzags, we employ Fermat spiral fills to achieve both continuity and a higher degree of fairness. The decomposition scheme is designed to accommodate contour-parallel and spiral tool paths.

Continuously fillable shapes. Polygon convexity [Ding et al. 2014] and monotonicity [Dwivedi and Kovacevic 2004] were chosen in the domain decomposition approaches to achieve tool path continuity since both shape properties ensure a continuous fill by the zigzagging pattern. For monotone polygons, a limited set of scan directions guarantee this, while for a convex polygon, any scan direction leads to a continuous zigzagging fill. Spirallability is their counterpart for spiral or contour-parallel fills and to the best of our knowledge, such a shape property has not been studied before.

3 Spirals, Fermat Spirals, and Spirallability

Before introducing spirals, we first study contour-parallel tool paths and relate them to the Euclidean distance transform of a 2D region. Then, we describe how to convert the parallel contours into a regularly spaced spiral pattern and define spirallability. Finally, a spiral fill is converted into a Fermat spiral fill, where we can choose the start and end points on the boundary arbitrarily.

Contour-parallel path as iso-contour. Let R be a connected 2D region whose boundary is denoted by ∂R . A Euclidean distance transform for ∂R defines a scalar distance field \mathcal{D}_R over R where for each point $p \in R$, $\mathcal{D}_R(p)$ is the shortest distance from p to ∂R . An iso-contour associated with distance d is composed of all points in R whose scalar value is d ; the boundary ∂R is the iso-contour associated with the iso-value 0. Subject to the width of the fill material for the fabrication process, the set of contour-parallel tool paths correspond to a set of equidistant iso-contours, which are all disconnected from each other, as shown in Figure 4(a).

Spiral and spirallability. Two adjacent iso-contours can be connected to form a single continuous path by breaking and *rerouting* the contours, as shown in Figure 4(b). Rerouting adjacent contours in such an offsetting fashion would lead to a spiral pattern. If the distance field within R has a *single* local maximum or plateau, to which all the iso-contours would ascend, then these contours can be rerouted into a single continuous spiral path that fills the 2D

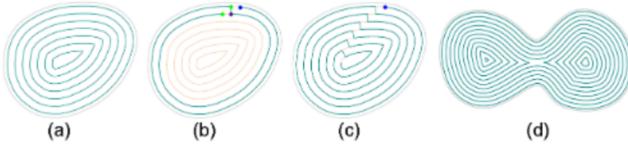


Figure 4: From contour-parallel paths (a) to a spiral (c), by breaking and rerouting adjacent iso-contours (b). In general, the distance transform has multiple local maxima (d). If the maximum is unique, the region is spirallable (a-c).

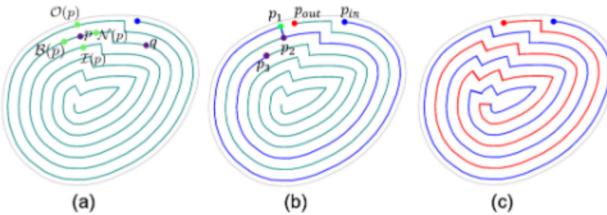


Figure 5: Rerouting a spiral (a) into a Fermat spiral (c). (a) A point p and its corresponding inward $\mathcal{I}(p)$ and outward link $\mathcal{O}(p)$, as well as points $\mathcal{B}(p)$ and $\mathcal{N}(p)$ which are before and after p along the path. (b) Starting at p_{in} , reroute at $p_1 = \mathcal{B}(p_{out})$ and go inward to $p_2 = \mathcal{I}(\mathcal{B}(p_{out}))$, and continue. (c) Resulting Fermat spiral.

region R , as shown in Figure 4(c). We call such a region R *spirallable*. Non-spirallable regions have multiple pockets corresponding to separate local maxima and cannot be converted into a single continuous spiral with a simple rerouting as described.

Fermat spiral. A spiral fill path π for a spirallable region R can be converted into a Fermat spiral. As we show below, we can also choose the start and end/exit points of the Fermat spiral traversal arbitrarily, with both points lying on the region boundary.

Starting from a point $p \in \pi$, trace the upward gradient line over the distance field \mathcal{D}_R to intersect π at $\mathcal{I}(p)$, if it exists. If p is close to the maximum of \mathcal{D}_R or center of the region R , where the path π is thinning out, then the gradient line may not intersect π . We call $\mathcal{I}(p)$ the *inward link* for p with respect to π ; see Figure 5(a). Similarly, we define the *outward link* $\mathcal{O}(p)$ for p by tracing the downward gradient and intersect. If p lies on ∂R , then such an intersection would not exist. The outward and inward links will serve as rerouting points for the conversion to a Fermat spiral.

To help describe the rerouting procedure, we impose a partial order \prec along path π based on inward traversal. Thus, the first point is the end point of π on the region boundary and the last point is at the region center. Two points $p \prec q$ if the inward traversal along π reaches p before q . Next, we impose a discretization spacing δ and denote the point preceding (respectively, succeeding) p along π at a distance δ by $\mathcal{B}(p)$ (respectively, $\mathcal{N}(p)$); see Figure 5(a).

Let p_{in} be the starting point of π and suppose that we would like the Fermat spiral to exist at p_{out} along the outermost portion of π . As shown in Figure 5(b), we start at p_{in} and travel along π until reaching $p_1 = \mathcal{B}(p_{out})$. Then we reroute the path inward from $p_1 = \mathcal{B}(p_{out})$ to its inward link $p_2 = \mathcal{I}(p_1)$, continue traveling along π until reaching $p_3 = \mathcal{B}(\mathcal{I}(\mathcal{B}(p_2)))$, and reroute from this point to its inward link. This form of inward rerouting is executed iteratively until reaching the center of the region, at which point, the traversal is reversed into an outward one with a turn. The outward rerouting is through the outward links, starting at the region center, passing through portions of π that were not traversed during the inward spiral, until the outward spiral exits at p_{out} ; see Figure 5(c).

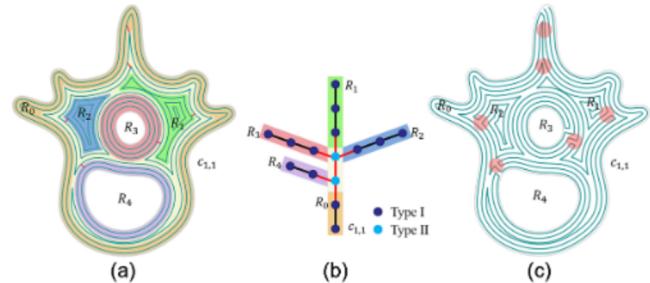


Figure 6: An example of a spiral-contour tree with five spirallable regions. (a) Interior decomposed sub-regions shown in distinctive colors. The short red lines indicate connection locations. (b) The minimum spanning tree of the spiral-contour tree. (c) Connecting adjacent Fermat spirals to form a single continuous path.

