Automatic Surface Segmentation for Seamless Fabrication Using 4-axis Milling Machines

S. Nuvoli[1] A. Tola[1,4] A. Muntoni[2] N. Pietroni[3] E. Gobbetti[4] and R. Scateni[1]

[1]Università di Cagliari, Italy
[2]ISTI-CNR, Italy
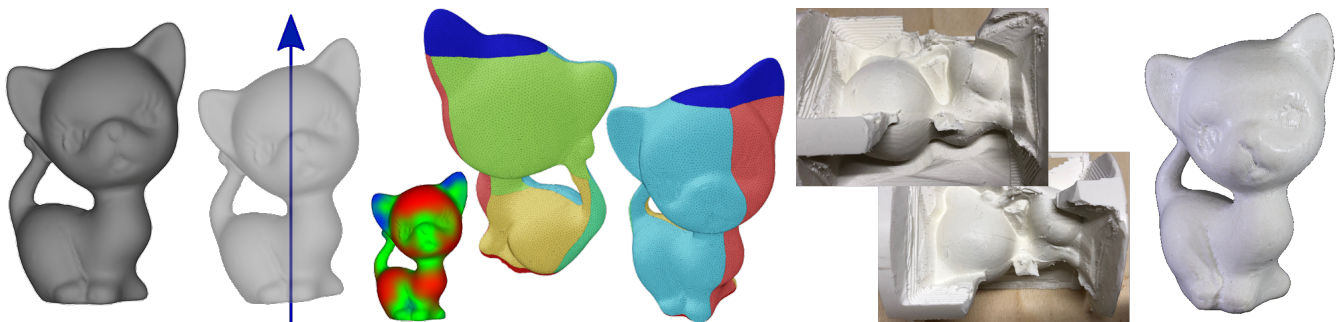[3]University of Technology Sydney, Australia
[4]CRS4, Italy

Figure 1: *Our processing pipeline in a nutshell (left to right): we first compute the rotation axis on the input mesh; we then partition the mesh in millable height-field portions (including the top and bottom ones) taking also into account the information of the saliency map; we compute a milling sequence and fabricate the object using the 4-axis milling machine; we clean up the result to obtain the final real object.*

Abstract

We introduce a novel geometry-processing pipeline to guide the fabrication of complex shapes from a single block of material using 4-axis CNC milling machines. This setup extends classical 3-axis CNC machining with an extra degree of freedom to rotate the object around a fixed axis. The first step of our pipeline identifies the rotation axis that maximizes the overall fabrication accuracy. Then we identify two height-field regions at the rotation axis's extremes used to secure the block on the rotation tool. We segment the remaining portion of the mesh into a set of height-fields whose principal directions are orthogonal to the rotation axis. The segmentation balances the approximation quality, the boundary smoothness, and the total number of patches. Additionally, the segmentation process takes into account the object's geometric features, as well as saliency information. The output is a set of meshes ready to be processed by off-the-shelf software for the 3-axis tool-path generation. We present several results to demonstrate the quality and efficiency of our approach to a range of inputs.

CCS Concepts

• *Computing methodologies* → *Mesh geometry models;*

1 Introduction

Digital fabrication technologies, which aim at producing physical objects from their digital representation, have significantly progressed in recent years. Thanks to the coupling between the increase in automation, accuracy, and flexibility and the reduction in reproduction costs, applications of digital fabrication are booming in many areas [NKI*18].

The most common technologies for physically producing 3D shapes can be broadly subdivided into two classes: *Additive Manu-*

facturing techniques (*3D printing*), which build objects by adding material layer by layer, and *Subtractive Manufacturing* techniques (*CNC machining*), which construct objects by cutting material away from a solid block.

Additive technologies have the significant advantage that they decouple the manufacturing process from the fabricated object's geometric complexity. Computer graphics and geometry processing boosted the use of 3D printing in multiple contexts and have recently explored its limits for the production of complex functional

shapes [SEPC16]. The increasing availability of inexpensive 3D printers has widened the user base, leading to a booming market.

While 3D printing may be considered optimal for small-scale production or rapid prototyping [NKI*18], industrial productions often require the manufactured products to be produced in large numbers and to comply with physical constraints that are inconsistent with additive technology (e.g., robustness or usage of natural materials such as wood or stone). Contrarily to 3D printing, CNC machining is usually faster and capable of producing massive parts using a wide range of materials. Moreover, since shapes are carved out from solid blocks of material, the resulting objects are structurally more robust than those created with a layer-depositing process. Therefore, subtractive manufacturing remains the dominant process in industrial settings.

Unfortunately, subtractive manufacturing has strict geometric constraints that impose severe limitations on the class of fabricated objects. In particular, traditional 3-axis CNC machining requires the manufactured object to be a height-field. Consequently, the mass production of intricate shapes is challenging, if not impossible, since it needs an expert to plan, then manually adjust the object's orientation to accommodate the different working directions. For these reasons, considerable effort has been spent on extending the fabrication DOFs. However, this task often requires expensive machines (e.g., 5-axis milling tools or robotic arms) controlled by complex planners [Tan14], making the user base of these solutions considerably narrower from that of 3D printing.

In this context, 4-axis CNC milling has a particular position in the spectrum of fabrication technology. It operates on the same axes of a 3-axis machine but also includes the rotation around one of the axes. By enabling the rotation of the volume between each carving operation, it supports the manufacturing of a much broader range of forms at a buying and maintenance expense that is only marginally higher than that of the classical 3-axis solutions. For this reason, the industry has been looking at 4-axis solutions to replace costlier production processes of specialized components, e.g., spiral bevel gears used in transmission systems [SJHC01]. Unfortunately, the few solutions that exist in the state-of-the-art are not capable of producing general complex shapes, forcing the extensive use of manual interventions (see Sec. 2).

In this article, we introduce a novel pipeline to fabricate complex shapes from a single block of material using a 4-axis CNC machine. We first devise the rotation axis that maximizes the surface's orthogonality to the milling directions while fitting the object in the fabrication workspace (i.e., the raw material block). Then, we fabricate the object's largest area by milling towards a sequence of directions orthogonal to the rotation axis. Finally, we mill the extremities secured to the rotation axis. To derive the optimal set of directions, we segment the object into multiple height-fields, which are then ordered into a milling sequence. The subdivision jointly optimizes individual height-field approximation quality, boundary smoothness, and the total number of patches while avoiding unwanted collisions between the milling tool and the surface. Additionally, we exploit saliency information (automatically-generated or user-provided) to avoid possibly visible seams in highly-detailed or semantically meaningful surface regions.

Our main contribution is a novel integrated, effective, and prac-

tical optimization scheme that extends height-field decomposition approaches to 4-axis fabrication. By exploiting the extra rotation axis, we extended the range of shapes that can be manufactured to a single solid piece of material, making subtractive fabrication an appealing solution for a wide range of users. In contrast to previous techniques that split objects into multiple components that are assembled afterward, our method produces seamless and robust shapes (see Sec. 4). Furthermore, by reducing 4-axis fabrication to a sequence of height-field milling processes, we can directly use any of the many 3-axis fabrication drivers available in the industry, making our solution immediately applicable in practice.

2 Related work

There is vast literature on CNC machining in Computer-aided design and mechanical engineering. Most of that research focuses on optimizing the tools and their path for the milling process and does not emphasize the surface decomposition. The task of planning the global carving process, and ensure the tool accessibility, is usually allocated to the experts. The increase in modeling capabilities and the resulting complexity of available 3D shapes made this manual task harder. Many computer graphics literature studied how to automatically decompose the surface or the milled volume into parts that are suitable for CNC fabrication.

2.1 Height-field surface decomposition

A common strategy to overcome the limitations of CNC machines is to decompose the object into multiple height-fields. A common approach is to separately fabricate each height-field by 3-axis CNC machining and assemble the pieces afterward. One of the first methods that followed this idea is the decomposition proposed by Alemanno et al. [ACP*14]. However, this method requires the user to specify the surface decomposition manually. Since it does not consider the assembly process, its usage is mostly limited to the simple cases of 3D-assembled low-reliefs. Most of the literature on height-field decomposition for fabrication is related to moldable pieces. Herholz et al. [HMA15] decompose the surface into different height fields, defining a rigid mold's components. This approach allows the user to extract the casted objects. Flexmolds [MPBC16] and Metamolds [AMG*18] use a relaxed visibility constraint to produce a more general patch decomposition that involves the use of deformable molds. While the task of planning a casting and a milling process have significant overlap (they both strive for high-field patch arrangements), the latter requires special care to guarantee that the tool can access the carving area at any moment during fabrication and that the milling angles produce good quality carvings.

