Eurographics Workshop on Graphics for Digital Fabrication (2016)
A. Medeiros e Sá, N. Pietroni, and K. Rodriguez Echavarria (Editors)

Tight printable enclosures and support structures for additive manufacturing

Samuel Hornus,[1] Sylvain Lefebvre,[1] Jérémie Dumas [1] and Frédéric Claux [2]

[1] Inria Nancy - Grand Est
[2] XLIM, Université de Limoges

Abstract

Additive manufacturing is a process by which a three dimensional object is created layer after layer, through selective deposition of material. It often requires the automated generation of auxiliary shapes, to temporarily support the object, to protect its surface, or to carve inner cavities and reduce material usage.

In this context, we define a printable enclosure *as a minimal volume enclosing a given shape and whose boundary can be printed at the smallest possible thickness while ensuring proper bonding between layers. Such an enclosure is well suited to serve as auxiliary structure for additive manufacturing: it is easy to print and require little material. In this paper, we demonstrate its use on three different applications: enclosing a print within protective walls that are close to the surface; generating large inner cavities whose walls are printable, and finally modeling support structures that provide a dense support to the downward facing surfaces while vanishing as quickly as possible below the supported object.*

We obtain the shape of an enclosure by considering constraints on its set of slices along horizontal planes. In practice, the set of slices is discrete and the constraints afford for an efficient sweep-like construction algorithm using morphological operations on the slices. We discuss the printability and optimality of the enclosures and their boundary walls.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation

1. Introduction

Additive manufacturing often requires the generation of auxiliary shapes to help the manufacturing process. Typical examples are internal cavities to reduce material use, walls protecting the printed model from defects (oozing or heat dissipation), or support structures to print overhanging regions.

These auxiliary shapes are often invisible on the final printed model, either because they are hidden from view—eg internal cavities—or destroyed after the process completes—eg supports, protective walls. A common goal in the generation of these shapes is that they should be small and *printable*. That is, they should comply with the constraints of the manufacturing process. In this work we focus on modeling such auxiliary shapes for layer-by-layer additive manufacturing by deposition of material along paths. Typical examples are printers using thermoplastic filament (Fused Filament Fabrication), extruding paste (eg Fab@Home) or contour crafting for large scale constructions.

When material is deposited, it properly sticks to the layer below only if there is enough bonding surface. Typically proper bonding is achieved if half the width of the newly deposited material track is supported by the previous layer, but this threshold depends on many parameters (material, temperature, layer thickness) and therefore changes between processes. The layer-to-layer bonding constraint translates geometrically into a lower bound on the slope of the boundary surface of the printed volume. In order to be printable, the walls of a shape should see their slope above a minimal value.

We propose to consider shapes that *minimally* enclose a given volume while having a printable boundary: the *walls* of the shape are sufficiently close to vertical to guaranty proper bonding of the layers at the minimal possible thickness. A direct application is to generate protective walls of minimal thickness, that closely follow an input shape. We also demonstrate the benefits of our technique for generating inner cavities, support structures, and for closing small overhangs in input models.

Contributions. Our main contribution is the formal definition of a minimal enclosure with printable walls in terms of constraints on its set of slices. This definition leads to a simple algorithm exploiting morphological operations on the slices to compute the final enclosure volume. We propose several applications and print a variety of objects with our technique on filament printers.

DOI: 10.2312/gdf.20161074

2. Previous work

To the best of our knowledge no existing technique addresses the problem with the same generality as our work. However, there is related work in each of the applications we target.

Support volumes. The most typical approach for support consists in extruding downwards the faces which are at too steep an angle to print (*KISSlicer*, *Makerware*, [SHEE13]). This defines a *support volume* which is then printed filled with a weak pattern, so that it can be removed. Another option is to print the support with dissoluble material. The main disadvantage of this approach is the large amount of material being used, and the required print time.

Huang *et al.* [HYW*09] reduce material usage by shrinking the extruded support volume in its middle section. The shrinkage is chosen according to the overhang constraints. Closer to our work, the approach of Heide [Hei11], generates a support shape from top to bottom, slice by slice. At each step, the convex envelope of the slice polygon is offset inwards (shrunk) and intersected with the slice polygon which has been offset outwards (dilated). The effect is that internal holes are reduced in size while the outer convex envelope shrinks. This reduces the support volume size and complexity in the lower regions. The common point with our technique is the slice-by-slice generation of the support shape in a downward sweep. However, our technique produces volumes that vanish much more quickly and are minimal under the given constraints.

Autodesk Meshmixer and Vanek *et al.* [VGB14a] produce thin support structures resembling trees. The structures are very sparse and therefore use little material while being easy to remove. Dumas *et al.* [DHL14] follow a similar direction but rely on scaffolding structures with horizontal bridges, which are less sensitive to torque. These methods however support the model in a sparse set of points, which reduces the bottom quality of the surfaces. The generation cost and structure complexity grows with the number of supported points, which can become problematic on models with large, intricate overhang configurations. Our approach integrates well with these techniques as it allows to densely support the bottom of the print and produces a quickly vanishing support volume.

Protective walls. Additive manufacturing by deposition is one of the few technologies to support multiple materials. For instance, many available FFF printer feature multiple extruders mounted on a same carriage, e.g. the *Ultimaker 1* can be equipped with two extruders, while the *Rova3D (ORD Solutions)* supports up to five extruders.

The main difficulty arising from the use of multiple extruders is *oozing*. While one extruder prints, material oozes from the others. The oozing material deposits onto the printed object, producing defects. A typical approach shared by several slicing software is to print a structure close to the object, cleaning the extruders before reaching the print. *Makerware* prints walls to the left and right of the object, Reiner *et al.* [RCM*14] print a tower close to the object. An important consideration is that these structures should be close to the object surface to reduce travel time and the chance that oozing resumes. Hergel and Lefebvre [HL14] create a wall by extruding the silhouette of the object along the up direction, however the distance to the object typically increases in the upper regions. More recently the slicer *Cura (Ultimaker)* introduced an “oozing shield” — a tight envelope around the object that prints without support. A reading of the source code reveals that the algorithm used by Cura corresponds to the algorithm that results from our analysis of minimal enclosures (see §4). To the best of our knowledge there is no publication describing this technique or analysing its good behavior. Besides providing a more generic approach which is applicable to other aspects of 3D printing (cavities and support), in this paper we explain the desired behavior of the algorithm and discuss how to control the appearance of local minima to minimize the enclosed volume.