The way the rerouting points are placed in the above conversion procedure leads to jaggies or staircasing at every turn along the Fermat spiral. These artifacts are removed by a post-optimization.

4 Continuous Fermat Spiral Fill

In this section, we describe our algorithm for constructing a continuous path fill, as connected Fermat spirals, for an arbitrary, singly-connected 2D region R . The key is to properly reroute levelset curves or iso-contours derived from the Euclidean distance transform of the region boundary ∂R . Within a pocket, the rerouting produces a Fermat spiral. Between pockets and near branching regions, rerouting serves to connect the spirals.

Given a prescribed path fill width w specifying spacing between iso-contours, we construct the set \mathcal{L} of iso-contours using the Clipper algorithm [Johnson 2015] over R . We index an iso-contour by $c_{i,j}$, where i indicates its distance from the region boundary ∂R , $d(\partial R, c_{i,j}) = (i - 0.5)w$, and j is an index among all iso-contours with the same distance index i . For example, $c_{i,j}$ and $c_{i,j'}$ with $j \neq j'$, would belong to two separate pockets. Without loss of generality, we assume that $c_{1,1}$ is always the outer region boundary ∂R .

We build a tree, called the *spiral-contour tree*, whose nodes are the iso-contours and whose edges denote their connectivity with edge weights encoding how preferable it is to connect the iso-contours. The tree is used to recursively reroute the contours in a bottom-up fashion, producing a single continuous path.

Tree construction. We first connect iso-contours with consecutive iso-values, e.g., $c_{i,j}$ with $c_{i+1,j'}$, into an initial graph. To this end, we define a *connecting segment* on $c_{i,j}$ towards $c_{i+1,j'}$ as

$$\mathcal{O}_{i,j,j'} = \{\mathbf{p} \in c_{i,j} \mid d(\mathbf{p}, c_{i+1,j'}) < d(\mathbf{p}, c_{i+1,k}), k \neq j'\},$$

where $d(\mathbf{p}, c)$ denotes the distance from a point \mathbf{p} to points along a contour c . The segment $\mathcal{O}_{i,j,j'}$ is formed by possible rerouting points between the two iso-contours. We add an edge between $c_{i,j}$ and $c_{i+1,j'}$ to the graph if $\mathcal{O}_{i,j,j'} \neq \emptyset$. The weight assigned to the edge is $\text{length}(\mathcal{O}_{i,j,j'})$. The preference is to *not* reroute over a long connecting segment since such a segment is preferred to remain intact to form long, low-curvature paths.

After building the initial graph on iso-contours, we compute a *minimum-weight* spanning tree, the spiral-contour tree, with $c_{1,1}$ as the root; see Figure 6(b). The tree nodes fall into two types. Type I nodes have degrees less than or equal to two and they correspond to iso-contours that form spirallable regions. Specifically, each such region, e.g., R_0, R_1, \dots, R_4 in Figure 6(a), is formed by a path of

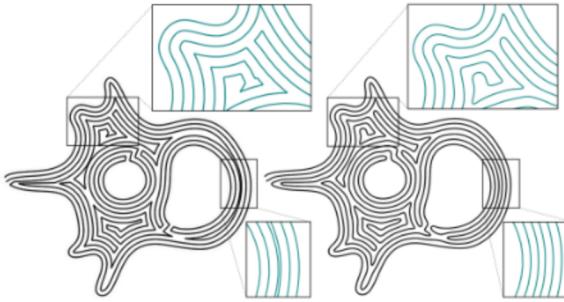
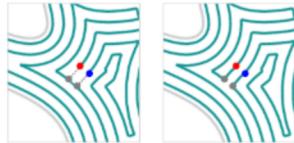


Figure 7: Connected Fermat spirals before (left) and after (right) local optimization. Observe improved curve spacing and reduction of staircasing artifacts in the optimized spirals.

Type I nodes. Type II nodes have degrees greater than two, e.g., those colored in light blue in Figure 6(b), and they correspond to branching iso-contours. Such an iso-contour provides an interface between spirallable regions and possibly other Type II nodes.

Rerouting. To obtain a globally continuous path, we reroute the iso-contours in a bottom-up fashion, starting from leaf nodes and ending at the root. There are two types of rerouting operations. The first connects iso-contours in a spirallable region, e.g., R_0 in Figure 6, into a single Fermat spiral with start and exit points next to each other. This operation follows the procedure described in Section 3 and illustrated in Figure 5. The second operation connects the start and exit points of a Fermat spiral to a Type II iso-contour, at the closest points (gray points), as shown in the inset figure. Wherever possible, rerouting points are reused to avoid creating new points representing sharp turns.



Curve optimization. The tool path obtained so far is globally continuous and covers the input region R , but it is only C^0 continuous and possibly suffers from highly nonuniform spacing. In post-processing, we locally optimize the curve to improve its fairness and spacing. The current curve is first adaptively sampled based on curvature so that more samples are placed near sharp turns. The objective function is a weighted sum of three terms: the first term penalizes large perturbations; the smoothing term is defined by a chord-length weighted 1D discrete Laplacian; and the spacing term keeps shortest distances between adjacent curve segments close to a fixed, pre-defined patch spacing. We solve the optimization via iterative Gauss-Newton until curve updates become negligible; a result showing the paths before and after optimization is shown in Figure 7. Implementation details of the optimization step, including precise problem formulation, optimization procedure, and parameter setting, can be found in the appendix.

5 Results

We show tool path generation results on shapes with varying degrees of concavity and hollowness. Comparisons are made to conventional zigzag and contour-parallel fill patterns in terms of path continuity, amount of sharp turns, print time, as well as visual quality of the interior fill and fabricated surface exterior.

3D printer and setting. Our experiments have been conducted on a RepRap Prusa i3 FDM printer with firmware Marlin 1.1.0-RC. Printing results and analyses are based on the default printer

Input	#segZ	#segC	%stZ	%stC	%stF
dancer 1	22	14	5.87%	1.40%	1.38%
dancer 2	19	10	6.58%	1.55%	1.08%
dancer 3	21	13	4.11%	1.19%	0.81%
crane	8	17	4.86%	0.46%	0.93%
butterfly	16	24	1.81%	0.83%	0.52%
hand	9	11	4.84%	1.07%	0.56%
gear	51	105	1.18%	2.11%	0.23%
paw	20	55	1.25%	0.51%	0.31%
h-slice1	53	58	4.35%	1.08%	0.81%
h-slice2	47	56	5.12%	0.88%	0.70%

Table 1: Number of tool path segments (#seg) and percentage of sharp turn points (%st), which we explain in the text, for conventional zigzag (Z), contour-parallel (C), and our CFS fills (F). The 10 shapes are from the last two rows of Figure 8.

setting, with tool path width set at 0.4mm, layer thickness at 0.2mm, and maximal nozzle speed at 80 mm per second. G-code is used to transfer the tool paths to the 3D printer.

