

# Clean color: Improving multi-filament 3D prints

Jean Hergel<sup>1,2</sup> and Sylvain Lefebvre<sup>1,2</sup>

<sup>1</sup>Inria

<sup>2</sup>Université de Lorraine

---

## Abstract

*Fused Filament Fabrication is an additive manufacturing process by which a 3D object is created from plastic filament. The filament is pushed through a hot nozzle where it melts. The nozzle deposits plastic layer after layer to create the final object. This process has been popularized by the RepRap community.*

*Several printers feature multiple extruders, allowing objects to be formed from multiple materials or colors. The extruders are mounted side by side on the printer carriage. However, the print quality suffers when objects with color patterns are printed – a disappointment for designers interested in 3D printing their colored digital models. The most severe issue is the oozing of plastic from the idle extruders: Plastics of different colors bleed onto each other giving the surface a smudged aspect, excess strings oozing from the extruder deposit on the surface, and holes appear due to this missing plastic. Fixing this issue is difficult: increasing the printing speed reduces oozing but also degrades surface quality – on large prints the required speed level become impractical. Adding a physical mechanism increases cost and print time as extruders travel to a cleaning station.*

*Instead, we rely on software and exploit degrees of freedom of the printing process. We introduce three techniques that complement each other in improving the print quality significantly. We first reduce the impact of oozing plastic by choosing a better azimuth angle for the printed part. We build a disposable rampart in close proximity of the part, giving the extruders the opportunity to wipe oozing strings and refill with hot plastic. We finally introduce a toolpath planner avoiding and hiding most of the defects due to oozing, and seamlessly integrating the rampart.*

*We demonstrate our technique on several challenging multiple color prints, and show that our tool path planner improves the surface finish of single color prints as well.*

Categories and Subject Descriptors (according to ACM CCS): I.3.0 [Computer Graphics]: General—

---

## 1. Introduction

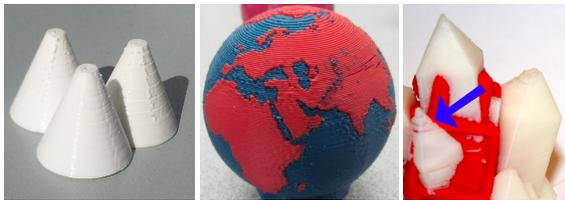
3D printing holds promise for CG artists and home users to print colorful, interesting designs. Unfortunately, full color printing is currently limited to high-end printers such as the ZCorp 450 or the MCor IRIS printers. Entry level printers using plastic filament allow to explore the possibility to print with different colors. These printers are largely based on the Fused Deposition Modeling (FDM) technology, pioneered by Stratasys and popularized by the RepRap community and low-cost manufacturers like MakerBot or Ultimaker. The printers operate by adding layer after layer of material to form an object – a concept known as *additive manufacturing*. A plastic filament is pushed through a heated nozzle. The thin thread of melted plastic that exits the tip of the nozzle immediately fuses with the layer below, creating

a solid object. Typical layer thickness ranges from 0.1 to 0.3 millimeters. The block formed by the stepper motor pushing the filament and the heated nozzle is called an *extruder*.

The main approach for FDM color printing consist of using multiple extruders. The *Replicator 1 Dual* (Makerbot) and its successor the *Replicator 2X* made this concept available to a large audience. Both feature two extruders moving together on a single carriage. *CubeX Trio* (3D Systems), *Tricolour Mendel* (RepRapPro) – three extruders – and *Ultimaker* (Ultimaker) are other printers of similar design. A key advantage of this approach is its simplicity and low cost: the extruders are mounted side by side on a single moving carriage and print one after the other, forming layers with different colors.

Unfortunately the print quality is significantly down-

graded when printing with multiple extruders. The major issues are shown Figure 1. *Zippers* are created when the extruder stops printing a path to travel to a different location. Zippers occur on both single-color and dual-color prints (Figure 1, left). *Color smears and strings* occur when a plastic of the first color is accidentally deposited on a region of a different color, due to plastic oozing from an extruder by gravity (see Figure 8). These flaws are especially noticeable on prints relying on colors with high contrast (Figure 1, middle). On prints with translucent plastic, smears can be seen through several layers of material. *Holes* occur after a smear: the plastic that oozed from an extruder will not be deposited where it is needed next (Figure 1, right). There is no easy way to avoid this issue, as it is difficult to precisely predict how much plastic disappeared due to oozing. The occurrence and severity of these fabrication defects depend of the type of plastic used for printing. In this paper we focus on the use of ABS plastic – one of the most common thermoplastics.



**Figure 1:** Typical defects in single and dual color prints. Left: The surface of the cones is not smooth due to zippers. Middle: Picture from *Two-color-world* by user *m6mafia* printed by *SJAmakers* on *Thingiverse*. Color smears appear throughout Eurasia due to the idle extruder oozing plastic. Right: Holes appear due to missing plastic after oozing.

**Contributions.** We propose a novel approach for improving the quality of multi-color prints. Our approach is implemented by means of software only: we do not require any change to the printers themselves. Since some fabrication defects are unavoidable, we strive to hide them in low visibility regions of the object.

Building upon our analysis of print defects (Section 3) we propose three novel components working together to minimize visible fabrication defects:

- An algorithm to find the best azimuth angle for printing the object: the angle that minimizes the time during which an extruder overlaps an area of a different color.
- A method to build a double-walled rampart around the object, allowing extruders to be wiped and refilled with plastic, thereby reducing color smears and holes.
- A path planning algorithm optimizing navigation in between printing paths. The planner reduces the visual impact of color smears and zippers by avoiding or hiding defects whenever possible. It further exploits the rampart to keep the extruders clean and ready.

Our approach sacrifices time and material to improve the

quality of printed parts: the rampart is disposable and our path planner takes more time to carefully navigate around the part. However, this is compensated by a significant increase in print quality.