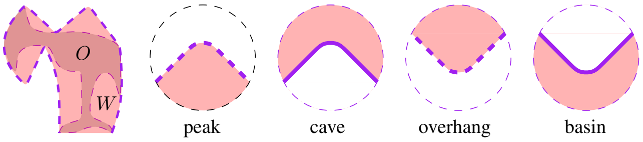

Figure 1: From left to right. *An object O inside an enclosure W (enclosed volume in orange, walls in purple). Close-ups on different configurations of local extrema for an enclosure (side view).*

Printability constraints. A number of approaches consider printability constraints, either in terms of volume, shape or mechanical properties. Luo *et al.* [LBRM12] consider the problem of partitioning an object that does not fit a print volume into several smaller parts that are later re-assembled. Hu *et al.* [HLZCO14] decompose a shape into pyramidal parts that can be independently printed. Our goal is different, we seek to *generate* shapes that minimally enclose a volume while having printable walls. Stava *et al.* [SVB*12] consider reinforcing an object for 3D printing by automatically adding struts. Prevost *et al.* [PWLSH13] carve and deform a shape so that it remains properly balanced after manufacturing. Telea and Jalba rely on voxels and morphology operators to detect non printable areas of a model [TJ11]. We rely on a similar formalism to generate the minimal enclosures and tight supports.

Hollowing. Wang *et al.* [WWY*13] carve the inner volume of an object and then optimize a truss structure to reinforce it. Vanek *et al.* [VGB*14b] similarly hollow a large object and decompose its surface into smaller pieces. The pieces are packed and oriented to save print time and material usage. These techniques are primarily targeted at powder based 3D printers. When used with deposition printers, our approach can help further reduce material use by maximizing the size of the cavity and reducing the required support, since the walls of our synthesized cavities remain printable.

3. Overview

We start by giving intuitive explanations regarding the enclosures we seek to build and their desirable properties. The formal definition and algorithms are given in §4.

Besides the slope constraint mentioned in the introduction, we face an additional challenge to ensure printability: local extrema. Figure 1 illustrates the typical configurations of local-extrema that occur on the boundary of an enclosure. Assuming the inside of the

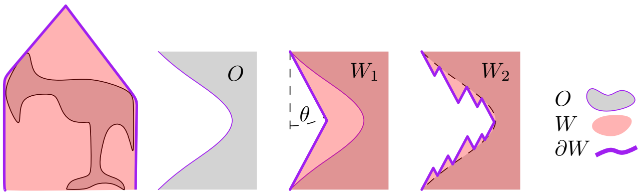

Figure 2: From left to right. *An enclosure without overhangs, disallowing downward facing walls. Part of an object O. A desirable enclosure W_1 for this part of O, following our definition. Another enclosure W_2 with smaller volume, but too many basins and overhangs making it difficult to print.*

enclosure is empty, *peaks* and *caves* can be printed directly while the tips of *overhangs* and *basins* require a pillar supporting them from below.

Our method produces the enclosure of *smallest* volume having neither basins nor caves (see §4.4), while peaks and overhangs may appear on the wall of the enclosure. Excluding basins exactly achieves our goal as it guarantees that the upper part of an enclosure is printable without support. However, on the lower part, each overhang requires a support pillar, and each additional pillar increases print time and material use.

Overhangs could be eliminated altogether by constraining the walls to never face downward. The result would be enclosures with vertical walls and a printable slanted roof as illustrated in Figure 2 left. This however results in a larger enclosed volume (see §7.3). Interestingly, the problem of minimizing the volume under the slope constraint has a trivial but undesirable solution using a very large number of spurious local-extrema as illustrated by enclosure W_2 in Figure 2.

These two types of enclosure, with zero or very many local extrema, are the two extremes in between which we set ourselves to reach a compromise, balancing the minimization of the volume and the creation of a small number of overhangs. We argue that the approach described in the next section offers a good compromise. We also discuss how to gain some control over this trade-off in §5.2.

4. The minimal enclosure

We now describe the problem more formally, give a precise definition of the enclosures we seek to generate as well as a complete algorithm.

4.1. Reasoning on slices

In order to match the way the objects are printed, layer by layer, we find it easier to reason about horizontal slices of the volumes that we manipulate. For any volume $V \subset \mathbb{R}^3$, we define the slice $V_{|z}$ as the intersection of V with the horizontal plane at height z. We consider slices as objects living in the two-dimensional plane $\mathbb{R}^2$. We write $S \times \{z\} = \{(x,y,z) \mid (x,y) \in S\} \subset \mathbb{R}^3$ for the slice S positioned at height z in the printing space.

For the discussion, we use the morphological operators on planar shapes. If S is a planar shape and r a non-negative number, the *dilation* $S^{\uparrow r}$ is the set of points at distance r or less from S: $S^{\uparrow r} = \{p \mid \exists q \in S, |p-q| \leq r\}$. Similarly, the *erosion* of S of radius r is $S^{\downarrow r} = \overline{\overline{S}^{\uparrow r}}$ where $\overline{X} = \mathbb{R}^2 \setminus X$. The *opening* of S of radius r is $\textsc{Open}(S,r) = S^{\downarrow r \uparrow r}$. The *closing* of S of radius r is $\textsc{Close}(S,r) = S^{\uparrow r \downarrow r}$. These operators satisfy

$$\textsc{Open}(X,r) \subset X \subset \textsc{Close}(X,r). \tag{1}$$

4.2. Problem definition

We seek to compute a volume $W \subset \mathbb{R}^3$, the *enclosure*, with boundary surface ∂W (the *wall)* having the following properties:

- The wall ∂W is printable: In the discrete setting (the machine prints a discrete set of thick slices), the boundary of each slice must be supported by the boundary of the slice below it. Sections 4.3 and 4.4 show how this translates into morphological constraints on each pair of slices.
- The enclosure W contains O and is minimal with respect to inclusion, ie there is no volume $W' \subset W$ that satisfies the same properties. In the discrete setting, we want to minimize the area of the slices of W while ensuring that for all z, $O_{|z} \subset W_{|z}$.
- The enclosure W does not produce unnecessary local-extrema, eg it avoids producing cases such as W_2 in Figure 2. We discuss this constraint in §4.4.

We use an angle parameter θ to constrain the slope of the boundary ∂W of the enclosure: On an *idealized* smooth surface representing the enclosure W, the z component of any unit normal vector of ∂W would satisfy $|z(n)| \leq \sin\theta$. Note that when $\theta = 0$ the wall ∂W is a vertical extrusion; the larger θ is the more freedom W has. Since our goal is to print a wall with the minimal thickness, there is an upper bound on admissible values of θ allowing for proper bonding between layers.

We view $W_{|z}$ as a deformable planar shape as z varies. Let us assume that we sweep the slices of W from the bottom up. Then, the constraint of "printability" of its boundary ∂W says that a slice should not deform too quickly. Given two heights z and z' satisfying $z < z'$, parts of $W_{|z'}$ may have grown too fast from $W_{|z}$, others may have shrunk too fast and new connected components may have appeared during the sweep from z to z'. We observe that fast-shrinking regions become fast-growing regions if we reverse the sweep direction, so we focus on taming fast-growing regions. Fast-shrinking ones will be tamed by symmetrizing the obtained constraints on the slices of W.