Tool path generation. Figure 8 shows tool paths generated by our algorithm for a variety of shapes with varying exterior and interior structures. Note that the two honeycomb input shapes in Figure 8 and the one from Figure 1 are all 2D slices of the 3D porous structures constructed by the work of Lu et al. [2014]. Each tool path is continuous and composed of connected Fermat spirals. All results are produced with the default parameter setting. There are no tunable parameters for initial CFS construction. For curve optimization, the parameters are fixed as discussed in the Appendix.

Table 1 shows the percentage of sharp turns and the number of disconnected tool path segments for three fill patterns: conventional zigzag, contour-parallel fills, and ours. We do not report the latter number for CFS since it always produces a single path. All the zigzag and contour-parallel paths shown and fabricated in our experiments were generated with the Slic3r software [2016].

To count the number of sharp turns along a tool path π , we uniformly sample 50,000 points along π and at each point, we estimate its integral curvature [Pottmann et al. 2009] with a circle of radius 0.2mm, which is appropriate for the size of fabricated layers and the default fill with in our experiments. A point is deemed to represent a sharp turn if the smaller of its associated area coverage for curvature estimation is less than 30% of the circle area. In Table 1, we report the percentages of points deemed as sharp turns. It is quite evident that the number of sharp turns produced by CFS is much lower than that of zigzags and it is more comparable to, but generally still lower than, that of contour-parallel fills. On the other hand, the latter exhibits high contour plurality.

We report running times of our algorithm in Table 2 for a partial list of shapes in Figure 8; other relevant statistics are provided as well. Currently, the spiral construction and connection algorithm is implemented in C++ while the curve optimization phase is implemented in MATLAB. All of the above times are measured on an Intel® Core™ i7-6700 CPU 4.0GHz with 16GB RAM.

Under- and over-fill. Under- and over-fills occur as a result of non-uniform spacing between curve segments along a tool path. Since it is difficult to measure the extent of these fill artifacts for real prints, we provide an estimate by thickening a computed tool path at its expected fill width and measure the intersection and gap after the process. Figure 9 visualizes the over- and under-fills for one CFS path before and after path optimization. Figure 10 (top) compares the amount of under- and over-fills over several shapes.

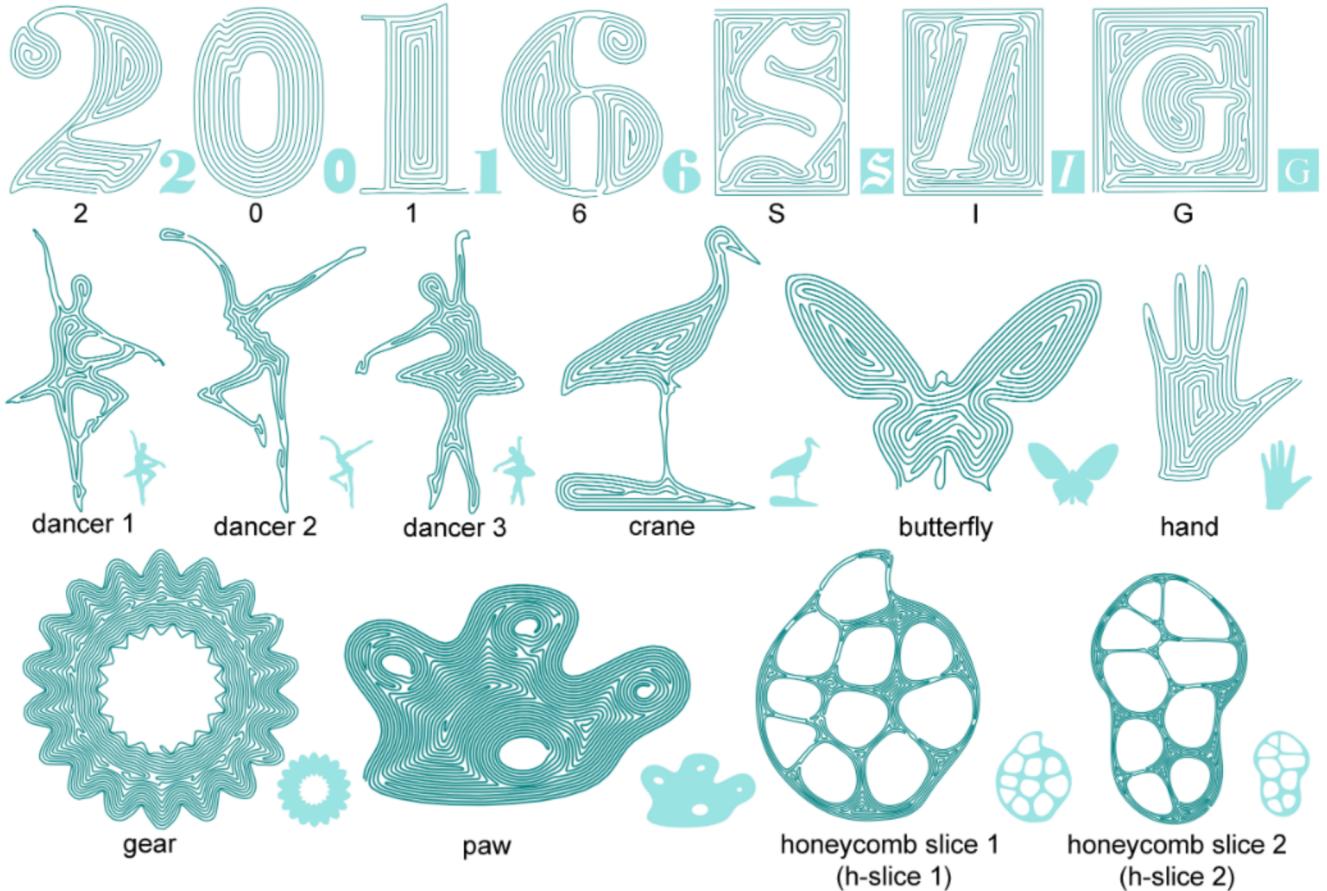


Figure 8: A gallery of continuous CFS tool paths generated by our algorithm. Shown as insets, the input shapes, both synthetic and from slices of fabricated 3D objects (see the honeycomb slices in last row as well as in Figure 1), exhibit varying degrees of complexity in terms of convexity/concavity of boundaries and hollowness. The top two rows show lower-resolution results for ease of visualization. The bottom row shows higher-resolution results, which are closer to that of real fabrication and also demonstrate robustness of our method.