## 2. Prior work

### 2.1. Background on 3D printing

Our work focuses on 3D printing for Computer Graphics applications, but is rooted in FDM technology. Fused filament fabrication creates the object by adding layer after layer of melted plastic filament. To print an object, it is necessary to convert its mesh surface representation into a set of *tool-paths*. These paths describe the motion of the extruder, as well as how much filament has to be pushed through the hot nozzle. They may describe a *print path* along which plastic is deposited or a *travel path* where the extruder simply moves from one location to the next.

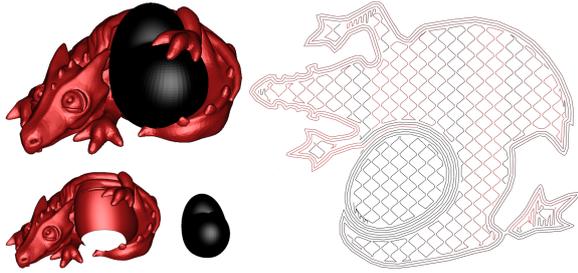
The first part of the process is the *slicing* stage. The input mesh is sliced into a number of layers, each slice capturing the contours defining the interior of the object (let us assume here that the mesh is watertight and non self-intersecting). There are many possible ways to slice in layers [PRD95]. In our software, we take the simple approach of slicing the object uniformly along its height – but our approach is orthogonal to the slicing process. We typically use 0.3 mm slices.

Print paths are extracted by offsetting the contours, taking into account the width of the printing nozzle. The first paths to be extracted are the *perimeters*. They form the outer, visible surface of the printed object. A number of additional print paths, *shells*, are extracted by offsetting the contours further inside. Perimeters and shells are cyclic paths in most cases. The interior is then covered by a set of *infill paths*. Figure 2 reveals the print paths of a slice. As shown in this same figure, multiple color models are generally specified as multiple meshes. Recent approaches enable procedural descriptions of the choice of material within the object [VWRKM13, Lef13].

In this work we assume the input meshes represent well formed 3D objects that can be 3D printed. Note that approaches have been proposed to reinforce 3D models before printing [SVB\*12, ZPZ13, WWY\*13].

### 2.2. Path planning

Several approaches improve the motion of the printing extruder, with the primary concern of minimizing the print time. This leads to a traveling salesman problem (TSP) formulation [WMJC02, CW04]. However, we found this formulation to not be well suited to our scenario (Section 6). Motion planning for multiple extruder systems has been investigated [CC06, CZ10]. However, contrary to our setting, printing heads can move independently and concurrently. The problem is thus significantly different. Prior work also considers hiding inside the part defects similar to zippers that



**Figure 2:** Left, top: Two-color dragon-heart model (Thingiverse, thing 17204, user andreas). Left, bottom: The model is made of two meshes partitioning space. Each volume is filled with plastic by a different extruder. Right: A slice of the model, showing perimeters (outermost contours), shells (inner contours) and infill patterns inside.

occur with selective laser sintering (SLS) [XCY11]. This is possible thanks to the optical nature of the laser motion. Extruders behave differently due to their physical motion: crossing over a printed path risks creating a defect. Nevertheless we follow a similar intuition and generalize this idea by taking into account the visibility of the surface.

### 2.3. Part orientation

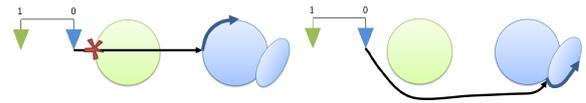
Part orientation, and in particular the layering direction, is chosen by considering several important factors: surface roughness [PRD95], volumetric error between the print and its digital model [MRI00], as well as several other factors such as build time, cost, or support structure minimization [FF95, CFN\*95, AAD98]. Therefore, our approach preserves the vertical orientation. However, it exploits the azimuth angle of the part to minimize the amount of color smears that can occur.

### 2.4. Fighting ooze

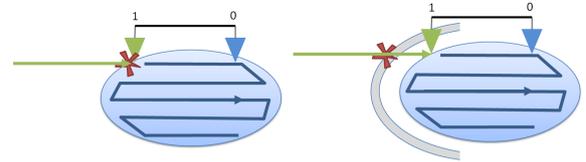
**Software.** Open source slicers such as *Skeinforge* typically fight ooze through retraction: after printing the filament is rolled back out of the hot plastic chamber – a move referred to as *deprime*. Before printing again the inverse motion – *prime* – is applied. The loss of pressure prevents plastic from oozing immediately. Prime/deprime effectively suppresses oozing along short travel moves, but ooze re-appears after a few seconds on idle extruders.

*RepRapPro* prints each color in sequence, lowering the temperature of the idle extruders to cool plastic down and stop oozing. However, this requires to pause the printing process at each color change. *KISSlicer* relies on a wipe pillar on the side of the print to clean the extruders. Travel to the wipe pillar takes time and may cross over the print. In a release concurrent to our work *Makerware 2.3* introduced walls to wipe the extruders clean of oozing filament. Section 7 includes a comparison.

**Physical devices.** A device obstructing the idle nozzle could



**Figure 3:** Case 1. Left: Extruder 0 (blue) travels across the green region, depositing ooze. Right: Our path planner carefully navigates around the part.



**Figure 4:** Case 2. Left: Extruder 0 (blue) prints an infill segment, which makes Extruder 1 (green) cross over the blue region, depositing ooze. Right: Our rampart catches the ooze before the part.

prevent ooze. This however increases the cost and the weight of the extruder carriage. High end printers use a wiper on the side of the print carriage to clean the oozing filament [PWT07]. This is not expensive, but the process implies frequent back and forth motion towards the cleaning station.

## 3. Overview

Our algorithm targets low-cost FDM printers of the RepRap family, equipped with multiple extruders. We focus our explanations on *dual printing* (two extruders), but our approach can be adapted to more extruders. The extruders are mounted on a single carriage at a fixed offset  $\delta$  (2D vector). We denote by  $\delta_i$  the offset of extruder  $i$ , with  $\delta_0 = 0$  and  $\delta_1 = \delta$ .