4.3. Enforcing the slope and volume constraints

An obvious approach to limit fast-growing regions is to bound the growth of a slice during the sweep (Figure 3-*Left*). Setting $r = (z'-z)\tan\theta$, we require that

$$W_{|z'} \subset W_{|z}^{\uparrow r}. \tag{2}$$

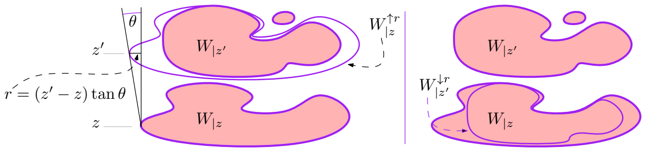

Figure 3: *View of slices of W at height z (bottom) and z' (top).* Left. *The large component of $W_{|z'}$ is included in $W_{|z}^{\uparrow r}$ but its small component is not.* Right. *The small component has disappeared from $W_{|z'}^{\downarrow r}$ so that $W_{|z'}^{\downarrow r} \subset W_{|z}$.*

This "bounded growth" constraint (2) on W is too strong, however, since it does not allow small connected components in $W_{|z'}$—defined as a component $C \subset W_{|z'}$ such that $C^{\downarrow r} = \emptyset$—to start appearing between heights z and z' (Figure 3-*Left*). Constraint (2) thus severely limits the possibility to obtain an enclosure W tightly fit around O.

In order to weaken condition (2) we replace $W_{|z'}$ with its morphological opening of radius r:

$$W_{|z'}^{\downarrow r \uparrow r} \subset W_{|z}^{\uparrow r}. \tag{3}$$

The morphological opening takes the small connected components out of the constraint so that they are allowed to survive by (3). Note that (3) is indeed weaker that (2) since (2) ⇒ (3) (using (1)).

4.4. Avoiding spurious extrema

As we have previously discussed in §3, using only the slope and volume constraints the enclosure may exhibit many undesirable spurious extrema, which are basins at the top and overhangs at the bottom (see enclosure W_2, Figure 2). This indicates that constraint (3) is too weak in the sense that we can define an enclosure W that contains O and whose slices $W_{|z}$ do deform slowly and satisfy constraint (3) but is arbitrarily close to the volume O in the Hausdorff sense and is far from being printable, just like the volume W_2 in Figure 2.

To overcome this problem, we have to constrain the enclosure W more. The dilation operator $(\cdot^{\uparrow r})$ in the constraint over W gives too much freedom to W. Perhaps more importantly, the dilation operator is less amenable to a constructive algorithm since W should be *minimal* with respect to inclusion.

We obtain a satisfying stronger constraint by getting rid of the dilation operator on either sides of (3):

$$W_{|z'}^{\downarrow r} \subset W_{|z}. \tag{4}$$

Since dilation (as well as erosion) is an increasing operator, it holds that (4) ⇒ (3); so (4) is stronger than (3).[†] Because we shrink the slice $W_{|z'}$ by r, constraint (4) still allows small connected component unrelated to $W_{|z}$ to exist (Figure 3-*Right*). We thus settle, for now, on constraint (4) to model the enclosure W. In order to handle fast-shrinking regions, we symmetrize it by removing the assumption that z is smaller that z':

$$\boxed{\forall z \forall z' \quad W_{|z'}^{\downarrow |z'-z| \tan\theta} \subset W_{|z}.} \tag{5}$$

Hats. Constraint (5) has another important property for which description we need to define the *hat* of a slice. The hat of slice $W_{|z}$ is the 3D volume symmetrical with respect to the horizontal plane at height z and bounded by the symmetrized and scaled graph of the euclidean distance to $\partial W_{|z}$ defined on $W_{|z}$:

$$\begin{aligned} \widehat{W_{|z}} &= \bigcup_{z' \in \mathbb{R}} W_{|z}^{\downarrow |z'-z| \tan\theta} \times \{z'\} \\ &= \Big\{ (x,y,z') \in \mathbb{R}^3 \mid (x,y) \in W_{|z} \\ &\qquad \text{and } |z'-z| \le \tfrac{\operatorname{dist}\left((x,y), \partial W_{|z}\right)}{\tan\theta} \Big\}. \end{aligned}$$

Figure 4 shows some hats. Clearly, if W satisfies constraint (5), then all the hats of W are contained in W, which immediately eliminates all the undesired enclosures exhibiting any number of local extrema and a high degree of un-printability. Furthermore, since W itself is contained in the union of its hats, W must be of the form:

$$W = \bigcup_{z \in \mathbb{R}} \widehat{W_{|z}}. \tag{6}$$

In Figures 2 for example, the enclosure W_1 is equal to the union of its hats, but the enclosure W_2 is not. The same holds for enclosure W in Figure 4.

Finally, the enclosure W must also contain O. This is easily expressed with the condition

$$\forall z \quad O_{|z} \subset W_{|z}. \tag{7}$$

In the next section, we show how to compute an enclosure satisfying constraints (5) and (7) in the case of a discretely sampled set of slices.

4.5. Solving the discrete case, SIMPLEENCLOSURE

In additive manufacturing printed objects are modeled as a discrete collection of slices. So that we only need to model the enclosure, or any volume, as a discrete set of slices as well; this makes the problem amenable to a simple algorithmic solution. The heights of the slices are given by $z_i = \Delta \times i$ for $i \in \mathbb{Z}$. To simplify the notation, we write $V_{|i}$ instead of $V_{|z_i}$. The constraints (5) and (7) above become, in the discrete case:

$$\forall (i,j) \in \mathbb{Z}^2, W_{|i}^{\downarrow |i-j| \Delta \tan\theta} \subset W_{|j} \quad \text{and} \quad O_{|j} \subset W_{|j}. \tag{8}$$

For convenience, we write $P(i,j)$ for the proposition $\left(W_{|i}^{\downarrow |i-j| \Delta \tan\theta} \subset W_{|j}\right)$. Constraint (8) becomes:

$$\forall (i,j) \in \mathbb{Z}^2, \quad P(i,j) \wedge \left(O_{|j} \subset W_{|j}\right). \tag{9}$$

[†] but (4) is neither stronger nor weaker than (2).

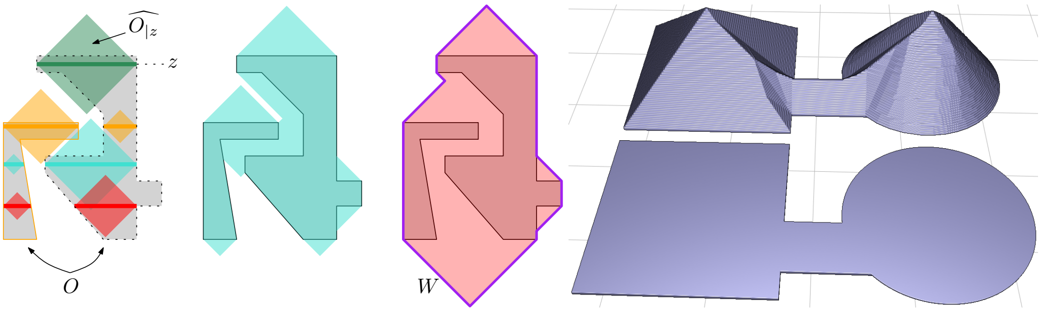

Figure 4: *In this illustration,* $\theta = \pi/4$. Left: *An object O (gray) with two components. The hat of four slices are shown in various colors.* Middle (cyan): *The union of the hats [of the slices] of O.* Middle (red): *The minimal volume satisfying constraints* (5) *and* (7)*. We call it the* simple enclosure *of O. It is equal to the union of its own hats; see eq.* (6)*.* Right: *A slice (front) and the upper half of its hat (back).*