Input	#P	#R	CFSt (s)	OPt (s)	Total (s)
dancer 1	4	31	0.25	1.676	1.926
dancer 2	6	27	0.297	1.59	1.887
dancer 3	4	33	0.203	7.085	7.288
crane	2	42	0.125	1.917	2.042
butterfly	4	51	0.359	4.479	4.838
hand	1	30	0.125	7.277	7.402
gear	19	143	0.766	8.978	9.744
paw	8	147	0.813	9.429	10.242
h-slice 1	22	148	0.834	7.092	7.926
h-slice 2	22	145	0.95	7.412	8.362

Table 2: Some statistics and running times for our CFS tool path generation algorithm. We report the number of pockets (#P) and the number of rerouting points (#R) of the tool paths. For running times, we report time needed for rerouting to generate the initial connected Fermat spirals (CFSt) and time for curve optimization (OPt), as well as the total. All running times are in seconds.

As one would expect, smoothing tends to increase gaps and under-fills, especially near sharp corners (e.g., four corners of the rectangular ‘G’ and many corners of the gear model). Overall, our curve optimization (smoothing plus spacing) tends to increase under-fills and reduce over-fills; see Figure 10. As shown in Figure 9, the

spacing term is seen to effectively remove or at least, more evenly distribute, severe over-fills of an un-optimized path.

The inability of our current curve optimization scheme to fill all the gaps can be attributed to limited curve movements. For example, curves cannot be elongated to alleviate under-fills. Near sharp corners and turns, there is a *trade-off* between curve fairness and gap size. Since the gaps are typically few and far inbetween, sacrificing fairness at few places to fill the gaps, e.g., by elongating the curve locally, is possible; we leave this for future work.

Visual quality. Figure 11 shows photos taken of four 2D layer shapes (the ‘S’, gear, and the two honeycomb slices from Figure 8) fabricated using the three fill patterns. Figure 10 (bottom) plots the estimated under- and over-fills over the four shapes.

Visually, we observe that zigzag incurs little under-fill and can generally maintain even material distribution along straight tool paths. However, fill quality degrades near region boundaries, showing both roughness and “aliasing” artifacts. The latter shows up near boundaries which are close to being parallel to, but are not parallel to, the scan direction (see the ‘S’ example). Since the zigzag fill is not globally continuous, fill artifacts also occur over areas where separate zigzag-filled segments join. In terms of estimated over-fills, as shown in Figure 10 (bottom left), zigzag incurs a larger amount than its counterparts since paths generated by Slic3r next

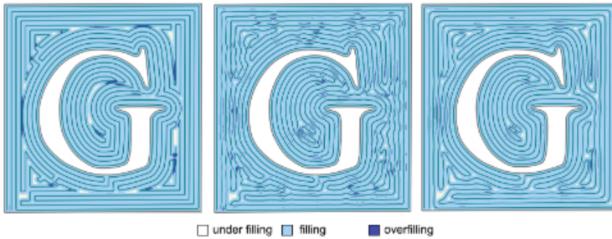


Figure 9: Estimated under- and over-fills visualized for a CFS tool path. Left: before path optimization. Middle: path optimized with even spacing only. Right: optimization with smoothing and even spacing, our default post-processing scheme, attenuates severe over-fills (dark blue spots in left), reduces total over-fills in most cases, but may introduce more gaps, especially near sharp corners (the four corners of the plate) due to curve smoothing.

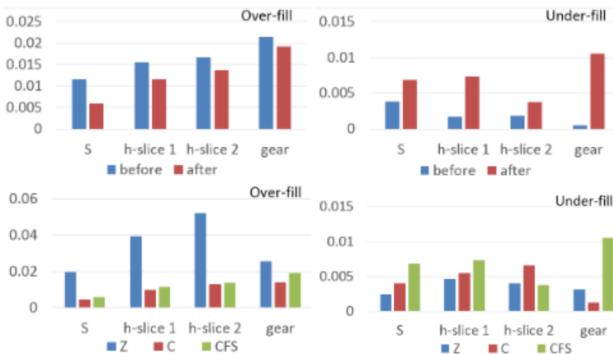


Figure 10: Estimated over- and under-fill rate comparisons on the four shapes from Figure 11. Top: Before and after curve optimization. Bottom: comparison between the three fill methods zigzag (Z), contour-parallel (C), and CFS.

to region boundaries have a distance smaller than $\frac{1}{2}w$ to the boundaries, leading to excessive over-fills along these paths.

For contour-parallel fills, visible artifacts (under- or over-fills) occur near the center of pockets and between adjacent, but separately contoured regions. In contrast, fabrication resulting from CFS appear to exhibit better overall quality with less visible artifacts. However, CFS appears to lead to relatively large amount of under-fills due to curve smoothing; see Figure 10 (bottom right). Of course, one should bear in mind that since 3D printing is a physical process, random device imprecisions which may cause visible artifacts in the filled layers are possible.

Figure 12 examines the surface qualify of a 3D object fabricated using our FDM printer, contrasting CFS fills to zigzag fills. The object is formed by a 50-fold vertical extrusion of the gear layer from Figure 8; the final cylinder is 1cm tall. There are visible gaps from a top view of the CFS fills, due to path smoothing as we discussed. On the other hand, our tool path optimization effectively distributes the (relatively large) total amount of under-fills over a large number of spots so that most individual gaps are small and can be filled by melting of the filament material. From the side views, CFS fills lead to smoother boundaries, while surface roughness arising from zigzag fills is evident. However, the latter is typically corrected by external contouring, but at the expense of under-fills between the contoured exterior and boundaries of the interior zigzag fills.