The input to the slicing process are two meshes, each describing the volume in space to be filled with the corresponding color/material. An example is given Figure 2. We name the meshes  $\mathcal{M}_0, \mathcal{M}_1$  and refer to the volume enclosed by each as respectively  $\mathcal{V}_0, \mathcal{V}_1$ . We assume non self-intersecting watertight meshes, with  $\mathcal{V}_0 \cap \mathcal{V}_1 = \emptyset$ .

To understand our approach, let us consider the major defect: strings of plastic deposited by oozing extruders. Strings are deposited in two circumstances, which we will refer to as *Case 1* and *Case 2* throughout the text:

- **Case 1** Travel moves (no extruder prints, all ooze). This is illustrated Figure 3, left.
- **Case 2** Print moves (one extruder prints, others ooze). This is illustrated Figure 4, left.

Case 1 is addressed by our path planner (Figure 3 right, Section 6). It navigates around the part while avoiding strings to deposit, finding travel paths where the idle extruder stays away from the print. If no such path can be found, our planner favors string deposition on matching colors. In last resort, it will always favor deposition in low visibility areas.



**Figure 5:** The left extruder uses black and the right extruder red. While the red (right) extruder prints the body of the dragon, some black (left) plastic will interfere with the red region. This is accurately predicted by the intersection volume (green). The printed model exhibits severe color smears and strings throughout this volume. Additional defects are due to poor path planning.

Case 2 is more constrained: the input printing paths cannot be changed. We address this in two ways. First, we minimize such cases through azimuth optimization (Section 4) and second, we create a rampart in close proximity of the shape, wiping extruders before they reach the surface (Figure 4 right, Section 5).

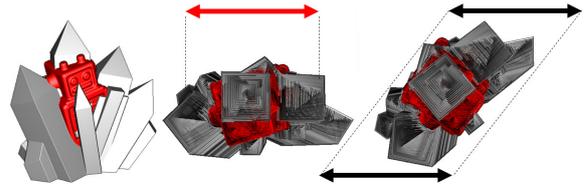
The planner and the rampart work in conjunction: the planner uses the rampart for circulation around the part. Each time an extruder crosses a rampart wall, it is wiped cleaned of any oozing string. In addition, we exploit the time during which the extruder circulates within the rampart to perform a *refill*: we run plastic through the extruder to refill the hot plastic chamber and keep it ready for printing. This prevents holes from appearing.

#### 4. Optimizing print azimuth

For clarity let us consider the case of two extruders separated along the X print axis. We refer to them as the *left* and *right* extruders. Whenever the right extruder prints, the left extruder creates a ghost image of the print shifted by  $\delta$  on the left. This ghost represents the locus of the points that may suffer color smears if the left extruder starts to ooze. Depending on the part size and azimuth, the ghost image will interfere (intersect) with what the right extruder is printing. Minimizing the size of this region will reduce the chance that smears occur. This is illustrated in Figure 5.

We perform the change of azimuth by a rotation around the Z-axis of angle  $\Theta$ , noted in matrix form as  $R_{\Theta}^Z$  in the following. We do not change the vertical orientation of the part (Section 2).

By reasoning throughout all the print layers, for a given orientation angle  $\Theta$  the total size of the interference region due to the left extruder is the volume  $I_L(\Theta) = (R_{\Theta}^Z \times \mathcal{V}_R) \cap (T_{\delta}^X \times R_{\Theta}^Z \times \mathcal{V}_R)$  where  $T_{\delta}^X$  is a translation matrix of  $\delta$  along the X-axis. Similarly, the interference region due to the right extruder is  $I_R(\Theta) = (R_{\Theta}^Z \times \mathcal{V}_L) \cap (T_{-\delta}^X \times R_{\Theta}^Z \times \mathcal{V}_L)$ .



**Figure 6:** Robot-ice (user Ultimaker on Youmagine): this model is larger than the spacing between the extruders, and its default azimuth results in significant color smears. Our azimuth optimization finds an angle avoiding these entirely.

We therefore search  $\Theta_{min}$  such that:

$$\Theta_{min} = \operatorname{argmin}_{\Theta} (I_R(\Theta) + I_L(\Theta))$$

In general computing the volume of the intersection between two meshes is a difficult problem. We rely on an approximate computation based on boolean mesh operations with dexels [VH86, WLC10]. This integrates very well within our slicer which also relies on this principle to extract slices [Lef13]. The best angle is selected after testing for all angles by 10 degree increments.

Figure 6 shows an example of optimized azimuth angle. A better azimuth angle strongly reduces the defects due to Case 2. However, they often cannot be entirely suppressed in particular on large parts. The rampart, described next, further reduces these defects.

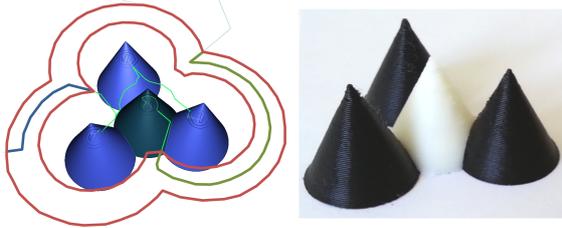
#### 5. Rampart

The rampart is a disposable structure built around the part. Its primary purpose is to catch the oozing strings before they reach the part (Case 2, Figure 4). Its secondary purpose is to provide a space where extruders can be refilled with plastic.

We construct the rampart in close proximity of the part. This allows to wipe the extruders clean as close as possible to the surface, and also keeps travel time low by avoiding traveling to a distant wipe station.

**Geometry:** The outline of the rampart is obtained by rendering the part as seen from above, in black on a white background. We process this silhouette image to only keep the connected component containing the image border (inner holes are removed). The remainder is then offset towards the outside to obtain the contour of the first wall of the rampart. Another offset gives the second wall. We use 2 mm between the part and the first wall, and 4 mm for the inner spacing. These contours are added to each layer at all heights, creating a vertical extrusion of the rampart contours.