2.2 Volume decomposition

Several methods consider the volume of the fabricated object during the optimization process [SEPC16]. Most of the techniques that use additive manufacturing decompose the object into multiple components that can fit the 3D printer's workspace. Chopper [LBRM12] decomposes the model into pyramidal parts that minimize the use of supports [HLZC14]. Dapper [CZL*15] optimizes the packing of the parts in the printing volume. Recently, Filoscia et al. [FAG*20] proposed a new method to decompose a 3D-printed object to minimize the seams' visual impact. However, these methods are not directly

related to the planning of subtractive fabrication. The decomposition strategy proposed by Araujo et al. [ACA*19] guarantees an assembly sequence. Regarding molding, composite molds [AMG*19] are the only ones that consider explicitly the volume surrounding the casted object.

Most of these methods aim for height-field approximations or consider the volumetric access to allow for assembly. However, they are designed for additive fabrication or casting, while subtractive techniques pose additional challenges. Since the milling tool carves the object from the outside toward the inside instead of merging planar layers, extra constraints should guide the milling angles and secure the milling tool's accessibility. Fanni et al. [FCM*18] proposed a decomposition method based on polycube mappings and analyzed the manufacturability of their decomposition for additive and subtractive techniques, taking into account both 3- and 4-axis milling. Since their method does not consider particular constraints relative to subtractive techniques, their final decompositions prove feasible in practice just for additive techniques. The approach proposed by Muntoni et al. [MLS*18] decomposes, instead, a 3D object into height fields, then projects the decomposition toward the interior, covering the entire volume and, at the same time, ensuring each piece to be manufacturable with 3-Axis CNC machines. DHF-Slicer [YAV*20] replaces the single height-field decomposition with a more general double-height-field one, decomposing objects into small sets of bounded-height 3-axis millable parts (slices). The method is fully automatic and, given similar milling precision and slice height constraints, produces coarser decompositions than previous approaches [MLS*18]. However, in all these decomposition methods, the user must assemble the pieces after fabrication to obtain the final object, leading to visible seams and reduced robustness.

To avoid decomposing the object into multiple parts, DSCarver [ZZX*18] uses a (3+2) axis milling machine that acts like a traditional 3-axis milling machine during carving but supports two-degree-of-freedom (2-DOF) rotation of the drilling tool between each milling pass. DSCarver automatically plans and optimizes the different phases and rotation angles by covering the input surface with a minimum number of accessible regions and then extracting a set of machinable patches from each accessible area. Thanks to the extra degrees of freedom, this method can produce significantly complex objects in a single piece, including high-genus ones. Their approach, however, requires full exploitation of the 3+2 DOFs. For a similar setup, VDAC [MYZ*20] strives, instead, to jointly optimize setup and path planning by focusing on minimizing both the number of setup directions required for 3-Axis CNC milling and the number of carving path transitions/repositions while giving priority to the former. While DSCarver focuses primarily on accessibility analysis, VDAC targets rough-stage machining, where the carving starts from the initial material block and stops close to the target object. Our work considers a much more constrained situation of these 3+2 approaches. We need to find a primary axis and directions orthogonal to it to minimize the setup phases on a 4-axis machine. In this context, Frank et al. [FWJ06] have proposed, given a user-determined main axis, to carve the object by slicing it into layers and producing 360-degree tool paths to machine each 2D layer taking into account visibility. However, the process tends to generate many rotations and proves practical only for objects that can be approximated with a small number of slices. We significantly improve this class of ap-

proaches by automatically determining an axis, reducing the number of rotations through height-field decomposition, and producing a complete object through three-region segmentation.

3 Automatic fabrication planning

A 4-axis CNC machine extends the classical 3-axis CNC device with an extra degree of freedom corresponding to the object's rotation around an axis. In a typical manufacturing setup, as illustrated in Fig. 2, the carved volume (called *stock*) is fixed to a rotating shaft through a solid flat base. In this kind of setup, manufacturing the two extremes of the object is not trivial since both extremes of the stock have to be robustly fixed to the rotating shaft during the whole fabrication, as shown in Fig. 2. For security reasons and to ensure fabrication stability, the drill bit should not approach the shafts. Thus, the fabrication of a 3D object starting from a solid block of material must involve at least three phases: while we can fabricate the large majority of a shape by exploiting the different milling directions provided by the rotation axis, two additional fabrication phases are necessary to carve the top and bottom regions.
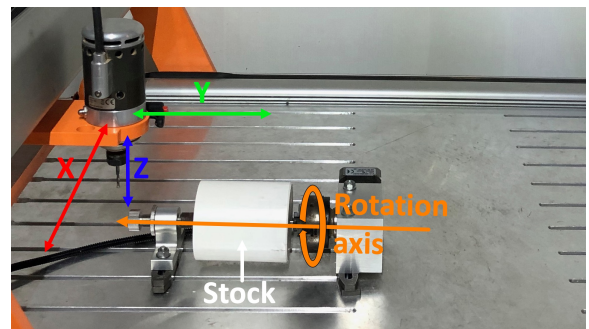

Figure 2: *Our fabrication setup. The milling tool on the left has three degrees of freedom in the operational field of the machine; the rotating shaft adds the fourth one.*

Our fabrication approach assumes that at least one of the two extremes of the stock is a flat base; this is a reasonable requirement since stocks used for 4-axis machining are generally cylinders. The top and bottom regions' manufacturing is simple 3-axis milling passes: the first one is performed by fastening the block on its flat base. In contrast, for the second, a mold is used to properly place the block and to guarantee a proper alignment in the process. Note that, without these two phases, we would not be able to carve a portion of the block. Fig. 3 illustrates the entire process.
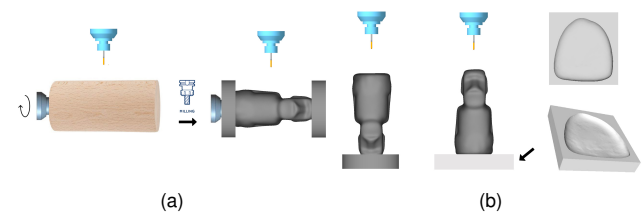

Figure 3: *Fabrication process outline. We first carve the set of height-fields on the side of the model from the raw block of material (a); then, we unpin the block from the rotation tool, and we mill the two height-fields (top and bottom) (b). Notice that we use a mold (colored with lighter grey) to keep the model correctly in place for milling the second between top and bottom.*

Our automatic fabrication planning method aims at finding the most efficient and accurate plan to organize the milling process. We decompose the fabrication process into a sequence of 3-axis milling processes. Each fabrication stage produces a suitably oriented height-field representation of a small surface region. In particular, for a given rotation axis α, we partition the object into three different regions: a *side* region, a *top* region, and a *bottom* region. We then approximate each of the bottom and top parts as a single height field whose elevation is aligned with α, and the side region as a set of height fields whose elevations are orthogonal to α. Hence, our automatic fabrication planning involves the following steps:

- We pre-filter the geometric detail of the object to produce a regularized representation suitable for further processing (Sec. 3.1).
- We find the rotation axis α that produces the best height-field approximations in the three regions (top, bottom, and side). Intuitively, in this optimization process, we favor the orthogonality of the fabricated surface to the resulting milling directions (Sec. 3.2).
- Given the rotation axis α, we segment the model into a set of non-overlapping height fields. We first derive the top and bottom regions that can be fabricated as a single height field. Using a graph-cut algorithm, we segment the remaining side regions into a set of height fields that can be fabricated by rotating around α, taking into account fabricability, efficiency, and accuracy criteria (Sec. 3.3).
- We recover the details lost in the pre-filtering phase (Sec. 3.4).
- We produce the final fabrication plan, determining the order of fabrication of the individual charts and the process to produce each of them (Sec. 3.5).

Details on each of these different steps are provided in the following sections.

3.1 Prefiltering

Since a rough meshing can cause several problems to segmentation algorithms, we initially re-mesh the input shape to a dense and regular uniform tessellation. Additionally, high-frequency details might induce excessive fragmentation in the final height-field decomposition. To tackle this issue, we use the same approach as Muntoni et al. [MLS*18], which allows users to sacrifice the fidelity to the input to enable a more natural and faster fabrication. We first remove high-frequency surface details using the low-pass Taubin filter [Tau95]. Then, after the segmentation, we reintroduce high-frequency features while enforcing the fabricability constraints (see Sec. 3.4 for more information). This pre-processing step is optional and controlled by the user. Fig. 5b shows the effect of prefiltering.