Fabrication time. Figure 13 compares fabrication times recorded on the RepRap Prusa 3D printer for the three fill methods. We ob-

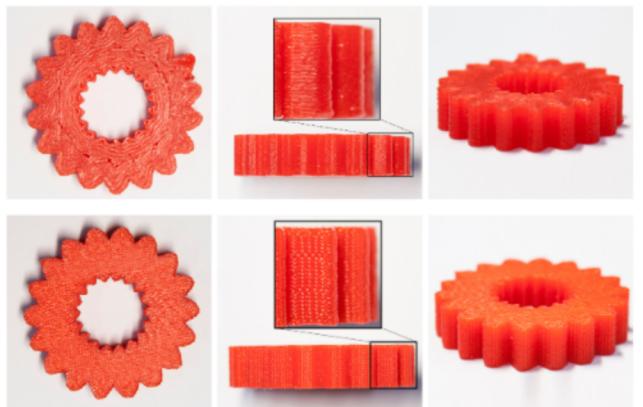


Figure 12: Photographs of 3D fabrications using FDM. The object is an extruded cylinder from the gear layer of Figure 8). A few views and closeups are shown to reveal surface quality. Top: using CFS fills. Bottom: the same gear cylinder fabricated with zigzag.

serve that while the fabrication speed for CFS tool paths is generally more favorable than their counterparts on an FDM printer, the speed gains vary. For more complex layers, e.g., the honeycomb slices, the speed gains tend to be more significant.

Comparison to evolved labyrinths. In Figure 14, we compare our connected Fermat spirals to a result from stochastic curve evolution by Pederson and Singh [2006]. The results become more visually comparable if the evolution is rewarded by better alignment of the curves with the boundary of the input shape, as shown. However, the curve evolution performs inward erosions and as such, it is unlikely to maintain a fair and outline-conforming exterior path as our spiral approach. Moreover, the local stochastic movements are likely to result in more sharp turns throughout.

6 Conclusion, limitation, and future work

We present a region fill algorithm using connected Fermat spirals, achieving *global continuity*. Our algorithm extends the use of spirals as space-filling curves from regular convex shapes to non-convex shapes, even shapes with many interior holes. Our contributions are two-fold. At a conceptual level, we introduce the use of Fermat spirals to the construction of a new kind of space-filling patterns. The construction reflects compelling properties of Fermat spirals. The use of Fermat spirals prevents the curve from being locked in pockets. Furthermore, the freedom allowed in choosing start and end points along the boundary of a Fermat spiral facilitates a scheme which systematically joins a set of Fermat spirals. Practically, the new curves possess appealing properties for tool path planning in the context of layered fabrication.

In retrospect, connected Fermat spirals are not necessarily suitable for all layer shapes. Compared to their counterparts, it appears that they excel at filling layers with complex geometry, especially those with many holes, to achieve higher build quality both inside and on the exterior. If one were to print a 3D object with honeycomb interiors [Lu et al. 2014], a sensible plan would be to print the middle slices using our CFS fills while topping off the print, where the layer shapes are likely to be convex or spirallable, with zigzag or hybrid fills, possibly alternating between them.

As discussed previously, the connected Fermat spirals are not guaranteed to be truly space-filling. They also lack the regularities and mathematical rigors possessed by Peano or Hilbert curves; the definition of connected Fermat spirals is constructive and not con-

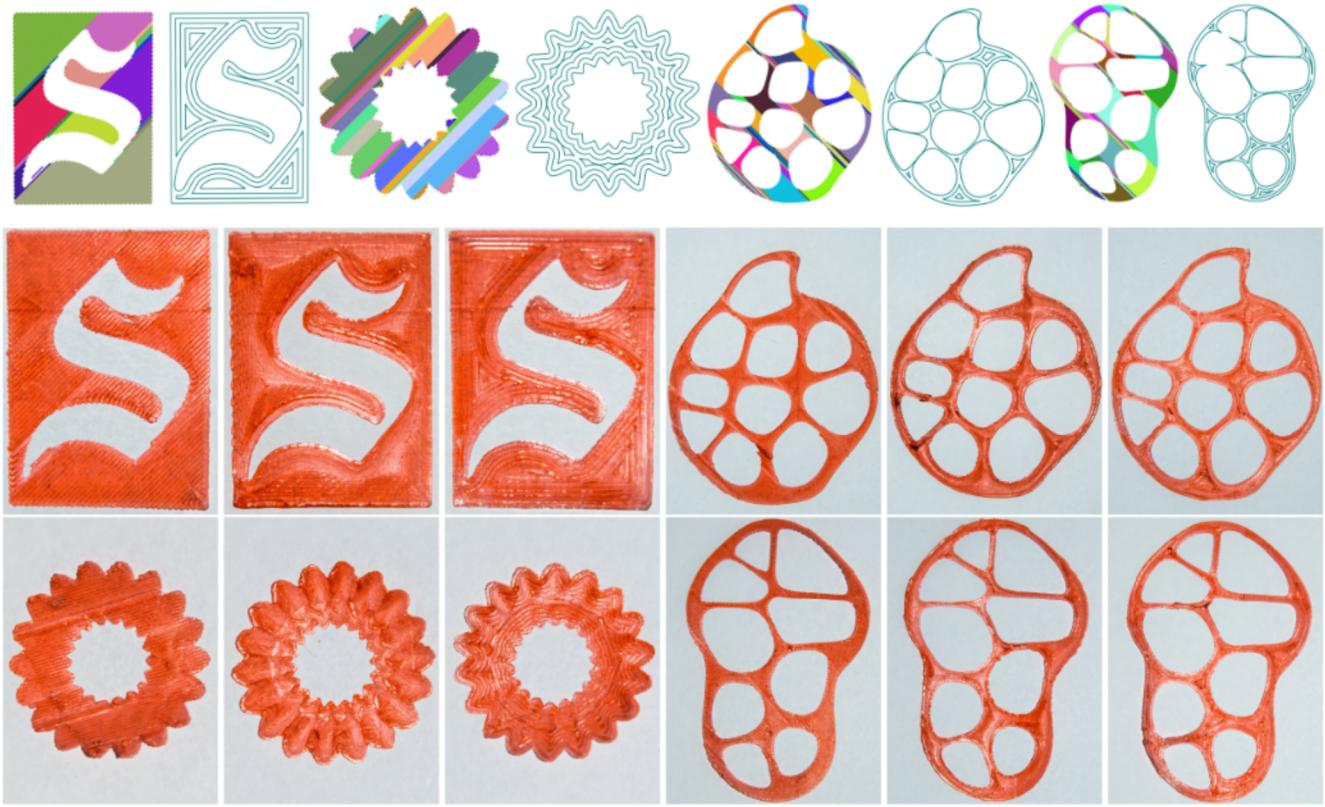


Figure 11: Photographs of fabricated layers using three fill methods (bottom two rows). In each of the four groups, the left result is from zigzag, the middle from contour-parallel, and the last from CFS. Top row shows schematic displays of the zigzag and contour-parallel fill patterns; for zigzag, we color the disconnected fill segments. The CFS fill patterns for the four shapes can be found in Figure 8.

ceptual. Our current algorithm generally results in less number of sharp turns compared to zigzag. However, it makes no attempt to minimize them. The local curve optimization scheme also leaves room for improvements. In particular, current curve displacements cannot “slide” adjacent segments against each other or elongate the curve to fill gaps. These operations are possible by adding attraction-repulsion forces as in the curve evolution scheme of Pedersen and Singh [2006]; we leave this for future work.