**Printing:** The walls of the rampart are built at the start of each layer. Having the walls reach the tips of the extruders ensures proper wiping. We print the rampart at high speed (120 mm/sec), using a different extruder for each wall. One subtlety is that while a rampart wall is printed, the other extruder may move over the part – risking ooze to deposit. We



**Figure 7:** Left: One slice near the top of the 4-cones model. The red outlines are the walls of the rampart. Lime-green paths are travel moves. The bold paths within the rampart walls show where the extruders are active to refill with plastic. Right: Printed model, using our technique.



**Figure 8:** Left: The 4-cones model printing. The red arrow shows how much ooze exits the left (white) nozzle while the right (black) nozzle prints. Right: The rampart captured a significant amount of stringing. Result shown Figure 7.

have to ensure that it will be properly wiped before reaching the part. Our approach is simple: we start printing the wall from a location which guarantees that the other extruder reaches the rampart before the print. On our dual-extruder printer, when the right extruder is used we start printing the rampart from its leftmost point, and vice versa.

## 6. Path planning for multiple colors

The azimuth optimization and the rampart reduce defects due to Case 2, where strings are deposited by the idle extruder while the other prints. We now consider Case 1, where both extruders are idle and traveling.

The path planner chooses in which order the print paths are visited, the point where to start printing each, and the travel path for moving from a print path to the next. We design our planner under the worst case assumption that idle extruders are always oozing. Our strategy is therefore to always avoid having the idle extruders cross over the part. In cases where crossing over the part is mandatory, we hide defects in low visibility regions. In addition our path planner exploits the rampart, circulating within its double walls. This encourages wiping and reduces the quantity of oozing: even if the idle extruder crosses over the part, the defect will be minimal. Each time an extruder travels within the rampart we perform a refill of its hot plastic chamber, ensuring that it is always ready to print.

## 6.1. Overview

After slicing, we obtain a set of 2D slices, generated from the input 3D models. In our slicer each slice is a set of paths, tagged with an extruder id and a type. The path type can be either of perimeter, shell, or infill (Figure 2). Recall that most print paths are cyclic (Section 2.1) and therefore we can freely choose their start/end point when printing.

Conceptually our algorithm works in high resolution images representing the slice. We set the resolution to 0.05 mm per pixel, which for a nozzle of size 0.4 mm covers  $8 \times 8$  pixels. This offers enough resolution for detailed prints. Our slicer generate paths by contour tracing the slice images; the paths are therefore sampled at the pixel resolution.



In addition to the slices we also consider the visibility volume computed from the surface, using ambient occlusion (AO) – an example is shown in the inset. This sparse 3D grid defines in every surface point a visibility coefficient. This determines whether a point along a perimeter is highly visible. We do not describe here the computation of the visibility volume and refer the reader to the survey on ambient occlusion techniques by Méndez-Feliu et al. [MFS09]. In this volume, at any point in space we store a value between 0 and 1; 1 being the value for the most visible point.

The toolpath planning problem is a constrained form of TSP (Section 2.2). Solving directly with a TSP heuristic poses several difficulties. First, we need to consider a large number of nodes: paths have to be finely sampled to properly take into account visibility. Second, TSP is a global compromise between the order with which paths are printed and the navigation between them, while we want to favour quality over print time (within reasonable bounds): navigation is of primary importance. Therefore we divide the problem in two distinct steps:

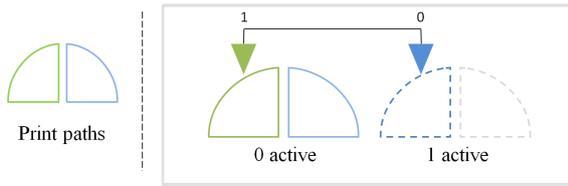
1. Ordering of paths, described Section 6.3;
2. Navigation between paths, described Section 6.4.

The ordering step decides the order in which print paths will be visited. The navigation is solved in a second step, maximizing the quality while taking into account the travel time. The benefit of resolving ordering first is to let us formulate the navigation optimization as a shortest path search in a graph, obtaining a high quality solution.

After these two steps we obtain a complete set of toolpaths, including travel paths that avoid or hide print defects.

## 6.2. Handling of multiple extruders

Our path planner always keeps track of the position of all extruders. However, all coordinates are rewritten using extruder 0 as the reference. The print paths for the other extruders are therefore translated into the frame of reference



**Figure 9:** Left: *Print paths for both extruders.* Right: *During printing, we consider the motion of extruder 0 only. The plain and dashed blue lines reveal its trajectory. However, when printing in green extruder 1 is activated, depositing plastic at the correct location (plain green line).*

of extruder 0, as illustrated in Figure 9. When printing with extruder 1, the carriage will move along a trajectory defined for extruder 0, but extruder 1 will be extruding plastic at the correct location.

### 6.3. Ordering of perimeters

Our ordering step makes sure that paths are visited in reverse order of inclusion, from the inner-most towards the exterior of the slice. This ordering avoids an extruder to have to re-enter a region whose perimeter was just printed.

Since some regions can enclose multiple child regions, we also have some degrees of freedom in the ordering. We exploit these to further minimize the risk of defects, in particular holes due to missing plastic.

**Inclusion tree.** Most of the paths, with the exception of infills, are cycles which follow the hull of the surface. We organize the slice into zones, where each zone is a region separated by a print path. This is illustrated Figure 10, left.

The notion of zone is well defined since, by construction, paths do not intersect: they capture the contours of the intersection of a plane with a well defined interior-exterior mesh. Infill paths are generated by the slicer only within the inner parts. In case of thin-features paths may come in contact with each other, but do not cross.