In order to solve this set of equations (while also satisfying the minimality of the solution with respect to inclusion), we observe that P is transitive:

$$\begin{aligned}\forall i < j < k \quad & P(i,j) \wedge P(j,k) \Rightarrow P(i,k),\\ & P(k,j) \wedge P(j,i) \Rightarrow P(k,i).\end{aligned}$$

It is therefore sufficient to ensure that equation (9) holds for consecutive slices only, in both directions. The simplest way to obtain the smallest enclosure W, is to combine the subset relations in (9) for consecutive slices into an equality:

$$\forall i \in \mathbb{Z} \quad W_{|i} = W_{|i+1}^{\downarrow \Delta \tan\theta} \cup W_{|i-1}^{\downarrow \Delta \tan\theta} \cup O_{|i}.$$

In order to break the circular dependency and compute W, we propagate the slices of W first from the bottom up and then from top to bottom:

```
function SimpleEnclosure($O$)
    temp ←PropagateUp($O$)
    $W$ ←PropagateDown(temp)
    return the enclosure $W$
----------------------------------------------------------------
function PropagateUp($O$)           ▷ The input $O$ is made of $n+1$ slices
    $V_{|0} \leftarrow O_{|0}$                 ▷ Initialization of the bottom slice
    for $i = 1$ to $n$ do                ▷ Propagate from bottom to top
        $V_{|i} \leftarrow O_{|i} \cup V_{|i-1}^{\downarrow \Delta \tan\theta}$
    return the volume $V$
----------------------------------------------------------------
function PropagateDown($O$)
    $V_{|n} \leftarrow O_{|n}$                   ▷ Initialization of the top slice
    for $i = n-1$ down to $0$ do        ▷ Propagate from top to bottom
        $V_{|i} \leftarrow O_{|i} \cup V_{|i+1}^{\downarrow \Delta \tan\theta}$
    return the volume $V$
```

It is simple to check that the volume W returned by the function SimpleEnclosure satisfies all the required constraints and is stable in the sense that no additional slice propagation can further grow W.

While the abstractly defined enclosure W extends above and below its defining object O (Figure 4), the parts of W above and below O are of no interest for the printing process, and we do not need to compute or print them, as is reflected in the pseudo-code above. So, the actual printed height of the enclosure is always equal to that of object O.

5. The enclosure in practice

In this section, we build upon the simple enclosure modeling procedure described in §4.5 and provide the missing elements required to make the wall ∂W of the enclosure W actually printable. Indeed, while the printed wall sustains itself, there are still some printability issues around the local extrema of the wall. The wall may exhibit four kinds of local extrema (Figure 1). The local maxima include *peaks* (where the enclosure lies locally below the peak) and *caves* (where the enclosure lies locally above the cave). The local minima include *basins* (the enclosure lies locally below the basin) and *overhangs* (the enclosure lies locally above the overhang).

Local minima pose a problem: in order to print a local minimum, one has to print additional support material below it (lest we obtain spaghetti). This necessary structure makes it impossible to print basins, since the space below a basin is *a priori* already occupied by the object to be printed. Overhangs on the other hand are acceptable since they do lie above empty space so that a support structure might be added, as we detail in §5.1. Local maxima are printable since they are supported by parts of the wall from below. We examine basins (minima) and caves (maxima) in §5.2. Peaks pose no problem and are not examined further. Finally, §7.3 describes a less effective but simpler alternative enclosure.

5.1. Supporting overhangs

As we have seen above, thanks to the hat-containment property, the generated enclosure may exhibit some (but not too many) overhangs. We use the technique of Dumas *et al.* in order to detect the overhanging points that require support and generate a suitable scaffolding structure [DHL14]. Since the number of points requiring support is minimal, the generated scaffold is much simpler than the scaffold that would be generated for supporting the object of interest O. We exploit this observation to generated simpler supports for O in §7.2.

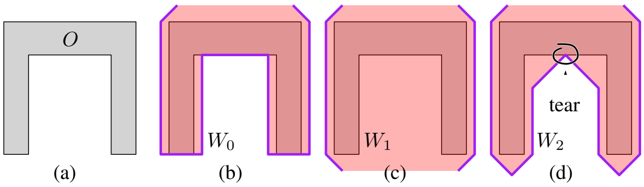

Figure 5: (a) *O is an upside-down cubical mug without handle. The volumes W_i are illustrated in transparent red color.* (b) *We have $W_0 = \textsc{PropagateUp}(O')$ where the slices of O' are dilations of those of O.* (c) $W_1 = \textsc{PropagateDown}(W_0)$. (d) $W_2 = \textsc{TightDownwardPropagation}(W_0)$.

5.2. Basins and caves

In the bottom-up propagation of the volume, the erosion of a slice is fused with the slice immediately above. This implies that all the basins of ∂O are filled by the enclosure, so that the resulting wall does not contain any basin. The absence of basin in W also means that all internal cavities of O have been filled in W. This is a property of the enclosure enforced by constraint (5), that is required for the elimination spurious local extrema (see §4.4). In several scenarios, the systematic filling of caves damages the tight fit of the enclosure wall to the object of interest. We discuss this issue below and show how a controlled departure from the strict observance of constraint (5) allows for tighter enclosures.

In the top-down propagation of the volume, the erosion of a slice is fused with the slice immediately below it. This implies that all the caves of ∂O are filled by the enclosure, in a way symmetrical to basins. This is often not a desired property when these caves (local maxima) are not at the top of *internal* cavities but are connected to the "outside" of O. For example, if O is a thick and hollow cube open at the bottom (Figure 5(a)), then the bottom-up propagation leaves O unchanged (Figure 5(b)),[‡] but the top-down propagation of SIMPLEENCLOSURE completely fills the cube (Figure 5(c)). Then, the wall, which is the outer surface of this cube will not protect the inner sides from oozing plastic.

The space inside the cube was filled since its ceiling was propagated all the way down. Indeed, the mid-height slices of the cube contain an empty square that gets filled by the eroded slice from above.