The amount of gains afforded by our new tool paths for layered fabrication is dictated by the mechanics of the motor controllers of the 3D printers. Contemporary, low-end printers rely on simple motor controls, approximating a smooth curve by piecewise linear segments. One may regard such a mechanism as catering to zigzagging tool paths. This may also be accounted for as a limitation of our approach, as we seek low-curvature but non-straight tool paths and do not take advantage of the control mechanisms of these low-end printers. On the other hand, it is possible to incorporate more sophisticated look-ahead and adaptive speed control algorithms to achieve higher motor speed for low-curvature but non-straight tool paths [Wang and Cao 2012]. As well, higher-end printers with more sophisticated controllers, like those of current Computer Numerical Control (CNC) machines, can also achieve a higher motor speed for tool path with smaller curvatures [Wang et al. 2010]. With such controllers, Fermat spirals would incur a significant speed-up.

In the future, we would like to investigate the interplay between fill patterns of consecutive layers. For better strength of FDM prints, consecutive layers should not be fabricated with close-to-identical fill patterns. Our construction scheme may be slightly perturbed so that the Fermat spiral fills of consecutive layers may interweave.

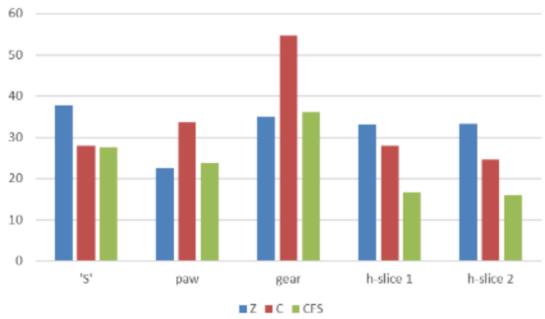


Figure 13: Real 3D printing times (in seconds) for fabricating several layer shapes, from the relatively simple ‘S’ to the more complex honeycomb slices from Figure 8, using CFS (orange), contour-parallel (C: cyan), and zigzag (Z: purple) tool path fills.

Another inter-layer optimization to consider is with respect to the start and end points of each layer, in term of increasing the coherence and avoiding redundant moves of the nozzle from one layer to the next. Finally, it would be interesting to re-examine optimization problems involving object orientation or decomposition while taking into account how the resulting 2D slices are filled.

Acknowledgments

We would first like to thank all the reviewers for their valuable comments and suggestions. The development of our problem formula-



Figure 14: Our CFS fill (left) vs. a result from stochastic curve evolution [Pedersen and Singh 2006], which is copied from Figure 16 of the paper. It is important to note that the outer boundary (right) is the input contour and not part of output of the curve evolution. The evolution results in the interior curve structure.

tion was critically influenced by discussions with Dan Halperin, to whom we are grateful. Thanks also go to Yang (Sherry) Liu (illustration), Shenqi Liu (3D printing), Sha (Ada) He (video), and the rest of the SDU lab-mates for their tremendous help on production. This work is supported in part by grants from National 973 Program (2015CB352501), NSFC (61232011, 61202147, 61332015, 61572291), NSERC Canada (No. 611370), Israeli Science Foundation (No. 1790/12), U.S.-Israel Bi-National Science Foundation (No. 2012376), and gift awards from Adobe and Intel.

References

- ARKINA, E. M., FEKETEB, S. P., AND MITCHELL, J. S. 2000. Approximation algorithms for lawn mowing and milling. *Computational Geometry* 17, 1-2, 25–50.
- CHEN, X., ZHANG, H., LIN, J., HU, R., LU, L., HUANG, Q., BENES, B., COHEN-OR, D., AND CHEN, B. 2015. Dapper: Decompose-and-pack for 3D Printing. *ACM Trans. on Graph* 34, 6, 213:1–213:12.
- DAFNER, R., COHEN-OR, D., AND MATIAS, Y. 2000. Context-based Space Filling Curves. *Computer Graphics Forum* 19, 3, 209–218.
- DING, D., PAN, Z. S., CUIURI, D., AND LI, H. 2014. A tool-path generation strategy for wire and arc additive manufacturing. *Int. J. of Adv. Manufact. Tech.* 73, 1-4, 173–183.
- DINH, H. Q., GELMAN, F., LEFEBVRE, S., AND CLAUX, F. 2015. Modeling and Toolpath Generation for Consumer-level 3D Printing. In *ACM SIGGRAPH Courses*, 17:1–17:273.
- DWIVEDI, R., AND KOVACEVIC, R. 2004. Automated torch path planning using polygon subdivision for solid freeform fabrication based on welding. *J. of Manufact. Sys.* 23, 4, 278–291.
- EL-MIDANY, T. T., ELKERAN, A., AND TAWFIK, H. 1993. Tool path pattern comparison: contour-parallel with direction-parallel. In *Geom. model. and imaging — new trends*, 77–82.
- GIBSON, I., ROSEN, D., AND STUCKER, B. 2015. *Additive Manufacturing Technologies*, 2nd ed. Springer.
- HELD, M., AND SPIELBERGER, C. 2014. Improved spiral high-speed machining of multiply-connected pockets. *Computer-Aided Design and Applications* 11, 3, 346–357.
- HILDEBRAND, K., BICKEL, B., AND ALEXA, M. 2013. Orthogonal slicing for additive manufacturing. *Computers & Graphics* 37, 6, 669 – 675. Shape Model. Intl. (SMI) Conf.
- HU, R., LI, H., ZHANG, H., AND COHEN-OR, D. 2014. Approximate pyramidal shape decomposition. *ACM Trans. on Graph* 33, 6, 213:1–213:12.
- JIN, G., LI, W., AND GAO, L. 2013. An adaptive process planning approach of rapid prototyping and manufacturing. *Robotics and Computer-Integrated Manufacturing* 29, 1, 23–38.
- JIN, Y. A., HE, Y., FU, J. Z., GAN, W. F., AND LIN, Z. W. 2014. Optimization of tool-path generation for material extrusion-based additive manufacturing technology. *Additive Manufacturing* 1, 4, 32–47.
- JOHNSON, A. 2015. Clipper - an open source freeware library for clipping and offsetting lines and polygons. <http://www.angusj.com/delphi/clipper.php>.
- KULKARNI, P., MARSAN, A., AND DUTTA, D. 2000. A review of process planning techniques in layered manufacturing. *Rapid Prototyping Journal* 6, 18–35.
- LU, L., SHARF, A., ZHAO, H., WEI, Y., FAN, Q., CHEN, X., SAVOYE, Y., TU, C., COHEN-OR, D., AND CHEN, B. 2014. Build-to-last: Strength to weight 3D printed objects. *ACM Trans. on Graph* 33, 4, 97:1–97:10.
- LUO, L., BARAN, I., RUSINKIEWICZ, S., AND MATUSIK, W. 2012. Chopper: Partitioning models into 3D-printable parts. *ACM Trans. on Graph* 31, 6, 129:1–129:9.
- PEDERSEN, H., AND SINGH, K. 2006. Organic labyrinths and mazes. In *Proc. of Non-photorealistic Animation and Rendering (NPAR)*, 79–86.
- POTTMANN, H., WALLNER, J., HUANG, Q.-X., AND YANG, Y.-L. 2009. Integral invariants for robust geometry processing. *Comp. Aided Geom. Design* 26, 1, 37–60.
- PRÉVOST, R., WHITING, E., LEFEBVRE, S., AND SORKINE-HORNUNG, O. 2013. Make It Stand: Balancing shapes for 3D fabrication. *ACM Trans. on Graph* 32, 4, 81:1–81:10.
- REN, F., SUN, Y., AND GUO, D. 2009. Combined reparameterization-based spiral toolpath generation for five-axis sculptured surface machining. *Int. J. of Adv. Manufac. Tech.* 40, 7, 760–768.
- SLIC3R, 2016. Slic3r. <http://slic3r.org/>.
- STAVA, O., VANEK, J., BENES, B., CARR, N., AND MĚCH, R. 2012. Stress relief: improving structural strength of 3D printable objects. *ACM Trans. on Graph* 31, 4, 48:1–48:11.
- VANEK, J., GARCIA, J., BENES, B., MECH, R., CARR, N., STAVA, O., AND MILLER, G. 2014. PackMerger: A 3D print volume optimizer. *Computer Graphics Forum* 33, 6, 322–332.
- VANEK, J., GALICIA, J. A. G., AND BENES, B. 2014. Clever support: Efficient support structure generation for digital fabrication. *Computer Graphics Forum* 33, 5, 117–125.
- WANG, L., AND CAO, J. 2012. A look-ahead and adaptive speed control algorithm for high-speed CNC equipment. *Int. J. of Adv. Manufact. Techn.* 63, 705–717.
- WANG, L., CAO, J. F., AND LI, Y. Q. 2010. Speed optimization control method of smooth motion for high-speed CNC machine tools. *Int. J. of Adv. Manufact. Tech.* 49, 313–325.
- WASSER, T., JAYAL, A. D., AND PISTOR, C. 1999. Implementation and evaluation of novel buildstyles in fused deposition modeling. *Strain* 5, 6, 95–102.