Similarly to prior work [CC06], we build a tree representing the zones with the most exterior zone at the root (Figure 10, right). The tree contains two types of nodes: zone nodes and path nodes. Each parent zone node is separated from a child zone node by a path node, indicating which cycle separates both zones in the slice. The infill paths are attached to the zone representing the inside region they are filling. The tree is quickly built with a flood-filling algorithm in the sparse image of the slice, tracking which paths are neighbouring each zone.

**Ordering children in a zone.** Zones of the inclusion tree may have multiple child paths (Figure 10, right). We can freely choose in which order these children are printed.

Their ordering is not without impact on quality. In particular, the risk of holes can be reduced by printing long paths

first: any lack of plastic is more likely to go unnoticed. Indeed, the inner-most paths are infills and shells (and therefore invisible). Note that lack of plastic is a rare event since our path planner encourages the use of the rampart, where refills occur.

We therefore order the children by decreasing path length. Once the children of all zones have been ordered, we traverse the tree in post-order, gathering all paths. For the example Figure 10, the final print order will be  $F0, F1, F2, D, B, C, A$ .

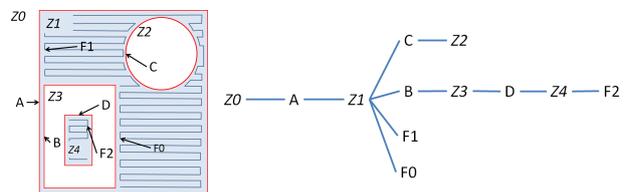
### 6.4. Navigation

After the order is determined we obtain a sequence of paths to be printed. We seek to generate travel paths in between. Each will navigate from one print path to the next, minimizing defects, using the rampart for circulation, and maintaining the print time low whenever possible.

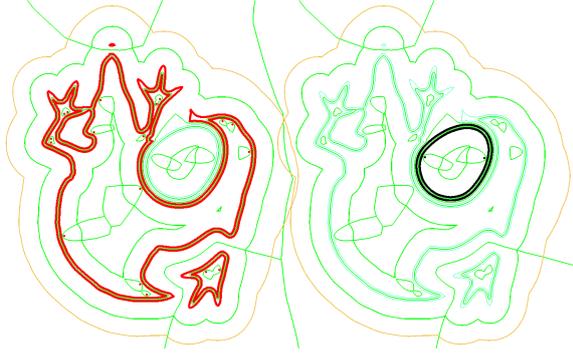
We optimize for navigation by a Dijkstra algorithm in a *navigation graph* comprising the paths to be printed, as well as a number of additional navigation pathways. The Dijkstra algorithm traverses the graph from one print path to the next, constrained by the order computed previously. Note that it computes a single navigation path and therefore solves for the global navigation problem – as opposed to solving for each in-between travel path independently.

The optimized navigation path is only concerned with travel: it is enough to reach the start of a print path and resume from its end, or to reach any single point along a cyclic print path – the extruder resumes from the same location after printing the cycle.

**Navigation graph.** The navigation graph has a sparse set of nodes which are located at integer coordinates in a grid. Multiple nodes can occupy a same coordinate. The resolution of the grid matches the resolution of the slice, with a larger extent to take into account the motion of all extruders. A node in the graph represents the reference extruder positioned at this location. Nodes are connected through edges representing segments in the plane. The cost of edges takes into account the defects that can occur when traveling this edge, considering the position of all extruders.



**Figure 10:** Left: *A slice showing print paths (A–D), infill paths (F0–F2) and zones (Z0–Z4). Each zone is a different region of the slice plane, with Z0 the outermost zone.* Right: *The inclusion tree captures inclusion relationships. Note how B, C, F0, F1 are children of the same zone Z1.*



**Figure 11:** The graph nodes of a slice of the dragon-heart model, for a dual left/right extruder printer. The right (red) extruder is the reference. Green paths are navigation pathways, the orange paths circulate within the rampart. The bold colored paths are the location of the reference extruder while printing; their color indicates which extruder is activated. Note the red/black dots corresponding to the start/end points of (open) infill paths.

The construction of the graph is described in the next paragraphs. An example graph is shown Figure 11.

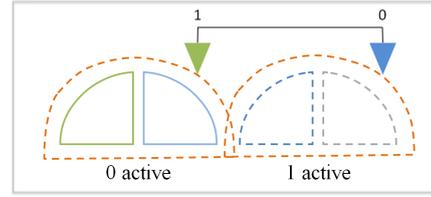
**Print paths.** We insert in the graph the paths along which plastic has to be deposited: cyclic paths are rasterized into the grid, adding nodes and edges between neighbors. We only add the start/end points of open paths.

**Navigation paths.** It would be wasteful to add one node per grid coordinate in the graph for navigation: once optimized, travel paths use only a small set of optimal pathways to navigate around different regions of the part. To speed up computations we explicitly select a small number of pathways and connect them to form the complete navigation graph.

We form the navigation pathways by adding to the graph a set of nodes  $\mathcal{N}$ . We start by inserting the nodes corresponding to multiple versions of the print paths, each translated in the reference frame of one extruder (dashed lines in Figure 9). We call these the *ghost paths*. Even though they are navigation paths they generally incur a large cost since they correspond to cases where an extruder passes over a mismatching color. They serve their main purpose at the next stage, when constructing additional navigation paths going around print and ghost paths as illustrated Figure 12.

We next compute a distance field in the grid [Dan80], from all path nodes added so far (print and ghost). The distance field indicates for each empty cell at coordinate  $e$  in the graph grid, the vector  $\mathbf{d}(e)$  to the closest cell occupied by a node  $n = G(\mathbf{d}(e))$ , with  $G$  the graph grid.