3.2 Determining the best overall milling orientation

The choice of the rotation axis α has a massive impact on the quality of the final segmentation and fabrication (see Fig. 4 for an illustration of the effect of selecting two different rotation axes). Therefore, the first step of our method is the selection of the axis that provides the best fabrication accuracy while tightly fitting the model inside the stock. To do so, we must solve two problems. First of all, we must determine how a given rotation axis partitions the surface into the side, top, and bottom fabrication regions (Sec. 3.2.1). Then, we must foresee how a given partitioning will impact the overall fabrication accuracy (Sec. 3.2.2).
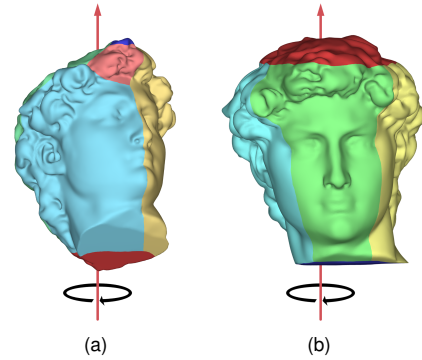

Figure 4: *Effect of axis selection. The choice of the axis in (a) generates height-fields of varying depth and size, thus complicating tool-path creation; the axis shown in (b) produces height-fields of similar size and depth, even if their number is reduced by one; moreover, (b) has better top and bottom regions to three-phase machining. Axis (a) is the principal inertia axis, while axis (b) is the axis selected by the optimization method of Sec.3.2.2.*

3.2.1 Partitioning into the top, bottom, and side regions

As we previously stated, given a rotational axis, we should partition the input geometry into three regions: the top and bottom areas (fixed to the rotation axis) and the side region (milled by exploiting the extra rotation degree of freedom). For a given rotation axis α, we fabricate the top and bottom regions along with the directions +α and −α. Given an input 3D mesh $\mathcal{M}$, we search for two disjoint top $\mathcal{P}_{top} \subset \mathcal{M}$ and bottom $\mathcal{P}_{bottom} \subset \mathcal{M}$ regions. A face $f_i \in \mathcal{M}$ is *visible* along a direction d if the dot product between face normal n_i and d is less than zero and it is not occluded along d. This visibility definition allows taking into account simultaneously for fabricability and access to the milling tool.

Given a rotation axis α, the top $\mathcal{P}_{top}$ and bottom $\mathcal{P}_{bottom}$ regions should be composed by the largest set of connected faces which are *visible* along −α and +α respectively. Additionally, those regions should also be as close as possible to the shape extremities along the rotation axis. Thus, we first initialize the top and bottom regions using the extremity faces (i.e., top-most or bottom-most face) along the rotation axis. We then iteratively grow each region until it reaches the visible face having the largest distance along the rotation axis. The remaining faces compose the side region. Notice that, given a rotation axis, it is always possible to select two extremity faces to initialize the top and bottom regions. Notice also that, by definition, the three regions are disjoint. Fig. 5 shows the resulting decomposition.

3.2.2 Finding the best axis

Determining a provably optimal axis, in theory, would require the evaluation of all the effects it has on fabrication. This is, however, very costly and could lead to repeatedly segment the model for several orientations, as the objective function can have many local minima (see Sec. 3.3). To reduce the solution space to a manageable size for non-trivial and possibly massive surfaces, we explicitly designed our approach as a pipeline, separating the various optimization steps into sub-problems solved sequentially. Thus, we propose a more efficient approximate solution that decouples axis selection from segmentation, rapidly determining a good, if not provably optimal, axis before the segmentation using only geometric measures on the mesh.
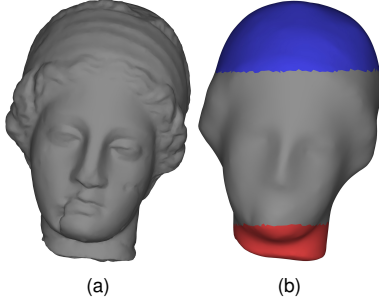
Figure 5: *Filtering and decomposition into regions given an axis. (a) Original model. (b) Smoothed model with top height field in blue, bottom height field in red, and side region in gray,*

Intuitively, the rotation axis that will produce the highest quality is the one that maximizes the overall alignment of the fabricated surface normal with the milling directions. Using the straightforward approach of selecting the principal inertia axis, e.g., as in (Fig. 4a), would work only for elementary symmetric shapes since such a choice does not take into account the fact that we must produce our objects in three phases, each one having different alignment constraints. Therefore, we compute an estimation of the alignment's quality by using different terms for each milling region.

For each of the top and bottom regions, where milling is aligned with the axis direction, given a candidate rotation axis d, we compute the alignment as the sum of the dot product between face's normal and the rotation axis, respectively:

$$A_{top}(d) = \sum_{f_i \in \mathcal{P}_{top}} a_i * (n_i \bullet d) \tag{1}$$

$$A_{bottom}(d) = \sum_{f_i \in \mathcal{P}_{bottom}} a_i * (n_i \bullet -d) \tag{2}$$

where a_i and n_i are the respectively the area and the normal of the face f_i.

Instead, for the remaining faces on the side, we must favor the orthogonality of the face normals to the rotation axis. This copes with the fact that the milling machine will work on a shaft, which will be orthogonal to the rotation direction.

$$A_{side}(d) = \sum_{f_i \in \mathcal{P}_{side}} a_i * (1 - |n_i \bullet d|) \tag{3}$$

We then choose the best axis direction $d^\star = \text{argmax}_d A_{top}(d) + A_{bottom}(d) + A_{side}(d)$. While the problem could be solved using continuous methods, e.g., coordinate descent or ADMM, for generality and ease of implementation, we use in this paper a simpler approach, in which we choose the best direction among a large set of candidate direction $d_1, ..., d_k$ uniformly distributed on a hemisphere using a Fibonacci sphere algorithm [SJ06] ($k = 2000$ in this paper). Using this sampling approach simplifies the future introduction in the optimization of further approximate terms to consider other currently neglected aspects, such as saliency, visibility, smoothness, and detail recovery.

3.3 Determining the optimal decomposition into height fields

Once we establish the rotation axis, we can partition the input geometry into the three regions using the method of Sec. 3.2.1. The top and the bottom areas are height fields fabricated in a single pass by milling along the rotation axis's primary direction. However, the side area is a complex surface whose fabrication requires to rotate the object during fabrication. While the space of the possible milling directions for the side region is composed of all the directions which are orthogonal to the rotation axis, to plan an efficient and accurate milling process, we should promote the formation of large smooth patches, as they support a continuous motion of the milling tool [ZZX*18]. Hence, this segmentation process's main task is to choose an optimal subset of directions and, at the same time, associate each direction with a surface patch composed of a set of faces. The patches constitute a partitioning of the side region, and they should not overlap. This class of problems can be efficiently solved using a multi-label graph-cut optimization [BVZ01] using discretized milling directions as labels. In principle, such a decomposition should take into account both fabrication quality and specific fabrication constraints. Still, this approach would require the incorporation of complex accessibility computations into the segmentation process. To make the approach scalable to massive and complex surfaces, we apply a pipeline approach. We segment the surface based on its geometric characteristics. Then, we produce a fabrication plan given the different charts.

3.3.1 Segmentation

We sample a set of uniformly generated fabrication directions $\mathcal{L}$ orthogonal to the chosen rotation axis (we used 120 possible directions in all the examples shown in this paper) and we define a label for each direction. We have to find a labelling function $\ell: \mathcal{P} \to \mathcal{L}$ that assigns a direction (label) to each face in $\mathcal{P}$. The graph-cut algorithm derives the optimal labeling by minimizing a function:

$$\text{argmin}_\ell \sum_{f \in \mathcal{P}} D(f, \ell) + \sum_{(p,q) \in E} S(p, q, \ell)$$

where $D(t, \ell)$ is the fidelity (or data) term and describes the cost of assigning the fabrication direction ℓ to the face f. Intuitively, this term tends to label each face in its optimal direction. $S(p, q, \ell)$ is the smooth term and describes the cost of assigning a given label to two adjacent faces p and q. This term favors forming a compact and smooth patch layout and holds the total number of patches.

The fidelity term should consider how much the face normals are aligned with the milling direction. Secondarily, we should avoid associating a face to a direction along which it is occluded by other object portions; indeed, that will prevent the milling tool from reaching the target point in the volume during the milling process. The visibility is precomputed per-face, per-direction using collision detection accelerated by an AABB Tree structure. In case of collision, we simply assign an infinite cost. We define the fidelity term as:

$$D(f, \ell) = \begin{cases} 1 - n_f \bullet \ell(f) & \text{if } f \text{ is visible from } \ell(f) \\ \infty & \text{otherwise} \end{cases} \tag{4}$$

where n_f is the normal of the triangle.

The smoothness term $S(p, q, \ell)$ minimizes the overall boundary lengths and determines the shape and location of the boundary between one patch and the others.

Since patch boundaries are the regions where one milling sequence is interrupted and then continued from another orientation, to improve the manufacturing process's quality, we should concentrate them in areas with low detail. In order to take into account this factor, we assume that our mesh is enriched by *saliency* information, providing at each face a measure of regional importance [LVJ05] (see Sec. 3.3.2). We define the smoothness term as:

$$S(p,q,\ell) = \begin{cases} c + d\,\frac{\mathscr{C}(p)+\mathscr{C}(q)}{2} & \text{if } \ell(p) \neq \ell(q) \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where c is the compactness term and d is the saliency factor. The compactness term c, influencing the number of patches, controls the smoothness of the segmentation. The saliency factor d is multiplied by the average saliency of the concerned faces, which is higher if the face is considered perceptually more critical. It prevents the formation of boundaries that separate a salient area (see Sec. 3.3.2).