Our idea is to lessen the effect of such large filling by introducing tears in it.

$$V_{|i} \leftarrow O_{|i} \cup V_{|i+1}^{\downarrow \Delta \tan\theta} \setminus T_{|i}.$$

The tear $T_{|i}$ has to satisfy three properties. First, it has to be thin so as to preserve the printability of the wall, that is, the erosion of $T_{|i}$ by $\Delta \tan\theta$ has to be empty. Second, since the slice $V_{|i}$ has to contain $O_{|i}$, we require that $T_{|i}$ lies in the complement $\overline{O_{|i}}$ of $O_{|i}$. Third, new local minima may appear as a consequence of tearing, so the tear has to be chosen so as to minimize their number. The medial axis of $\overline{O_{|i}}$ is a good candidate: it tears the parts of $V_{|i+1}^{\downarrow \Delta \tan\theta}$ that fall in $\overline{O_{|i}}$ evenly so that the separated pieces on either sides of the medial axis get completely eroded away in approximately the same time. There are several ways to compute a medial axis or skeleton from an input binary image. In practice we use

$$T_{|i} = \textsc{Skeleton}(\overline{O_{|i}})$$

where SKELETON is the integer medial-axis of Hesselink and Roerdink [HR08] with both constant pruning ($\gamma = 10$) and linear pruning (with an upper bound of the cosine of the bisector angle taken as 0.9).

We have found this process to grow pleasant printable voids where the original caves would have been filled (Figure 5(d)). The tearing process has a tendency to slightly increase the number of local minima or overhangs, thereby increasing the complexity of their support structure. For this reason, the tearing process may be enabled or not depending on the user's satisfaction with the generated enclosure. The tearing process is mandatory however in the particular application to hollowing an object out, that we describe in §7.1.

For future reference, we refer to this new top-down propagation as TIGHTDOWNWARDPROPAGATION.

```
function TightDownwardPropagation($O$)
    $V_{|n} \leftarrow O_{|n}$                                        ▷ Initialization of the top slice
    for $i = n-1$ down to 0 do                          ▷ Propagate from top to bottom
        $V_{|i} \leftarrow O_{|i} \cup V_{|i+1}^{\downarrow \Delta \tan\theta} \setminus \textsc{Skeleton}(\overline{O_{|i}})$
    return the volume $V$
```

6. Some implementation details

6.1. The angle θ and the radius of erosion

Let us assume that the filaments are extruded rectangles of height Δ and width ρ. Δ is also the height of each slice. Let $\beta \in [0, \rho]$ be the smallest width of the contact surface between two filaments on top of each other, that guarantees a correct bonding. We derive the angle θ used in (5) from the radius r that we use for eroding the slices: $r = \Delta \tan\theta$. The radius r itself should be as large as possible but not too large as to raise θ above some upper bound $\theta^\star$; so that we compute r as

$$r \leftarrow \min(\rho - \beta, \Delta \tan\theta^\star).$$

In the typical case where $\theta^\star = \pi/4$, $\rho = 0.4\,\text{mm}$, $\Delta = \beta = 0.2\,\text{mm}$, we obtain $\theta = \pi/4$ and $r = 0.2\,\text{mm}$. For a rougher print with $\Delta = 0.3\,\text{mm}$, we get the same value for r but the enclosure walls are more vertical since $\theta = 33.7°$.

Remark. The radius of erosion does not have to be constant over all the slices but can be fine-tuned to different slices, or even different components in a same slice, having different thicknesses. This makes our enclosure usable in tandem with adaptive slicing [PRD95].

6.2. Filament width and small features

Until now we have considered that the thickness of the plastic filament is negligible. Its actual thickness disallows the printing of features of roughly that thickness or smaller. It is important to remove

[‡] except for a pyramid above O.

those features from the slices during the computation of the enclosure, for otherwise, their removal in the later stage of the print-head path planning would remove the support of some slices and reduce the quality of the printed object. For this reason, each freshly computed slice is searched for non-printable boundary paths, which we discard before computing the next slice. The search extracts each loop in the boundaries of a slice and the path-planning code [Lef13] is called to decided whether each loop should be discarded or kept. Finally the slice is updated and the propagation may proceed.

6.3. Slice representation

There are several possible representations for slices, including polygonal and discretely sampled. In our implementation, slices are discrete boolean grids. They are typically computed on the GPU, by recording in-out events during rasterization or ray-casting. We use well optimized CPU procedures for set-wise operations and morphological operations (erosions and skeleton) on slices. We use a horizontal image resolution of 0.05 mm per pixel, which corresponds to the precision of our 3D printers. At this resolution, with the same parameters as above, the slices are eroded with a disk of radius 4 pixels.

Computing an enclosure only takes a few seconds for all the objects shown here.

7. Applications and results

7.1. Application 1: Hollowing a volume out

In this section, we model inner enclosures to maximally hollow the printed object O out while producing printable, ie self-supporting inner cavities. The generated cavities require support only for its few generated overhangs. It can thus be left almost empty, which reduces the quantity of material used for printing and the weight of the printed object. Most importantly, the printing time is significantly reduced.

Computing the cavity. The first step is to erode (in 3D) the object O with a sphere of radius s in order to obtain a first approximation of the cavities. The difference $O \setminus O^{\downarrow s} = O \cap \overline{O^{\downarrow s}}$ is a shell of thickness s. But the boundary of the cavities, $\partial(O^{\downarrow s})$ may not be printable. To remedy this situation, we simply apply the tight top-down propagation of §5.2 to the desired cavities $\overline{O^{\downarrow s}}$. The final, hollowed object is computed as

$$O' = O \cap \textsc{TightDownwardPropagation}\left(\overline{O^{\downarrow s}}\right). \qquad (10)$$

O' has the largest possible printable cavities given the constraint that its outer solid shell should be at least s thick.

Results. We first illustrate hollowing with a model of the comet 67P/Churyumov–Gerasimenko, which exhibits two roughly rounded lobes connected by a neck. The hollowing procedure automatically computes a single large cavity only 0.8 mm from the comet surface—just enough for two extruded filaments on the printers that we use. The walls of the cavity satisfy the slope constraint and present few local minima. The cavity is sufficiently simple that there is no need for scaffolding inside the cavity. The printed model is shown in Figure 7.

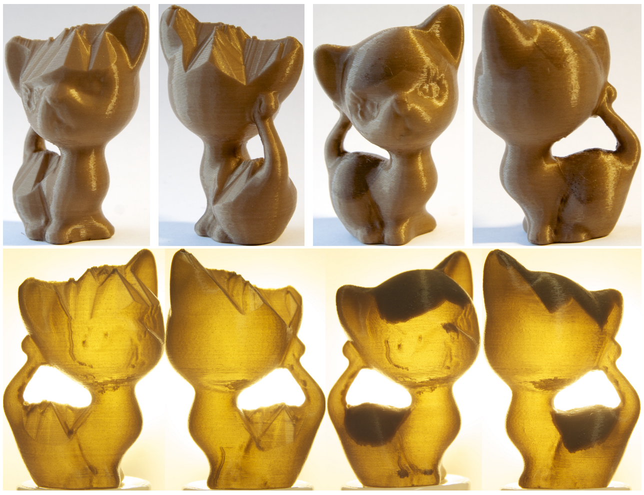

Figure 6: *The kitten.* Top-left. *Front and back view of the cavity.* Top-right. *Front and back view of the hollowed kitten.* Bottom. *Same as above with back light. The cavity is completely empty, while the hollowed kitten is filled between its inner cavity and its surface (eq. (10)).*

Our second example uses the *kitten* model, which we hollow as much as again leaving only a shell 0.8 mm thick. The hollowed out kitten can be printed directly without any support (Figure 6).