Using the distance field, we select a number of additional nodes: We form go-around pathways by selecting nodes at a fixed distance in the distance field (we use 4 times the nozzle width). We also form pathways between print paths, selecting the medial axis of the distance field – that is, select-



**Figure 12:** The orange dashed lines are navigation paths added around the print and ghost paths. The ghosts are necessary to generate the navigation paths allowing extruder 0 to travel while extruder 1 avoids crossing the print.

ing the nodes on either side of the medial axis which is located along integer cell edges. Finally, we add nodes for the path enclosed within the rampart, again in multiple versions translated in the frame of reference of each extruder. This provides a circulation around the part regardless of which extruder is active.

Figure 11 shows the nodes of a complete navigation graph.

**Edges.** We next add edges in the graph. Edges are always added in both directions ( $a \rightarrow b$  and  $b \rightarrow a$ ).

First, all nodes at neighboring integer coordinates are connected. Second, we connect the navigation nodes  $\mathcal{N}$  to the rest of the graph, adding edges  $(n, G(\mathbf{d}(n)))$ , with  $n \in \mathcal{N}$ . Finally, the navigation nodes are connected together: For each node  $n \in \mathcal{N}$  we gather the set of nodes  $\mathcal{C}(n) = \{m \in \mathcal{N} | n \neq m, G(\mathbf{d}(n)) = G(\mathbf{d}(m))\}$ . These are the navigation nodes having the same closest node as  $n$ . We then add all edges  $\{(n, m) | m \in \mathcal{C}(n), \mathbf{d}(n) \cdot \mathbf{d}(m) > 0.5\}$ , where " $\cdot$ " denotes the scalar product. This connects  $n$  to the other navigation nodes that are on the same side of the node  $G(\mathbf{d}(n))$ , within a tolerance.

**Edge costs.** The cost of following an edge between nodes  $p \rightarrow q$  in the graph is:

$$c(p \rightarrow q) = W_d * d(p, q) + T(p \rightarrow q)$$

with  $d(p, q)$  the Euclidean distance between the grid coordinates of  $p$  and  $q$  normalized by the largest extent of the slice, and  $W_d$  a weight allowing to trade between travel time and quality (we use  $W_d = 0.1$ ). The term  $T$  is computed as:

$$T(p \rightarrow q) = \sum_i^E (M_i(q + \delta_i) \cdot ao(q + \delta_i) + Z(p + \delta_i \rightarrow q + \delta_i))$$

where  $M_i(q + \delta_i)$  checks in the slice image whether at position  $q + \delta_i$  the extruder number  $i$  is above a mismatching color. If that is the case, then  $M_i(q + \delta_i) = 10$  and 0 otherwise.  $ao(x)$  returns the value at coordinate  $x$  in the AO grid. In case no value is available at  $x$  we search for the closest value in a small neighborhood (nozzle width), and return 0 if none is found. The  $ao(q + \delta_i)$  factor thus modulates the penalty in function of the visibility of the point.

$Z(u \rightarrow v)$  takes into account edges entering or exiting print paths. These are the locations where zippers may occur and

strings may deposit. Therefore, it is defined differently depending on the type of edge:

$$Z(u \rightarrow v) = \begin{cases} 0 & \text{if the edge } u \rightarrow v \text{ does not exist} \\ ao(u) & \text{if } u \text{ on print path, } v \text{ on navigation} \\ ao(v) & \text{if } v \text{ on print path, } u \text{ on navigation} \\ 0 & \text{otherwise} \end{cases}$$

Our edge cost favors travel paths avoiding strings to deposit: the term  $M_i$  strongly penalizes any crossing of an extruder above a print path of a different color. In all other cases the term  $Z(u \rightarrow v)$  ensures that defects (zippers, strings) will be located in low visibility areas, as given by ambient occlusion.

Edges belonging the rampart pathway have a smaller cost (10 times smaller) to encourage their use.

**Shortest path optimization.** We solve navigation by searching for the shortest path from the start point of the slice, through all print paths, to a point of the last print path. We select the start point from the end point of the previous slice, or the origin for the first slice. The path is constrained to traverse the print paths in the order defined by the ordering step. Open paths are visited as soon as their first point is reached, and the shortest path search resumes directly from their last point. Cyclic paths can only be considered visited when all their nodes have received their shortest path cost.

We backtrack the overall shortest path from the node of the last perimeter having smallest cost. The shortest path goes through one node of each cyclic print paths and the first/last node of non-cyclic paths. The sequences of nodes between each print path node form travelling paths. We insert them into the initial sequence of paths to obtain the final sequence of paths. These are then used to produce the instructions for the printer (G-code).

### 6.5. Triggering refill

The path planner generates travel paths that tend to circulate within the walls of the rampart. After path planning, we follow each travel path and detect when it enters/exits the rampart. We then activate the extruder which circulates the rampart, allowing plastic to flow. This is illustrated in Figure 7.

Our goal is to run a sufficient volume of plastic through the nozzle to properly refill the hot chamber. Since the travel path within the rampart is already determined, we can only vary the travel speed or the extrusion speed. We use a fixed extrusion speed ensuring reliability. We therefore adapt the travel speed. Given a path length  $L$  (mm), a volume to push  $v$  ( $mm^3$ ), a filament extrusion speed  $e$  ( $mm/s$ ), and a filament diameter  $f$  (mm), we compute the travel speed as:

$$s = \frac{L \cdot e \cdot \pi \cdot f^2}{4 \cdot V}$$

For mechanical safety we ensure the speed remains below a

maximum ( $120mm/s$ ). This guarantees that enough plastic is pushed, regardless of the length of the segment within the rampart. We use  $v = 2mm^3$  and  $e = 2mm/s$ .

While it is possible for several refill paths to overlap, we did not find that to be a source of concern during printing – however we disable refill on the first few layers where the nozzle is close to the print bed.

## 7. Results

We implement our method in our slicer [Lef13]. We print on a Replicator 1 dual with Sailfish 7.4, using ABS plastic.

Our prints are all obtained with same values for all parameters. Our parameters are 30 mm/s for perimeters, 60 mm/s for other print paths, 120 mm/s travel, 20% infill and 2 shells, 0.5mm prime/deprime. All the results shown in this paper are exactly as they came out of the printer: we do not clean them in any way.