- WIKIPEDIA, 2015. Fermat's spiral — wikipedia, the free encyclopedia. [Online; accessed 28-November-2015].
- WIKIPEDIA, 2016. Labyrinth — wikipedia, the free encyclopedia. [Online; accessed 10-April-2016].
- YANG, Y., LOH, H., FUH, J., AND WANG, Y. 2002. Equidistant path generation for improving scanning efficiency in layered manufacturing. *Rapid Prototyping Journal* 8, 1, 30–37.
- YAO, M., CHEN, Z., LUO, L., WANG, R., AND WANG, H. 2015. Level-set-based partitioning and packing optimization of a printable model. *ACM Trans. on Graph* 34, 6, 214:1–214:11.
- ZHANG, X., LE, X., PANOTOPOULOU, A., WHITING, E., AND WANG, C. C. L. 2015. Perceptual Models of Preference in 3D Printing Direction. *ACM Trans. Graph.* 34, 6, 215:1–215:12.

Appendix: Details on Curve Optimization

Problem formulation. To formulate the objective function for curve optimization, we discretize the input curve via curvature-based sampling so that we place more samples around high-curvature sections. Let $\mathbf{p}_1^0, \dots, \mathbf{p}_N^0$ denote the set of samples in a sequential order. Our goal is to perturb these points so that the path becomes smoother while it preserves a prescribed spacing constraint, i.e., the distance between neighboring curve segments shall be fixed at d (the pre-defined patch spacing) as much as possible.

In our formulation, we consider three objectives. The first penalizes the *magnitude of the perturbations* and it is formulated in the least square sense, $f_{\text{regu}} = \sum_{i=1}^N |\mathbf{p}_i \mathbf{p}_i^0|^2$, where $\mathbf{p}_1, \dots, \mathbf{p}_N$ are sample points and $|\mathbf{p}_i \mathbf{p}_i^0|$ denotes edge length.

The second objective term is the *smoothness potential*. We modify the standard mid-point scheme (e.g., $\mathbf{p}_i + \mathbf{p}_{i+2} \approx 2\mathbf{p}_{i+1}$) so that it reflects the varying density. Specifically, we define

$$u_i = |\mathbf{p}_{i+1}^0 \mathbf{p}_i^0| / (|\mathbf{p}_{i+1}^0 \mathbf{p}_i^0| + |\mathbf{p}_{i+2}^0 \mathbf{p}_{i+1}^0|),$$

and formulate the smoothness potential as

$$f_{\text{smooth}} = \sum_{i=1}^{N-2} \|(1-u_i)\mathbf{p}_i + u_i \mathbf{p}_{i+2} - \mathbf{p}_{i+1}\|^2.$$

The third objective term maintains the *spacing after smoothing*. To this end, we compute for each point \mathbf{p}_i the closest points (or footpoints) on the adjacent curve segments. There are two cases that can occur for the footpoints. In the first case, they can lie on edges and for each such edge $\mathbf{p}_j \mathbf{p}_{j+1}$, the footpoint is given by

$$\begin{aligned} \mathbf{f}_{i,j} &= (1-t_{i,j})\mathbf{p}_j + t_{i,j}\mathbf{p}_{j+1}, \\ t_{i,j} &= (\mathbf{p}_i - \mathbf{p}_j)^T(\mathbf{p}_j - \mathbf{p}_{j+1}) / |\mathbf{p}_j \mathbf{p}_{j+1}|^2, \end{aligned} \quad (1)$$

with $0 \leq t_{i,j} \leq 1$. In the second case, they can lie on vertices and for each such vertex \mathbf{p}_j , it satisfies $t_{i,j} \leq 0$ and $t_{i,j-1} \geq 1$. To find the footpoints that lie on adjacent curve segments, we pick footpoints $\mathbf{f}_{i,j}$ (or \mathbf{p}_j) so that $\mathbf{p}_i \mathbf{f}_{i,j}$ (or $\mathbf{p}_i \mathbf{p}_j$) does not intersect with