Figure 8 shows a third example: the *minotaur* requires a support structure but can be printed mostly empty.

Discussion. The usual way to fill the inside of a printed object is to use a sparse and strong pattern. Our hollowing technique lets us print objects that are mostly empty. Table 1 details the amount of material necessary to print selected models, with and without infill. For most models, it is clear that our hollowing technique sharply reduces print time and material use. Note that the numbers in Table 1 are obtained for a completely filled object above the cavity (eg opaque region in Figure 6, right), but we could further reduce plastic use by using a sparse infill in these regions.

Compared to a standard infill which material usage grows to the cube with object dimensions, our cavities incur a much smaller penalty and print time. Our objects are also guaranteed to print whereas selecting a maximally sparse infill for a given object is difficult and requires tedious trial and error.

7.2. Application 2: Low-complexity scaffolding

In this section, we look at the problem of generating support structures to allow printing of shapes with arbitrarily oriented surfaces and overhanging parts. This is a ubiquitous problem in layer-wise manufacturing. In the remainder we assume that the object is given in its final orientation.

Given an object O we wish to print, we define its partial support $\mathcal{S}(O)$ as

$$\mathcal{S}(O) = \textsc{PropagateDown}(O) \setminus O.$$

	This paper		State of the art	
Object	**Hollow**	**Hollow with our support**	**Infill**	**Infill with bridges**
Comet 67P	1329 (56’53”)	1453 (63’18”)	4238 (152’41”)	4328 (156’51”)
Minotaur	1186 (50’30”)	1576 (72’31”)	2267 (85’28”)	2713 (107’24”)
Kitten	2151 (87’43”)	—	4309 (153’55”)	—
Bear	998 (46’49”)	1469 (74’25”)	991 (46’36”)	1245 (59’15”)

Table 1: *For each object, the required length of plastic filament is shown, in millimeters, together with the* estimated *print time in parenthesis.* ***Hollow*** *corresponds to a cavity 0.8 mm deep (the width of only two filaments) hollowing the object.* ***Hollow with our support*** *is the same with the additional crust, support and scaffolding (§7.2) required to actually print the object.* ***Infill*** *corresponds to the object with one shell and a sparse infill setting of 30 %.* ***Infill with bridges*** *is the same with the additional scaffolding from [DHL14] required to actually print the object.*

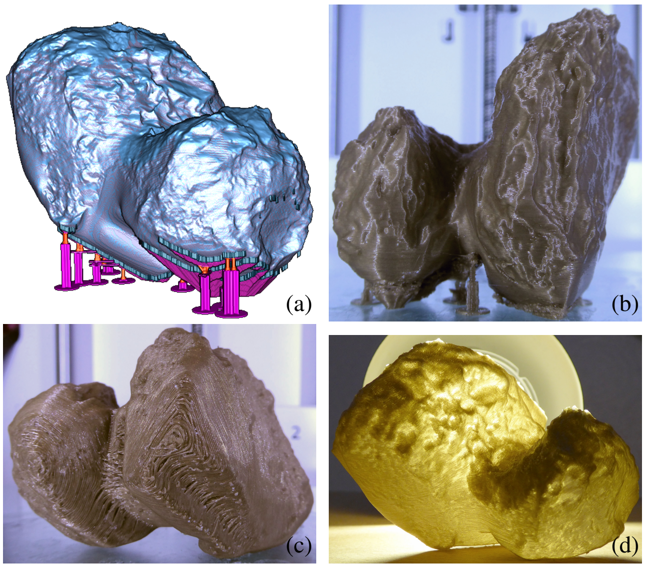

Figure 7: ***The comet 67P.*** (a) *The 3D model. The different shades of blue at the top are locations where the print will be thicker that* 0.8 *mm.* (b) *The print.* (c) *The bottom of the comet after support removal.* (d) *Back-lighting reveals that the comet is almost empty. Credits: the 3D model was made by Mattias Malmer using data from ESA/Rosetta/MPS for OSIRIS Team MPS/UPD/LAM/IAA/SSO/INTA/UPM/DASP/IDA.*

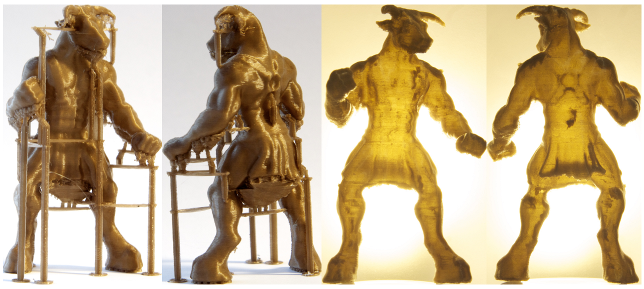

Figure 8: *The minotaur.* Left. *Front and back view of the hollowed minotaur with support.* Right. *Front and back view of the backlit hollowed minotaur after support removal. One can see that the internal cavity has left material only in the feet, calves, hands, shoulders and head.*

$\mathcal{S}(O)$ is a subset of the simple enclosure of O and has the following desired properties as a support structure:

- It provides a dense support of all the downward facing surfaces of O that are not self-supporting.
- It vanishes quickly to form a limited number of small local minima (overhangs). Supporting these ensures that the combination of object O and $\mathcal{S}(O)$ is printable. Since the number of these new overhangs is much smaller than the overhangs of O, the processing work for computing a standard support structure is minimized in time and material usage (see §5.1).

Since $\mathcal{S}(O)$ is not part of O, it should be easily removable from once the printing is complete. One way to make $\mathcal{S}(O)$ detachable is to fabricate it using a dissoluble material. In order to print the object and its support with a single type of plastic filament, we have implemented the following practical alternative, which we illustrate on a toy example in Figure 9:

1. The downward-facing surfaces ($z(n) \leq -\sin\theta$) are extruded downward by a small distance (eg 2 mm), to obtain the *crust* $\mathfrak{C}$. The crust is printed with thin filaments along a very sparse pattern.
2. The partial support of the crust $\mathcal{S}(\mathfrak{C})$ is synthesized and printed with the similar sparse pattern. To make this support strong, its perimeter is also printed with a full filament.
3. A scaffolding is computed to support $\mathcal{S}(\mathfrak{C})$ with the method of [DHL14]. Any other sparse support technique could be used.

Before printing, all the parts of the slices that dot not belong to the object O are slightly eroded. The printed support structure can then be manually detached from the manufactured object with the help of a small tool.