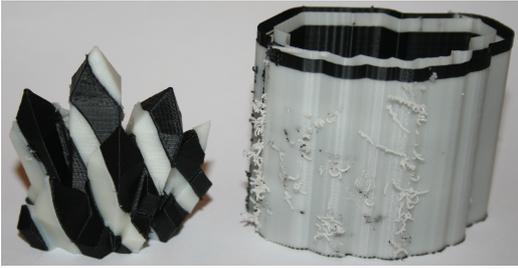
Figure 20 shows how zippers are hidden in low visibility regions by the edge cost defined Section 6.4. This benefits both single and dual color prints.

### 7.1. Multiple color prints

We first test two challenging models. *Dragon-heart* has a significant imbalance between both extruders, implying that one is idle for long periods of time. The body is larger than the extruder spacing in all directions. The dragon is printed at 0.2 mm layer height. *Two-color-world* combines fine, detailed features with a circumference which is larger than the extruder spacing. We print it in black and natural plastic at 0.3 mm layer height. The later being translucent, any color smears end up being visible. Nevertheless, we reach high print quality with crisp color separation as shown Figure 19. This is to be compared with ReplicatorG/Skeinforge



**Figure 13:** Our result (right) has almost no defects compared to the same model printed with Skeinforge/ReplicatorG (left). Pictures focus on the worst regions of the surface. Both models use the optimized azimuth angle.



**Figure 14:** A black and white crystal and its rampart. Note the significant amount of stringing captured on the rampart. Small defects are due to thin color slabs not adhering.

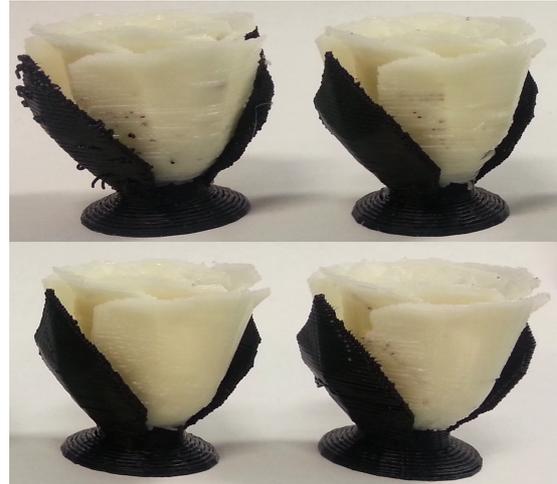
as shown in Figure 18. Figure 13 compares dragon-heart printed with ReplicatorG/Skeinforge and our technique, both using our azimuth angle optimization. Figure 17 shows the Robot-ice model printed in opaque red within a crust of translucent natural white ABS, at 0.3 mm layer height. As shown by the back-lit model our approach produces a clean output within the volume itself. Figure 15 compares the result of progressively enabling each component of our technique (none, rampart only, full) versus the result of Makerware 2.4. The model fits within the extruder spacing, so all defects are due to Case 1. For this comparison we match the settings of Makerware and use 150 mm/s for the travel speed, 40 mm/s for the perimeters and 90 mm/s for all other paths. Layer height is 0.3 mm. The print time for our full technique is 46 min versus 24 min without, using respectively 4.6 and 2.3 meters of filament. The overhead goes down on larger prints (see Figure 16). Makerware 2.4 prints in 39 min using 3.91 meters of filament. Some differences with Makerware are explained by the different approaches used for slicing, in particular regarding the thin petals; however color smears are clearly visible in all but our result.

Figure 16 summarizes the performance of our approach. Timings for processing include all steps (azimuth, path planning, AO). Timings are measured on an Intel I7 4770, 3.4 GHz equipped with a GeForce GTX 770. We process multiple slices in parallel (8 threads). Note that the relative overhead in filament length becomes smaller on larger objects or with denser infills (here we use only 20% infill).

### 7.2. Limitations and future work

The rampart wastes a small amount of plastic – to be compared with the cost of failed or low-quality prints. Plastic is also easily recycled in filament (see for instance the *Filabot*). Printing the rampart may involve crossing over the print, risking color smears. This is visible at close inspection on the right side of the Robot-ice, Figure 17.

While we strongly minimize the amount of defects, some color smears may still be visible when one extruder prints for a long time and, due to the object geometry, misses the wipe. Vertical complexity also increases the rampart-surface distance which reduces its efficiency. A direction of future



**Figure 15:** Quality comparison. Top left: Print without our approach. Top right: Print with only the rampart (no navigation, no refill). Bottom left: Our final result. Bottom right: Print from Makerware 2.4. Rose by user Jillian (*Thingiverse*)

Model	Processing	Print time	Filament
World	4m22s	2h20m [1h32m]	12.3 [8.3]
Dragon	4m25s	1h51m [1h16m]	7.50 [4.3]
Robot	8m40s	1h53m [1h06m]	9.91 [4.9]

**Figure 16:** Timings for processing, printing and length of plastic filament used (meters). Numbers in brackets are timings/lengths without using our technique.

work is to further optimize the shape of the rampart to perform more wipes. Our ordering step could be improved to take print time into account. Our edge cost could also easily incorporate additional terms, for instance hiding zippers in high curvatures of the surface.

### 8. Conclusion

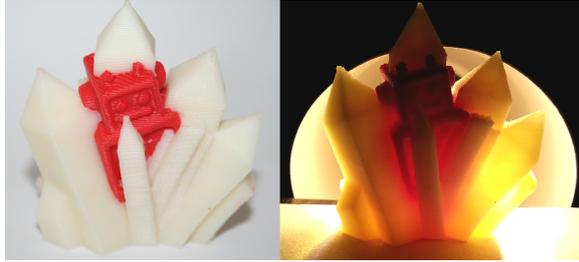
Our work makes it easier to get satisfactory results with multiple color prints. By carefully analysing the defects and the behavior of the extruders we proposed a technique removing or hiding most issues. Our algorithm finds a better azimuth angle for printing and uses a specialized path planner exploiting a disposable rampart in close proximity of the shape to protect the printed part and refill the extruders with plastic. Our approach also improves the quality of single color prints by hiding zippers in regions of low visibility. This is a side effect of our edge cost definition, which seeks to enter perimeters in hidden locations.