Figure 15: An edge footpoint (left) and a vertex footpoint (right).

the existing curve. Let $\mathcal{E} = \{(\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_{j+1})\}$ and $\mathcal{V} = \{\mathbf{p}_i, \mathbf{p}_j\}$ collect all triplets that specify edge footpoints and vertex footpoints, respectively. We formulate the spacing-preserving term as

$$f_{\text{space}} = \sum_{(\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_{j+1}) \in \mathcal{E}} (|\mathbf{p}_i \mathbf{f}_{i,j}| - d)^2 + \sum_{(\mathbf{p}_i, \mathbf{p}_j) \in \mathcal{V}} (|\mathbf{p}_i \mathbf{p}_j| - d)^2,$$

where $\mathbf{f}_{i,j}$ is as defined in (1).

Combining the three terms into a weight sum, we arrive at the following optimization for spacing-preserving path smoothing:

$$\underset{\mathbf{p}_1, \dots, \mathbf{p}_N}{\text{minimize}} \quad f_{\text{regu}} + \alpha f_{\text{smooth}} + \beta f_{\text{space}}. \quad (2)$$

In our experiments, we set $\alpha = 200$ and $\beta = 1.0$.

Optimization. The objective function (2) includes both a discrete component, i.e., the footpoint configuration \mathcal{E}, \mathcal{V} , and a continuous aspect, i.e., a non-linear least squares in terms of the curve vertex positions. We propose to alternate between updating both. When the curve vertex positions are fixed, we compute the footpoints using the procedure described above. When the footpoint configuration is fixed, we apply Gauss-Newton method to optimize (2) due to its non-linear least square nature.

Gauss-Newton optimization seeks a local displacement of each vertex $\bar{\mathbf{p}}_i = \mathbf{p}_i + \mathbf{d}_i$ to optimize (2). f_{regu} and f_{smooth} are quadratic in vertex positions, so we keep their expressions in the optimization. Both $|\mathbf{p}_i \mathbf{f}_{i,j}|$ and $|\mathbf{p}_i \mathbf{p}_j|$ are non-linear, and we approximate them via linear approximations derived from computing derivatives. Denote $\mathbf{e}_j = \mathbf{p}_j - \mathbf{p}_{j+1}$, then $|\mathbf{p}_i \mathbf{f}_{i,j}|$ is approximated as:

$$|\bar{\mathbf{p}}_i \bar{\mathbf{f}}_{i,j}| \approx |\mathbf{p}_i \mathbf{f}_{i,j}| + \mathbf{g}_{ij1}^T \mathbf{d}_i + \mathbf{g}_{ij2}^T \mathbf{d}_j + \mathbf{g}_{ij3}^T \mathbf{d}_{j+1}, \quad (3)$$

where \mathbf{g}_{ij1} , \mathbf{g}_{ij2} , and \mathbf{g}_{ij3} are defined as,

$$\begin{aligned} \mathbf{g}_{ij1} &= \frac{\mathbf{p}_i - \mathbf{f}_{i,j}}{|\mathbf{p}_i \mathbf{f}_{i,j}|}, \quad \mathbf{g}_{ij2} = -(1-t_{i,j})\mathbf{g}_{ij1} - \mathbf{g}_{ij1}^T \mathbf{e}_j \frac{\partial t_{i,j}}{\partial \mathbf{p}_j}, \\ \mathbf{g}_{ij3} &= -t_{i,j}\mathbf{g}_{ij1} + t_{i,j}\mathbf{g}_{ij1}^T \mathbf{e}_j \frac{\partial t_{i,j}}{\partial \mathbf{p}_{j+1}}. \end{aligned} \quad (4)$$

The second term $|\bar{\mathbf{p}}_i \bar{\mathbf{p}}_j|$ is approximated by

$$|\bar{\mathbf{p}}_i \bar{\mathbf{p}}_j| \approx |\mathbf{p}_i \mathbf{p}_j| + \mathbf{g}_{ij}^T (\mathbf{d}_i - \mathbf{d}_j), \quad (5)$$

where $\mathbf{g}_{ij} = (\mathbf{p}_i - \mathbf{p}_j) / |\mathbf{p}_i \mathbf{p}_j|$. Substituting (3) and (5) into (2), we arrive at the following least squares for optimizing $\mathbf{d}_1, \dots, \mathbf{d}_N$:

$$\begin{aligned} &\sum_{i=1}^N |\mathbf{d}_i \mathbf{r}_i|^2 + \alpha \sum_{i=1}^{N-2} \|(1-u_i)\mathbf{d}_i + u_i \mathbf{d}_{i+2} - \mathbf{d}_{i+1} - \mathbf{r}'_i\|^2 \\ &+ \beta \sum_{(\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_{j+1}) \in \mathcal{E}} (\mathbf{g}_{ij1}^T \mathbf{d}_i + \mathbf{g}_{ij2}^T \mathbf{d}_j + \mathbf{g}_{ij3}^T \mathbf{d}_{j+1} - r_{ij})^2 \\ &+ \sum_{(\mathbf{p}_i, \mathbf{p}_j) \in \mathcal{V}} (|\mathbf{p}_i \mathbf{p}_j| + \mathbf{g}_{ij}^T (\mathbf{d}_i - \mathbf{d}_j) - d)^2, \end{aligned} \quad (6)$$

where $\mathbf{r}_i = \mathbf{p}_i^0 - \mathbf{p}_i$, $\mathbf{r}'_i = \mathbf{p}_{i+1} - (1-u_i)\mathbf{p}_i - u_i \mathbf{p}_{i+2}$, $r_{ij} = d - |\mathbf{p}_i \mathbf{f}_{i,j}|$, and $r'_{ij} = d - |\mathbf{p}_i \mathbf{p}_j|$.

We minimize (6) by solving the induced linear system. Let \mathbf{d}_i^* be the optimal displacement, the vertex at the next iteration is updated as $\mathbf{p}_i \leftarrow \mathbf{p}_i + \mathbf{d}_i^*$. We terminate Gauss-Newton optimization when $\max_{1 \leq i \leq N} \|\mathbf{d}_i\| < 10^{-6}$. Once we have optimized the vertex positions, we proceed to recompute the footpoints. The procedure alternates until the maximum displacement on curve positions is below 10^{-5} . In our experiments, we have found that 4-8 iterations are sufficient for convergence in each phase.