3.3.2 Saliency computation

Our algorithm can use any per-face saliency factor to encode semantic and geometric information. It can be the result of automatic, semi-automatic, or interactive processes. In our framework, we have implemented both a fully automatic method and an interactive tool.

For automatic saliency computation, we start by computing at each vertex the multi-scale saliency metric $\mathscr{C}(v)$ proposed by Lee et al. [LVJ05], dividing it by the number of scales to normalize the value between 0 and 1. Since the multi-scale approach tends to localize the high frequencies in small areas (Fig. 6a), our final saliency field is found by propagating these peak values to their neighborhood through a diffusion process. In practice, we first propagate the maximum value of each vertex v to its neighborhood N_v, repeating this step a fixed number of times (Fig. 6b), and then perform a Laplacian smoothing (Fig. 6c). For this paper's results, our experiments led us to apply five propagation and 20 smoothing steps. We finally compute the saliency $\mathscr{C}(f)$ for each face as the arithmetic means of the value at its vertices.

We also support user-driven saliency determination by letting the user interactively select/unselect with a brushing tool the areas of interests (Fig. 6d). This makes it possible, in particular, to have users mark semantically essential regions.
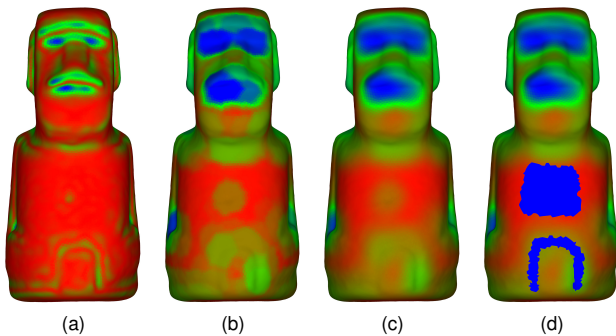
in the segmentation, providing quality in the segmentation in a fully automatic way. The last column presents the two final segmentations used to manufacture the Batman and David models.

3.3.3 Charts optimization

Once we have our per-face segmentation ℓ, we can derive the pairwise disjoint charts $C_1, C_2, C_3, ... C_n$. Each chart $C_i \subset P$ is composed of a connected set of faces assigned to a fabrication direction in $\mathcal{L}$. Note that there could be multiple charts associated with the same direction. The objective function for graph-cut segmentation favors forming a small number of charts with regular boundaries and usually avoids creating charts with multiple boundaries.

However, even if we use a large compactness term, we might have small isolated charts, mostly due to occlusions in complex shapes. To improve the fabrication process's efficiency, we delete charts with an area of less than a given threshold (in our experiments, we used 1% of the total shape area). When we delete a chart, we iteratively re-assign the faces to the best adjacent charts, starting from the border and propagating in the interior. This strategy can assign some face to a direction from which it is non-visible. In this case, we sacrifice the exact fidelity of the shape to allow a more straightforward fabrication. Notice that this step, similarly to pre-filtering (Sec. 3.1), is optional and controllable by the user through the tuning of the merging threshold.

The smoothness of the chart boundaries, moreover, plays a crucial role in a high-quality result. Even if the graph-cut usually produces acceptable results, the boundaries might depend on the initial tessellation (Fig. 8a). Therefore, we perform a Laplacian smoothing of the boundary lines, re-projecting the coordinates in the shape's tangent space and refining the original triangle mesh. Fig. 8 shows the effect of this process.

3.4 Detail recovery

Once we have obtained the decomposition for each chart, we reintroduce the high-frequency details lost during pre-filtering (Sec. 3.1)
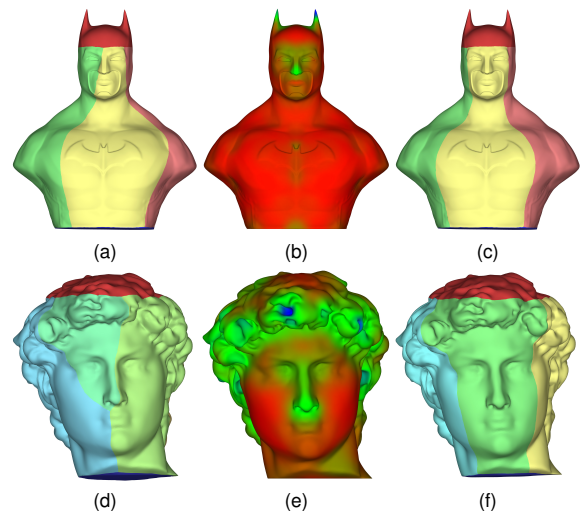
(a) (b) (c)

(d) (e) (f)

Figure 7: *Effect of the saliency term on the segmentation. From left to right: segmentation without considering saliency, final saliency values, segmentation with saliency.*
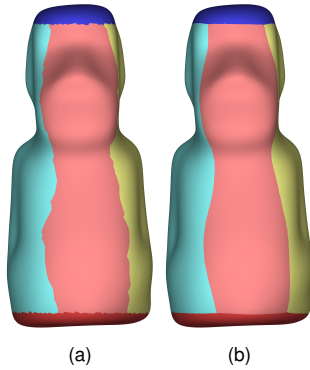
(a) (b) (c) (d)

Figure 6: *Saliency field computation. We show how we generate mesh saliency for the Moai model. The salience values are between 0 (red), and 1 (blue).*

Fig. 7 shows that the saliency term makes an enormous difference

Figure 8: *Boundary smoothing process. (a) Original segmentation. (b) Segmentation after boundary smoothing.*

by using Laplacian surface reconstruction framework [Sor06]. As in Muntoni et al. [MLS*18], we constrain the reconstruction with height-field constraints that ensure fabricability, without taking into account that high-frequency detail reintroduction could cause some triangles to be no longer visible due to slight occlusions (Fig. 9). Again, we sacrifice the exact fidelity of the shape to simplify the process. The tool-path generation software will automatically produce the closest fabricable shape.

Figure 9: *Occlusion effects. Black faces represent triangles no longer visible from the spindle after restoring the high frequencies.*

3.5 Generating the fabrication sequence

Our optimization pipeline allows us to use 4-axis machines by partitioning the fabrication into a sequence of 3-axis milling processes. Thus, we can easily use any available 3-axis fabrication driver to compute the tool-path for each height-field.

The physical fabrication of objects composed of many charts poses, however, extra considerations. As shown in Fig. 10a, depending on the depth of the surface to mill, the drill bit could not reach that depth without tool collisions. For each chart, we must guide the tool-path generator to engrave portions of volume, ensuring accessibility and avoiding milling surfaces that belong to other charts. We must thus embed each chart surface in the stock, computing the volume that has to be carved to mill it (Sec. 3.5.1), order the charts into a fabrication sequence (Sec. 3.5.2), and finally construct the final fabrication plan (Sec. 3.5.3), By solving these problems, we

make it possible to use our pipeline with any of the available 3-axis software.

3.5.1 Embedding chart surfaces into the stock

To be fabricated, each shape to be milled must embed its chart into the given stock, meeting the following constraints:

- it contains the entire surface of the chart to mill;
- it does not contain surface assigned to other charts, or, when not possible, it is minimum;
- it can be milled entirely without collisions of the drill bit.

Note that, in our examples, we used a cylinder as stock to maximize the final result size and minimize scrap material, but any other shape is allowed.
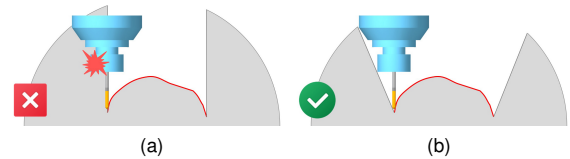

Figure 10: *2D representation of how we generate the surface (called walls) for embedding the chart surface into the stock. (a) Limitations of a surface that is parallel with respect with the fabrication direction. (b) Surface that form a given angle guarantees fabricability and manufacturing quality.*

For each chart, we have to propagate the surface from its boundaries to the fabrication volume's extremes. This propagated surface must guarantee both physical access of the milling tool and high precision near the boundaries. From now on, we define this surface propagation as *walls*. As illustrated in Fig. 10a, the trivial solution could be to project the extrema of each chart in the direction orthogonal to the rotation axis (we would obtain parallel walls) to get the portion of stock to mill. This choice would lead to collisions and the impossibility to mill near the chart boundaries. Instead, we tilt the walls until they form an angle with the fabrication direction that avoids collisions when milling the borders of the chart (Fig. 10b). This solution allows for the milling tool's physical access but leads to other issues that must be considered. We will elaborate on these issues in the next sections.