Simplification of complex supports. Some objects have sufficiently complex shapes so as to entail a rather convoluted support $\mathcal{S}(\cdot)$. In order to simplify the structure of the support and further reduce the number of overhangs, we found the following heuristic useful. We replace the downward propagation loop body by:

$$V_{|i} \leftarrow O_{|i} \cup \textsc{Close}\left(V_{|i+1}^{\downarrow\Delta\tan\theta} \setminus O_{|i}, \omega\right),$$

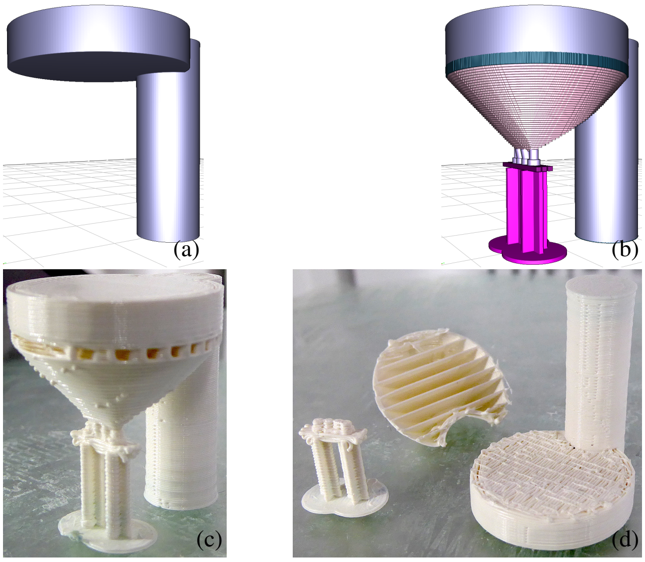

Figure 9: (a) *A toy example printed in the worst orientation for illustration.* (b) *In blue, the crust; in purple, the enclosure supporting the crust; below, the scaffolding.* (c) *The resulting print on the Ultimaker 2 bed.* (d) *The supporting structure has been detached from the object. Note how only two pillars are required, since the enclosure reduces in volume as quickly as possible while being printable.*

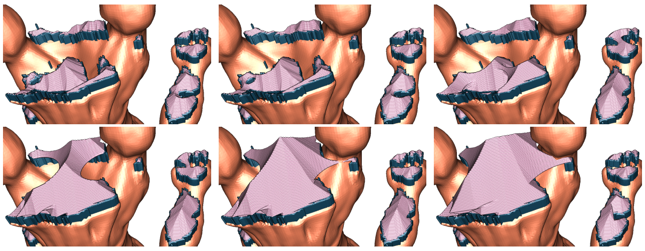

Figure 10: *Effect of* ω *on the shape of the support for the skirt of the minotaur (upside-down view). In reading order,* $\omega = 0, 1, 4, 6, 7, 10$.

where the length ω controls the amount of structural simplification. Figure 10 illustrates the effect of ω on the support of the Minotaur's skirt.

Figure 11 shows more complex examples of support computed with our technique on various models. The quality of the bottom supported surface of the printed object can be visually assessed in Figures 9, 7 and 12.

7.3. Application 3: Protective walls

When printing a dual-color (or bi-material) object, the wall ∂W of the enclosure is printed together with the object O to protect O from other activities surrounding it [HL14]. It is desirable that the wall does not touch O. This is accomplished during the computation of W by working on an object O' whose slices are dilations of the slices of O by some fixed radius, eg $d = 1$ mm: $O'_{|i} = O^{\uparrow d}_{|i}$. Figure 5(c,d) shows an example of such a protective wall. Figure 13 (top) shows the print of a simple enclosure for the dual-color dragon model [§]; the enclosure is two filaments thick, similar to the oozing shield generated by the *Cura* software for dual-color printing. A single filament thick print gives a similar result with a slightly broken surface (bottom).

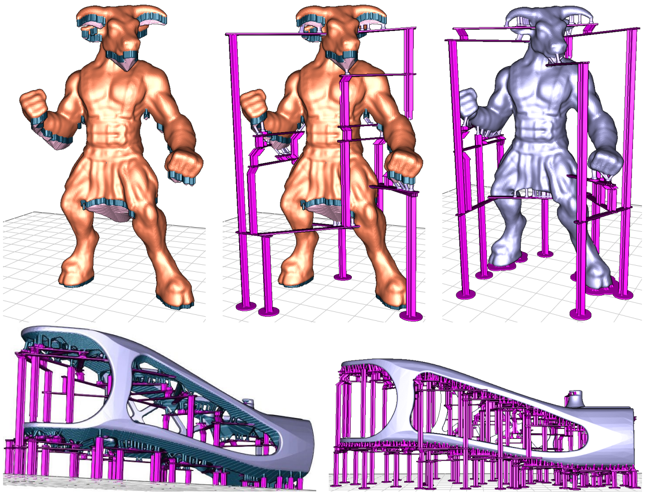

Figure 11: ***The minotaur***. Left. *The crust (blue) and its support (pink).* Middle. *The scaffolding holds the overhangs of the support.* Right. *The scaffolding of [DHL14] computed directly on the minotaur surface. Our technique reduces the overall complexity of the support structure.* ***Poppy's thigh***. Left. *The support and scaffolding generated with our technique forms a simpler structure that is significantly faster to print, for an equivalent final result.* Right. *The scaffolding generated by Dumas* et al.

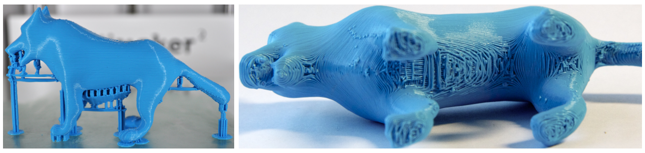

Figure 12: ***The wolf***. *Side view with our support structure and bottom view with support removed. The quality is equivalent to standard single-material support techniques.*

An enclosure that is easy to remove. When the printing is complete, the wall may not naturally separate from the objects it encloses, so that one may have to use scissors to cut the wall open. Alternatively, one might trade a looser enclosure for an easier separation. To do so, only the bottom-up propagation of the slices is performed to obtain $W' = \textsc{PropagateUp}(O')$. The enclosure is then defined as $W = \{(x,y,z) \in \mathbb{R}^3 \mid \exists z' \geq z, (x,y,z') \in W'\}$. The wall ∂W is then a height-field. After printing, the wall is separated from O by simply pulling it upward (Figure 14).

[§] http://www.thingiverse.com/thing:29088

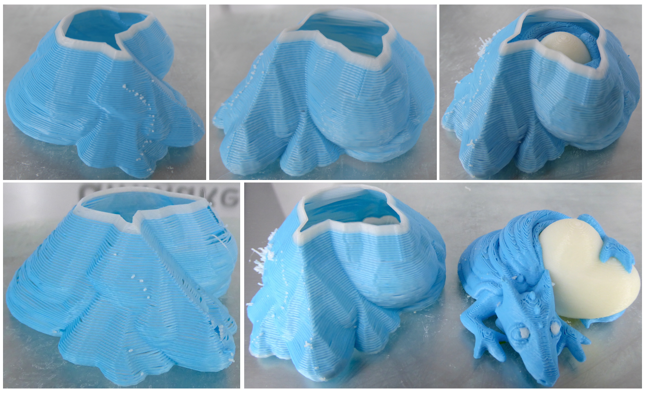

Figure 13: Top. *Back and front of a simple enclosure printed alone. The enclosure is two filaments thick and requires no support structure. Each other layer is of a different color.* Top right. *Same protective enclosure printed with the dual-color dragon model inside.* Bottom left. *This print is a single filament thick.* Bottom right. *The dragon has been taken out of its enclosure.*

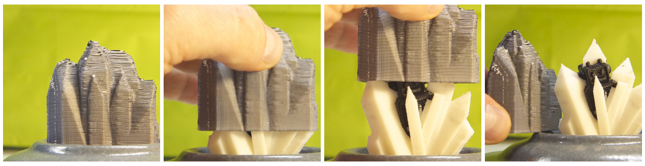

Figure 14: *A height-field enclosure protects a dual-color print and is easy to remove.*

Potential application to contour crafting Contour crafting builds structures made of concrete in a manner very similar to table-top printers, but with a much larger apparatus. The fast construction of buildings is a prime example of the future applications of contour crafting. Obviously, there is no way that concrete can be used to “print” a support during the layer-by-layer erection of the building. We see that our enclosure technique offers a way to automatically infer a structure that is constructible with contour crafting while staying close to the original design.