We hope our work will help CG enthusiasts turn their designs into colorful, fascinating physical objects.

### 9. Acknowledgments

We thank the anonymous reviewers for their help with improving our paper, Jérémie Dumas for figures and pictures,

our families for bearing with the 3D printers during week ends, and members of the 3D printing community who designed the models shown here. This work was funded by ERC ShapeForge (StG-2012-307877).



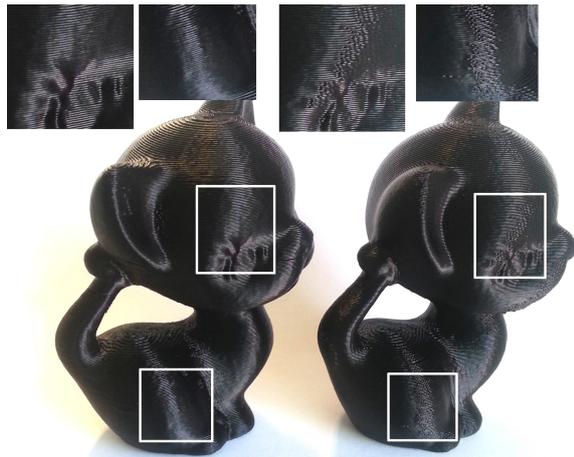
**Figure 17:** Robot-ice printed with our technique. The backlit model reveals the absence of smears within the volume.



**Figure 18:** From left to right: Two-color-world model, printed with ReplicatorG/Skeinforge, our technique without azimuth optimization and our complete technique.



**Figure 19:** Our result seen from different angles.



**Figure 20:** Left: Our result, zippers are hidden in low visibility regions. Right: Skeinforge with jitter plugin.

## References

- [AAD98] ALEXANDER P., ALLEN S., DUTTA D.: Part orientation and build cost determination in layered manufacturing. *Computer-Aided Design* 30, 5 (1998).
- [CC06] CHOI S., CHEUNG H.: A topological hierarchy-based approach to toolpath planning for multi-material layered manufacturing. *Computer-Aided Design* 38, 2 (2006).
- [CFN\*95] CHENG W., FUH J., NEE A., WONG Y., LOH H., MIYAZAWA T.: Multi-objective optimization of part-building orientation in stereolithography. *Rapid Prototyping Journal* 1 (1995).
- [CW04] CASTELINO K., WRIGHT P. K.: Tool-path optimization for minimizing airtime during machining. *Journal of Computing and Information Science in Engineering* (2004).
- [CZ10] CHOI S., ZHU W.: A dynamic priority-based approach to concurrent toolpath planning for multi-material layered manufacturing. *Computer-Aided Design* 42, 12 (2010).
- [Dan80] DANIELSSON P.-E.: Euclidean Distance Mapping. *Computer Graphics And Image Processing* 14 (1980).
- [FF95] FRANK D., FADEL G.: Expert system-based selection of the preferred direction of build for rapid prototyping processes. *Journal of Intelligent Manufacturing* 6, 5 (1995).
- [Lef13] LEFEBVRE S.: Icesl: A gpu accelerated modeler and slicer. In *European Forum on Additive Manufacturing* (2013).
- [MFS09] MÉNDEZ-FELIU Í., SBERT M.: From obscurances to ambient occlusion: A survey. *The Visual Computer* 25, 2 (2009).
- [MRI00] MASOOD S. H., RATTANAWONG W., IOVENITTI P.: Part build orientations based on volumetric error in fused deposition modelling. *The International Journal of Advanced Manufacturing Technology* 16, 3 (2000).
- [PRD95] PANDEY P. M., REDDY N. V., DHANDE S. G.: Slicing procedures in layered manufacturing: a review. *Rapid Prototyping Journal* 9, 5 (1995), 274–288.
- [PWT07] PATRICK W. TURLEY T. L. T.: Extrusion tip cleaning assembly, 2007. US Patent 7744364 B2.
- [SVB\*12] STAVA O., VANEK J., BENES B., CARR N. A., MECH R.: Stress relief: improving structural strength of 3d printable objects.
- [VH86] VAN HOOK T.: Real-time shaded nc milling display. In *Proceedings of SIGGRAPH* (1986).
- [VWRKM13] VIDIMČE K., WANG S.-P., RAGAN-KELLEY J., MATUSIK W.: Openfab: A programmable pipeline for multi-material fabrication. *ACM Transactions on Graphics* 32 (2013).
- [WLC10] WANG C. C. L., LEUNG Y.-S., CHEN Y.: Solid modeling of polyhedral objects by layered depth-normal images on the gpu. *Computer-Aided Design* 42, 6 (2010).
- [WMJC02] WAH P. K., MURTY K. G., JONEJA A., CHIU L. C.: Tool path optimization in layered manufacturing. *Iie Transactions* 34, 4 (2002).
- [WWY\*13] WANG W., WANG T. Y., YANG Z., LIU L., TONG X., TONG W., DENG J., CHEN F., LIU X.: Cost-effective printing of 3d objects with skin-frame structures. *ACM Transactions on Graphics* 32, 5 (2013).
- [XCY11] XIAOMAO H., CHUNSHENG Y., YONGJUN H.: Tool path planning based on endpoint build-in optimization in rapid prototyping. *Proceedings of the Institution of Mechanical Engineers (Part C)* (2011).
- [ZPZ13] ZHOU Q., PANETTA J., ZORIN D.: Worst-case structural analysis. *ACM Transactions on Graphics* 32, 4 (2013).