3.5.2 Ordering the charts

Each of the resulting meshes generated by the embedding procedure contains the respective chart's surface and oblique walls. However, as shown in Fig. 11a, oblique walls might collide with other charts. In this case, since each chart with its walls is engraved in a dedicated milling pass, we can improperly remove different portions of volume. To avoid unwanted volume removal, we can incorporate another chart's surface in each milling pass with an exact Boolean operation. However, when this solution is adopted, we mill that surface portion from a direction different from the one assigned. As illustrated in Fig. 11, the amount of surface milled from sub-optimal directions is affected by chart order. In the example shown, if C_2 is milled before C_1, part of the surface of C_1 will be milled from the direction selected for C_1. Conversely, if C_1 is milled before C_2, the problem does not appear, as the hypothetical wall generated for C_2 would fall outside the remaining volume that must
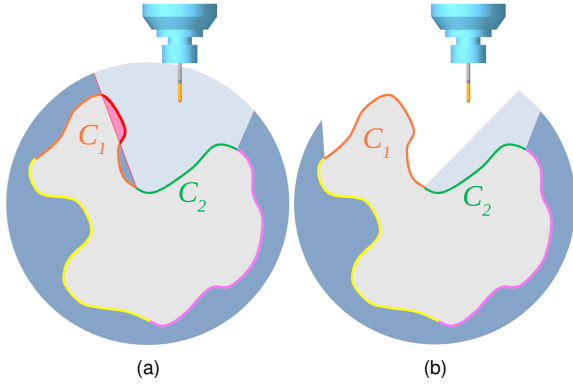
Figure 11: *Collision between walls and other charts. (a) The oblique walls of the chart C_2 collide with a portion of the surface of another chart C_1 (highlighted in red). In this case, the red area would be unintentionally removed when performing the milling of C_2 and, therefore, fabricated from a wrong direction. (b) Milling C_1 before C_2 solves the issue, because the remaining volume to be removed (lighter blue) does not contain the actual wall of C_2.*

be removed to mill C_2. Therefore, we introduce in our pipeline a sorting of the charts to reduce the conflicts.

To derive the best ordering, given the n charts $C_1, ..., C_n$ forming the decomposition of the side of our input shape, we define a $n \times n$ conflict matrix $\mathcal{M}$, whose entries $\mathcal{M}(c, d)$ (we consider only $c \neq d$) are defined as the total area of the surface belonging to a chart d that would be disrupted while milling the chart c:

$$\mathcal{M}(c,d) = \sum_f a_f \quad \text{s.t. } f \text{ is reachable and } f \in d \qquad (6)$$

where a_f denotes the face area in faces f of the embedded shape. In other words, $\mathcal{M}(c, d)$ represents the cost, in terms of surface area, of fabricating a chart c before a chart d. We want to find a chart ordering $(c_1, ..., c_n)$ that minimizes the overall cost:

$$\underset{(c_1,...,c_n)}{\operatorname{argmin}} \sum_{i=1}^{n} \sum_{j=i+1}^{n} \mathcal{M}(c_i, c_j) \qquad (7)$$

We solve this optimization problem with a greedy approach. Conventionally, let $\mathcal{M}(c, d) = 0$ for $c = d$. At each iteration i, let $\mathcal{C} = \{C_1, ..., C_n\} \setminus \{c_1, ..., c_{i-1}\}$ be the set of charts that have been not processed yet. Then, we select the chart c_i as follows:

$$c_i = \underset{c \in \mathcal{C}}{\operatorname{argmin}} \sum_{d \in \mathcal{C}} \mathcal{M}(c, d) \qquad (8)$$

The cost term in the summation represents the chart's surface c_i having conflicts during the current iteration. In case of equal costs, we choose the chart c_i that maximizes $\sum_{d \in \mathcal{C}} \mathcal{M}(d, c)$, that is the area that will not conflict anymore with other charts in the next iterations. Although this heuristic does not guarantee finding the optimal solution, it works very well in practice. In all our experiments, we were always able to find a result with zero overall cost, except when we had cross-conflicts between charts. The part of the object belonging to charts not yet milled is usually entirely hidden by our embedding method.

3.5.3 Generating the final fabrication plan

Once we derive the best order, we can create the final fabrication plan. The 3-axis driving software typically requires two input shapes per chart: one representing the initial block of raw material to be carved (the stock) and the desired target shape. Hence, by determining these meshes, we can readily produce our shapes with any widely available 3-axis driver, entirely abstracting from the usage of a 4-axis machine.

Starting from the initial block and the first chart, we iteratively process each target shape. In case it was not possible to avoid all the collisions between charts and walls, the concerned chart surface is included in the target shape. In this way, we choose to fabricate it with less accuracy from another direction, instead of losing a portion of the final object. At each iteration, we use the resulting mesh of the previous step as the initial stock. We use robust Boolean operations described by Zhou et al. [ZGZJ16] to perform these operations. Fig. 12 shows a simple 2D representation of the process.

Our results leave intentionally small portions of waste material that can be removed manually after the fabrication. These small portions lie between adjacent charts, and the precision of the milling machine determines their thickness. To save fabrication time, we remove these portions manually at the end of the milling process. Note that we could avoid them by increasing the angle parameter used to generate the chart walls.
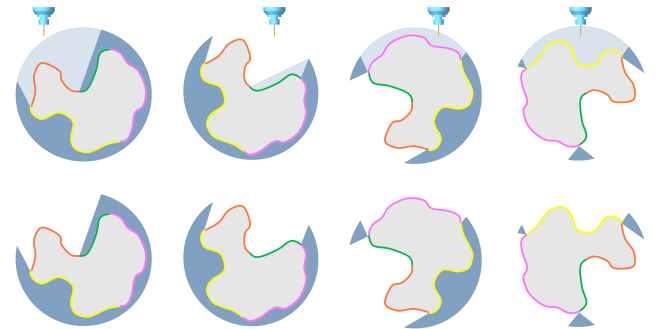

Figure 12: *Set of meshes generated by our method. On top, the stocks (with the part that will be milled in a lighter color), and on bottom the resulting shape. At each step, the resulting shape will be rotated and used as stock for the next step. The ordering of the charts is computed in order to minimize the area fabricated from a wrong direction.*

Finally, we generate the stock and target meshes for the top and bottom regions. The process is similar, but we remove the volume around to easily remove the waste material at the end of the manufacturing process. Indeed, the remaining stock portions in the final target shape can be easily detached at the end of the fabrication process, saving manufacturing time. The user can choose to process first, either the top or bottom region. We generate a mold for the first one to secure and align the model to the machine during the fabrication of the second region. Also, for the top, the bottom, and the mold, all the meshes required for the manufacturing are computed using robust Boolean operations. Fig. 13 shows all the meshes generated for the Batman model.
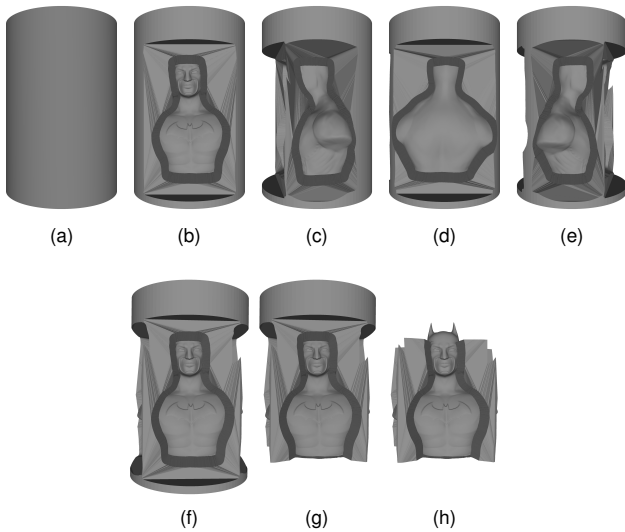
Figure 13: *Fabrication sequence. The first row shows the initial stock (a) and target shapes for side region (b, c, d, e). Each target shape is rotated properly and used as stock for the next step. The second row shows the resulting mesh after 4-axis fabrication and the target shapes for the bottom (g) and top (h) regions.*

4 Implementation and results

Our approach has been implemented in C++, using cg3lib [MN21] and VCG [CNR13] for geometry processing functions, Eigen [GJ*14] for linear algebra routines, libigl [JP*16] and CGAL [FP09] for mesh Booleans, PyMeshLab [MC21] for mesh measurements. We plan in a future release to use the method described in [CLSA20] as soon as it is available in open source for Boolean operations since it promises to be more efficient. To ensure our approach's replicability, we will provide the source code of the application released with an open-source license. In this paper, we present results obtained on a broad range of organic models.

4.1 Computational and machining setup

All the fabrication plans have been computed on a workstation with a 6-cores Intel i5-8600K processor clocked at 3.6 GHz and 16 GB of memory.