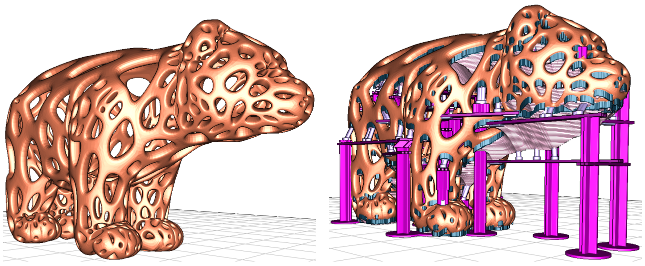

Figure 15: *The support structure for the bear model.*

8. Limitations

For certain types of object, support-structures can become difficult to remove —a limitation is shared by many techniques. For the *bear* model, for example (see Figure 15 and table 1), the support structure is too large with respect to the overall thinness of the model; it would further be very difficult to remove if not printed in dissoluble material. There is no material gain in hollowing the model, again because of the thinness of the fine structures of the bear.

9. Conclusions

We have described how to model enclosures around objects that can be printed with little support. We have shown three applications of enclosures.

1. The modeling of large inner cavities inside an object is perhaps the most important application. It significantly lowers the amount of material required to print the object and the time required to print it.
2. The modeling of support enclosures, that quickly vanish while ensuring a reliable support and being easy to separate from the object.
3. The modeling of tight protective walls to improve the quality of dual-color prints.

Our geometric constraints tend to be conservative, and in rare cases deviating from them can still result in successful prints (the kitten for instance may print without external support). An important benefit is that our technique never resulted in failed prints for any of the three applications. In a context where print failures incur a large overhead in material, print time and cleaning costs, this is we believe a major advantage.

Our technique likely also has potential for stereolithography printers but this remains to be tested. In particular, our approach is possibly overly conservative in this context, given that much larger overhangs are admissible when printing on resin printers, and the overhangs size may vary depending on the slice geometry.

Acknowledgments

We thank Marc Guyonnet and Yann Hortance who implemented the early drafts of the techniques presented here. This work was funded by ERC grant ShapeForge (StG-2012-307877).

References

[DHL14] Dumas J., Hergel J., Lefebvre S.: Bridging the gap: Automated steady scaffoldings for 3d printing. *ACM Transactions on Graphics 33*, 4 (July 2014), 98:1–98:10. URL: http://doi.acm.org/10.1145/2601097.2601153, doi:10.1145/2601097.2601153. 2, 5, 8, 9

[Hei11] Heide E.: Method for generating and building support structures with deposition-based digital manufacturing systems, 07 2011. US Patent 20110178621 A1. 2

[HL14] Hergel J., Lefebvre S.: Clean color: Improving multi-filament 3d prints. *Computer Graphics Forum 33* (2014). 2, 9

[HLZCO14] Hu R., Li H., Zhang H., Cohen-Or D.: Approximate pyramidal shape decomposition. *ACM Transactions on Graphics 33*, 6 (2014). 2

[HR08] Hesselink W. H., Roerdink J. B.: Euclidean skeletons of digital image and volume data in linear time by the integer medial axis transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence 30*, 12 (2008), 2204–2217. doi:http://doi.ieeecomputersociety.org/10.1109/TPAMI.2008.21. 6

[HYW*09] HUANG X., YE C., WU S., GUO K., MO J.: Sloping wall structure support generation for fused deposition modeling. *The International Journal of Advanced Manufacturing Technology 42*, 11-12 (2009), 1074–1081. 2

[LBRM12] LUO L., BARAN I., RUSINKIEWICZ S., MATUSIK W.: Chopper: Partitioning models into 3D-printable parts. *ACM Transactions on Graphics 31*, 6 (Dec. 2012). 2

[Lef13] LEFEBVRE S.: IceSL: A GPU accelerated modeler and slicer. In *18th European Forum on Additive Manufacturing* (2013). 7

[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. 6

[PWLSH13] PRÉVOST R., WHITING E., LEFEBVRE S., SORKINE-HORNUNG O.: Make It Stand: Balancing shapes for 3D fabrication. *ACM Transactions on Graphics 32*, 4 (2013), to appear. 2

[RCM*14] REINER T., CARR N., MECH R., STAVA O., DACHSBACHER C., MILLER G.: Dual-Color Mixing for Fused Deposition Modeling Printers. *Computer Graphics Forum 33*, 2 (2014), 479–486. 2

[SHEE13] STRANO G., HAO L., EVERSON R., EVANS K.: A new approach to the design and optimisation of support structures in additive manufacturing. *The International Journal of Advanced Manufacturing Technology 66*, 9-12 (2013), 1247–1254. 2

[SVB*12] STAVA O., VANEK J., BENES B., CARR N. A., MECH R.: Stress relief: improving structural strength of 3d printable objects. *ACM Transactions on Graphics 31*, 4 (2012), 48. 2

[TJ11] TELEA A., JALBA A.: Voxel-based assessment of printability of 3d shapes. In *Mathematical Morphology and Its Applications to Image and Signal Processing: 10th International Symposium, ISMM 2011, Verbania-Intra, Italy, July 6-8, 2011. Proceedings* (Berlin, Heidelberg, 2011), Soille P., Pesaresi M., Ouzounis G. K., (Eds.), Springer Berlin Heidelberg, pp. 393–404. URL: http://dx.doi.org/10.1007/978-3-642-21569-8_34, doi:10.1007/978-3-642-21569-8_34. 2

[VGB14a] VANEK J., GALICIA J. A. G., BENES B.: Clever support: Efficient support structure generation for digital fabrication. *Computer Graphics Forum 33*, 5 (2014), 117–125. 2

[VGB*14b] VANEK J., GALICIA J. A. G., BENES B., MĚCH R., CARR N., STAVA O., MILLER G. S.: Packmerger: A 3d print volume optimizer. *Computer Graphics Forum 33*, 6 (2014), 322–332. 2

[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), Article 177: 1–10. 2