Our method is of general use and can be applied to any 4-axis CNC machine. All the models presented here were fabricated using a CNC machine Stepcraft 2 840, with a working area (X, Y, Z) of $600 \times 840 \times 140$ mm with an added 4-axis module and an HF spindle 500 W by Stepcraft. We employed a 3 mm flat cutter with two flutes for the roughing phase, while for the finishing stage, we used a 1 mm ball cutter with two flutes. The machine imposes limits on the manufacturing size of the model. Our models must fit inside a cylinder of 70mm in height and diameter. To drive the machine, we use Fusion 360 by Autodesk for the tool-path generation.

4.2 Performance and quality evaluation

Our method has been evaluated on a variety of shapes (see Tab. 1). Fig. 14 shows the four results that have been manufactured and Fig. 15 presents several results of the segmentation process. The proposed pipeline can successfully fabricate objects with genus zero

or higher. In all our experiments, we used compactness term $c = 30$ and saliency factor $d = 25$.

Tab. 1 provides statistics about our experiments. For each model, we report the processing time. We split the timing into four intervals: (i) computation of the saliency map to avoid placing seams on semantically relevant portions of the shape; (ii) identification of the rotation axis; (iii) segmentation of the shape to associate each triangle to a single map linked to a height-field; (iv) partitioning the 4-axis process in a sequence of 3-axis milling. All times are in seconds. As it is possible to observe, segmentation is the task requiring the highest computational effort. In the last column, we list the number of resulting charts. As we can see, the number of charts is maintained very low despite the shape complexity.

In our results, we applied the prefiltering described in Sec. 3.1. To test their fidelity, we computed the distance between the input surfaces and the results after the reintroduction of high frequencies, using Metro [CRS98]. The resulting distance is less than 1×10^{-2} on average, with 2×10^{-5} variance and a maximum value of 1.7×10^{-2}. All these values are relative to the bounding box diagonal of the input models. We can thus conclude that, for the tested models, our fabrication plan produces meshes that are very close to the original mesh. It should also be noted that this accuracy is higher than what achievable in practice by our milling machine (see Sec. 4.1).

We also computed the total surface area that would be manufactured from a direction that is different from the one planned in our segmentations due to the chart ordering problem described in Sec. 3.5.2. This surface is on average 0.7% w.r.t. the input model's total surface and 11 models have 0% of this surface, showing the effectiveness of the greedy sorting method. In any case, manufacturing a chart from a different milling direction from the planned one only reduces quality and does not cause other significant issues on the final result. For example, we successfully manufactured the kitten model that had 2.9% of surface area manufactured from a different direction.

In Tab. 2, we report some of the physical measures regarding the fabrication process of the objects we manufactured.

4.3 Illustration of the fabrication process

To illustrate our pipeline's practical usage in a CNC machining workshop, we describe a model's complete fabrication process.

Fig. 2 shows the initial setup. We used PVC cylinders as stock for the kitten and the Buddha models and wood for the batman and David models. The cylinder is mounted in the rotation axis between the step motor and the tail-stock.

We start milling the side regions of the model, first performing the roughing step using the 3mm flat cutter for each direction and then performing a finishing step with a 1mm ball cutter. We show in Fig. 16 the side regions completed of the kitten model.

We then remove the block and place it on the plane, and we perform the fabrication of the top, the mold, and the bottom of the shape. Each of those three fabrication phases is a typical 3-axis milling pass requiring that the milled material be fixed and aligned to the machine plane. We first manufacture the top by fixing the
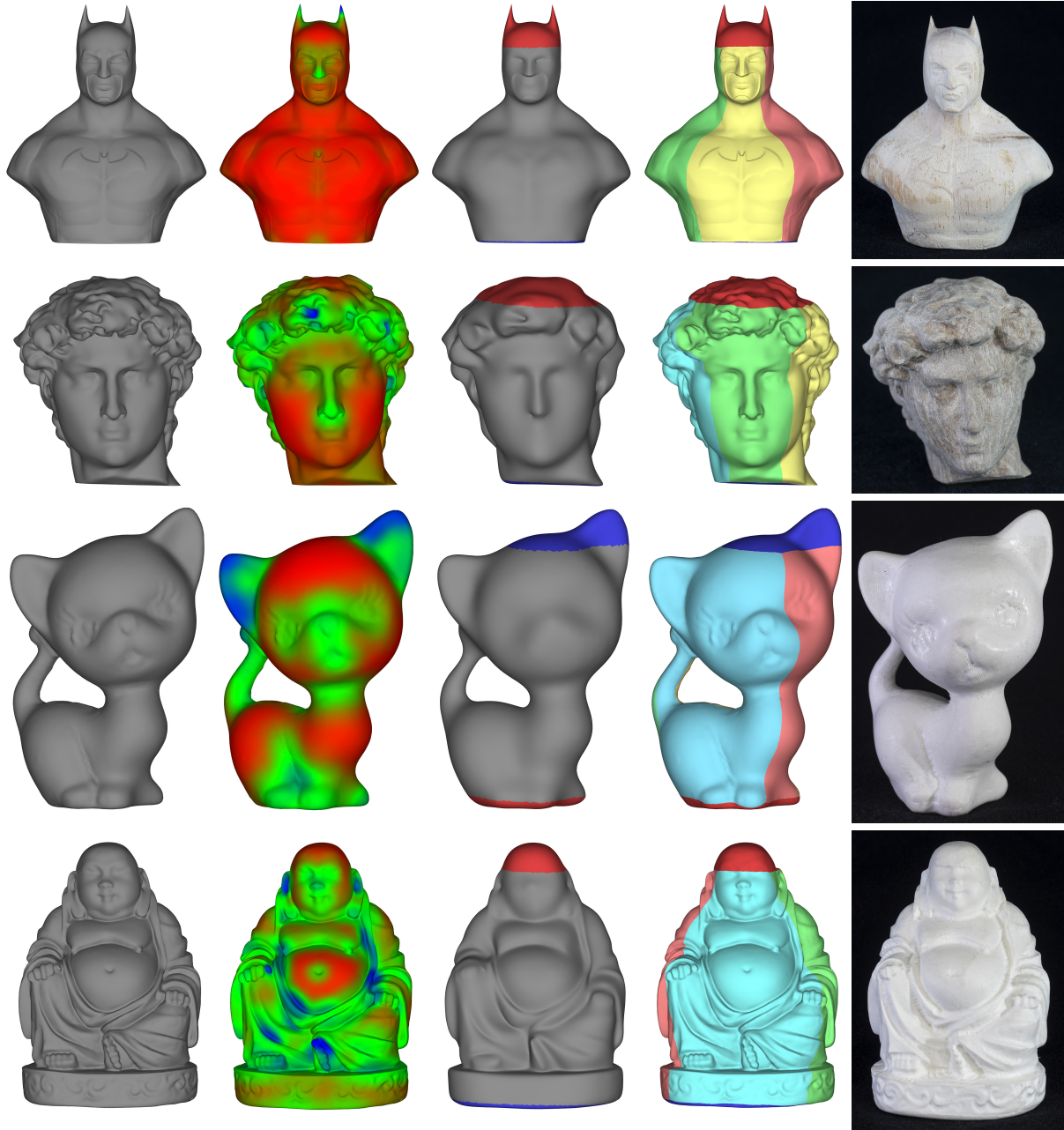
Figure 14: *Fabricated models. For the four fabricated models we show here (left to right): the initial shape; its saliency map; the two ends (in red and blue) displayed on the smoothed model; the decomposition of the model in height-fields; the fabricated object (in PVC for the kitten and the Buddha models, and wood for the batman and David models).*

stock on its original flat base (Fig. 17a). Then we use the mold to fix the top region, and we fabricate the bottom one (Fig. 17c). The mold (Fig. 17b) needs to be milled separately. Each step is composed of a roughing and a finishing phase. Fig. 17 shows the three fabrication steps and how each block is secured to the machine plane.

It's worth noticing that the model's small size (a few cm for the kitten) influences its quality since the cutter cannot be tinier than 1mm. The ratio between the cutter size and the shape size is an essential parameter of the fabrication step. In Fig. 18a, we show that,

at the end of the milling, a portion of the stock initially surrounding the shape may remain attached to it due to limitations in machine accuracy. The connection is very thin, and we can easily remove it (Fig. 18b). To bring the model to a smooth shape in those cases, we can use a rasp or sandpaper to remove the material along the seams (Fig. 18c). Despite this disadvantage, we can obtain excellent results.
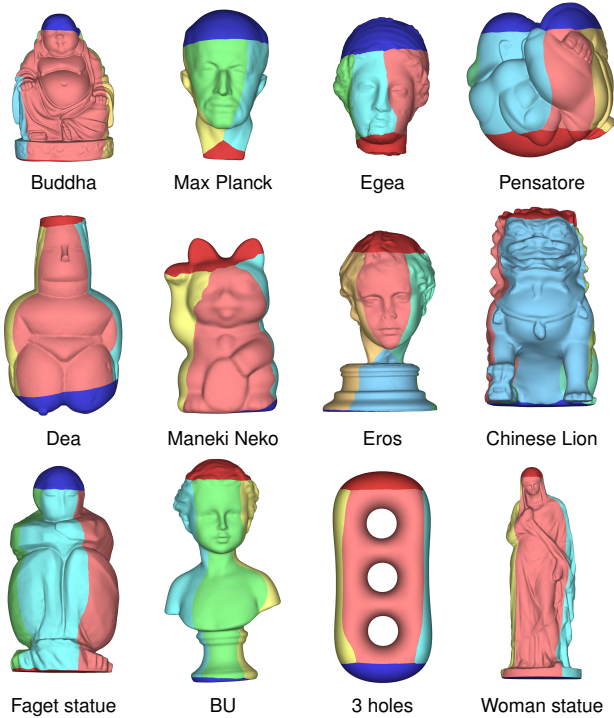
Figure 15: *Several final segmentations. Final segmentations obtained with our method.*

4.4 Limitations

Our pipeline enables us to manufacture 3D free-form shapes from a single solid block using 4-axis milling machines, but the whole pipeline bases upon heuristics. Every step of the pipeline aims to obtain a practical and reliable result and not explore the entire solution space searching for optimality. The decomposition of the whole problem in simpler sub-problems allows us to follow this approach with appraisable results. The theoretical question of

Table 1: *Model Statistics. The first column shows the number of faces (NF), the next four columns (S, saliency; O optimal orientation; A face association; M fabrication sequence and mesh generation) list the timing of the computation, in the last column we report the number of obtained charts (NC). Times are in seconds.*

Model	NF	S	O	A	M	NC
3 holes	43K	0.7	5.8	23.7	12.5	6
Batman	120K	11.4	34.6	279.2	31.3	6
BU	72K	4.8	19.1	118.7	21.9	6
Buddha	120K	8.5	36.8	151.7	30.0	6
Chinese Lion	60K	2.8	15.3	67.3	33.5	8
David	60K	2.6	14.7	72.7	19.9	6
Dea	40K	4.5	21.4	206.5	22.6	6
Egea	30K	1.3	7.3	22.7	12.0	6
Eros	120K	9.2	35.6	201.5	41.8	8
Faget statue	60K	2.5	15.2	85.9	16.7	6
Kitten	37K	1.4	8.4	66.6	17.0	7
Max Planck	54K	2.2	13.8	80.6	17.2	6
Moai	40K	1.2	9.0	60.0	13.7	6
Maneki neko	44K	1.6	10.0	28.4	14.8	6
Pensatore	60K	2.6	15.5	59.0	23.1	7
Woman statue	60K	3.6	13.8	73.4	16.6	6

Table 2: *Model dimensions. We report, for the four fabricated models, the size in millimeters of the target model and the initial stock.*

Model	Model height	Stock diameter	Stock length
Batman	70	72	92
David	54	62	86
Buddah	70	72	86
Kitten	70	72	88

Figure 16: *Finishing phase. Side regions finished with a 1mm ball cutter.*

finding the best solution remains an open issue, and our approach leads to some limitations, both theoretical and practical.

From a theoretical point of view to the best of our knowledge, there is no formal definition of shapes manufacturable in a single block using 4-axis milling machines. This classification is a non-trivial problem due to many constraints to consider to generate a working plan. It is trivial that shapes with large portions of the surface not visible (due to occlusion) from any direction unquestionably cannot be manufactured in a single block with classic 4-axis subtractive techniques. In Fig. 19 we show a failure example. When dealing with these kinds of shapes, our approach fails to find a proper segmentation covering the entire surface.

Our approach produces results that are highly influenced by choice of the rotation axis. Since the proposed method relies on a heuristic and occlusions are not fully taken into account, we cannot guarantee the result's optimality. Anyway, as long as the shapes do not have vast portions of the surface occluded from every direction, our method produces segmentations that might be composed of a high number of charts but theoretically viable.

In Fig. 19 we show the segmentation of a shape characterized by the presence of several protrusions pointing in different directions. Due to occlusions, the segmentation presents a large number of small non-visible charts. We could solve this by removing the charts, as explained in Sec. 3.3.3, but it would lead to less fidelity in the fabrication.

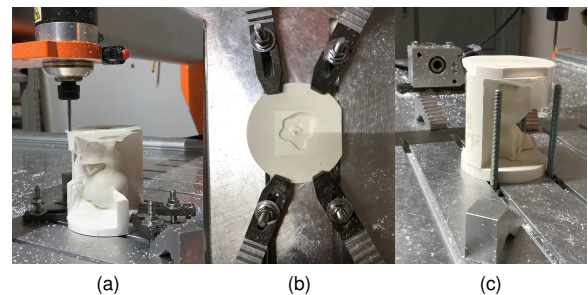Theoretically valid segmentations may not be practically feasible.

Figure 17: *Fabrication phases.. Manufacturing the first of the two ends of the model (a), then the second end of the model (c) using its mold (b).*
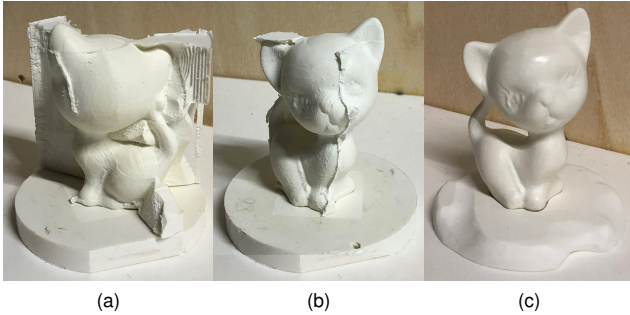
Figure 18: *Fabricated model before and after the final sanding pass. In (a) the model has still part of the stock that, initially, was surrounding it; in (b) only the portion that is to be sanded is present. in (c), the final polished model, using the mold as a base*
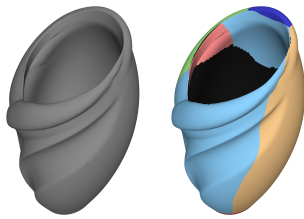

Figure 19: *Failure case due to intrinsic limitations of 4-axis milling. In the depicted cup-shaped surfaces, faces in black are non-visible from any possible direction.*

The shape shown in Fig. 20 does not have two substantial portions of the surface that we can pick as opposite extremes. Our approach finds two small regions (the bottom of one leg, in red, and the trunk tip, in blue) that, depending on the material used for the manufacturing and the final object's size, could easily lead to instability fabrication. Side regions must be wide enough to sustain the entire object weight during the whole manufacturing process. We could address this problem by just adding a term in the optimization problem described in Sec. 3.2.2 that penalizes the formation of small top and bottom areas. Alternatively, we can discard the axes that generate top and bottom regions smaller than a certain threshold. However, we cannot guarantee to have a suitable solution for the general case. In general, for this kind of shape, single-block fabrication on 4-axis machines is challenging, if not impossible, and it would be ideal for splitting the model into multiple components that can be fabricated separately and assembled afterward.
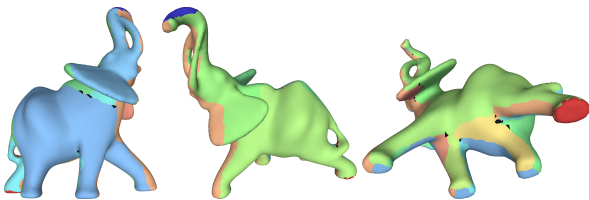

Figure 20: *Practical limitations. The fabrication of this shape has two problems: many small non-reachable charts and two little opposite extremes. We could solve the first by sacrificing fidelity on the non-visible surface. Still, the second issue could lead to instability during the object's manufacturing.*

5 Conclusions and discussion

We introduced a novel automatic method for the fabrication of non-regular shapes using 4-axis subtractive machinery. Our results demonstrate that our approach allows for the fabrication of highly detailed models, even with a genus different from zero. It permits us to make robust single-block results devoid of visible seams, in contrast to previous methods based on the assembly of separately produced parts. Taking into account saliency information improves the fabricated result's quality by guaranteeing the fabrication of the most semantically relevant areas of the model in a single step.

We designed our method so that it can be a step in existing pipelines. It can fit within current industrial fabrication settings with minimal upgrades. The user can control the accuracy and fidelity of the manufactured model through user-controllable detail filtering thresholds. We give the possibility to cut small details when necessary, to obtain effective fabrication plans.

We heavily rely upon the 3-axis milling processing for every single portion of the fabrication. Exploiting the presence of an additional rotation axis, we add value in avoiding the decomposition and reassembling of the shape. Following this strategy, users with a 3-axis milling machine, adding a rotation shaft, can use our method. We thus expect that our technique can have immediate practical applications, possibly at an industrial level in small and medium enterprises. It significantly extends the variety of shapes fabricable on standard low-cost machines.

As widely discussed in the previous section, not every shape is fabricable using our approach, avoiding further assembling. More generally, not every shape is fabricable, even using 5- or 6-axis professional machines. Our proposal has the advantage that the upgrade from a 3-axis to a 4-axis machine is dramatically more effortless than 3-axis to 5-axis. If needed, it is possible to decompose the input shape into multiple cylindrical components that can be milled independently and assembled afterward. This approach would result in fewer parts than the height-field decomposition methods.

References

[ACA*19] ARAÚJO, CHRYSTIANO, CABIDDU, DANIELA, ATTENE, MARCO, et al. "Surface2Volume: Surface Segmentation Conforming Assemblable Volumetric Partition". *ACM Trans. Graph.* 38.4 (2019), 80:1–80:16. DOI: https://doi.org/10.1145/3306346.3323004 3.

[ACP*14] ALEMANNO, GIUSEPPE, CIGNONI, PAOLO, PIETRONI, NICO, et al. "Interlocking Pieces for Printing Tangible Cultural Heritage Replicas". *Eurographics Workshop on Graphics and Cultural Heritage*. Eurographics Association, 2014, 145–154. DOI: 10.2312/gch.20141312 2.

[AMG*18] ALDERIGHI, THOMAS, MALOMO, LUIGI, GIORGI, DANIELA, et al. "Metamolds: computational design of silicone molds". *ACM Trans. Graph.* 37.4 (2018), 136:1–136:13. DOI: 10.1145/3197517.3201381 2.

[AMG*19] ALDERIGHI, THOMAS, MALOMO, LUIGI, GIORGI, DANIELA, et al. "Volume-aware design of composite molds". *ACM Trans. Graph.* 38.4 (2019), 110:1–110:12. DOI: 10.1145/3306346.3322981 3.

[BVZ01] BOYKOV, YURI, VEKSLER, OLGA, and ZABIH, RAMIN. "Fast approximate energy minimization via graph cuts". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23.11 (2001), 1222–1239. DOI: 10.1109/34.969114 5.

[CLSA20] CHERCHI, GIANMARCO, LIVESU, MARCO, SCATENI, RICCARDO, and ATTENE, MARCO. "Fast and Robust Mesh Arrangements Using Floating-Point Arithmetic". *ACM Trans. Graph.* 39.6 (2020), 250:1–250:16. DOI: 10.1145/3414685.3417818 9.

[CNR13] CNR. *The Visualization and Computer Graphics Library*. 2013. URL: http://vcg.isti.cnr.it/vcglib/ 9.

[CRS98] CIGNONI, PAOLO, ROCCHINI, CLAUDIO, and SCOPIGNO, ROBERTO. "Metro: measuring error on simplified surfaces". *Computer Graphics Forum*. Vol. 17. 2. 1998, 167–174. DOI: 10.1111/1467-8659.00236 9.

[CZL*15] CHEN, XUELIN, ZHANG, HAO, LIN, JINJIE, et al. "Dapper: decompose-and-pack for 3D printing". *ACM Trans. Graph.* 34.6 (2015), 213:1–213:12. DOI: 10.1145/2816795.2818087 2.

[FAG*20] FILOSCIA, IRENE, ALDERIGHI, THOMAS, GIORGI, DANIELA, et al. "Optimizing Object Decomposition to Reduce Visual Artifacts in 3D Printing". *Computer Graphics Forum* 39.2 (2020), 423–434. DOI: 10.1111/cgf.13941 2.

[FCM*18] FANNI, FILIPPO A., CHERCHI, GIANMARCO, MUNTONI, ALESSANDRO, et al. "Fabrication oriented shape decomposition using polycube mapping". *Computers & Graphics* 77 (2018), 183–193. DOI: 10.1016/j.cag.2018.10.010 3.

[FP09] FABRI, ANDREAS and PION, SYLVAIN. "CGAL: The computational geometry algorithms library". *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 2009, 538–539. DOI: 10.1145/1653771.1653865 9.

[FWJ06] FRANK, MATTHEW C, WYSK, RICHARD A, and JOSHI, SANJAY B. "Determining Setup Orientations From the Visibility of Slice Geometry for Rapid Computer Numerically Controlled Machining". *ASME. J. Manuf. Sci. Eng.* 128.1 (2006), 228–238. DOI: 10.1115/1.2039100 3.

[GJ*14] GUENNEBAUD, GAEL, JACOB, BENOIT, et al. *Eigen: a c++ linear algebra library*. 2014. URL: http://eigen.tuxfamily.org 9.

[HLZC14] HU, RUIZHEN, LI, HONGHUA, ZHANG, HAO, and COHEN-OR, DANIEL. "Approximate Pyramidal Shape Decomposition". *ACM Trans. Graph.* 33.6 (2014), 213:1–213:12. DOI: 10.1145/2661229.2661244 2.

[HMA15] HERHOLZ, PHILIPP, MATUSIK, WOJCIECH, and ALEXA, MARC. "Approximating Free-form Geometry with Height Fields for Manufacturing". *Computer Graphics Forum* 34.2 (2015), 239–251. DOI: 10.1111/cgf.12556 2.

[JP*16] JACOBSON, ALEC, PANOZZO, DANIELE, et al. *libigl: A simple C++ geometry processing library*. 2016. URL: https://libigl.github.io 9.

[LBRM12] LUO, LINJIE, BARAN, ILYA, RUSINKIEWICZ, SZYMON, and MATUSIK, WOJCIECH. "Chopper: partitioning models into 3D-printable parts". *ACM Trans. Graph.* 31.6 (2012), 129:1–129:9 2.

[LVJ05] LEE, CHANG HA, VARSHNEY, AMITABH, and JACOBS, DAVID W. "Mesh Saliency". *ACM Trans. Graph.* 24.3 (2005), 659–666. DOI: 10.1145/1073204.1073244 6.

[MC21] MUNTONI, ALESSANDRO and CIGNONI, PAOLO. *PyMeshLab*. Jan. 2021. DOI: 10.5281/zenodo.4438750 9.

[MLS*18] MUNTONI, ALESSANDRO, LIVESU, MARCO, SCATENI, RICCARDO, et al. "Axis-Aligned Height-Field Block Decomposition of 3D Shapes". *ACM Trans. Graph.* 37.5 (2018), 169:1–169:15. DOI: 10.1145/3204458 3, 4, 7.

[MN21] MUNTONI, ALESSANDRO and NUVOLI, STEFANO. *CG3Lib: A C++ geometry processing library*. Jan. 2021. DOI: 10.5281/zenodo.4431777 9.

[MPBC16] MALOMO, LUIGI, PIETRONI, NICO, BICKEL, BERND, and CIGNONI, PAOLO. "FlexMolds: Automatic Design of Flexible Shells for Molding". *ACM Trans. Graph.* 35.6 (2016), 223:1–223:12. DOI: 10.1145/2980179.2982397 2.

[MYZ*20] MAHDAVI-AMIRI, ALI, YU, FENGGEN, ZHAO, HAISEN, et al. "VDAC: Volume Decompose-and-Carve for Subtractive Manufacturing". *ACM Trans. Graph.* 39.6 (2020), 203:1–203:15. DOI: 10.1145/3414685.3417772 3.

[NKI*18] NGO, TUAN D., KASHANI, ALIREZA, IMBALZANO, GABRIELE, et al. "Additive manufacturing (3D printing): A review of materials, methods, applications and challenges". *Composites Part B: Engineering* 143 (2018), 172–196. DOI: 10.1016/j.compositesb.2018.02.012 1, 2.

[SEPC16] SÁ, ASLA MEDEIROS E, ECHAVARRIA, KARINA RODRIGUEZ, PIETRONI, NICO, and CIGNONI, PAOLO. "State of The Art on Functional Fabrication". *Eurographics Workshop on Graphics for Digital Fabrication*. 2016. DOI: 10.2312/gdf.20161073 2.

[SJ06] SWINBANK, RICHARD and JAMES PURSER, R. "Fibonacci grids: A novel approach to global modelling". *Quarterly Journal of the Royal Meteorological Society* 132.619 (2006), 1769–1793. DOI: 10.1256/qj.05.227 5.

[SJHC01] SUH, S.H., JIH, W.S., HONG, H.D., and CHUNG, D.H. "Sculptured surface machining of spiral bevel gears with CNC milling". *International Journal of Machine Tools and Manufacture* 41.6 (2001), 833–850. DOI: 10.1016/S0890-6955(00)00104-8 2.

[Sor06] SORKINE, OLGA. "Differential Representations for Mesh Processing". *Computer Graphics Forum* 25.4 (2006), 789–807. DOI: 10.1111/j.1467-8659.2006.00999.x 7.

[Tan14] TANG, TRAN DUC. "Algorithms for collision detection and avoidance for five-axis NC machining: A state of the art review". *Computer-Aided Design* 51 (2014), 1–17. DOI: 10.1016/j.cad.2014.02.001 2.

[Tau95] TAUBIN, G. "Curve and surface smoothing without shrinkage". *Proceedings of IEEE International Conference on Computer Vision*. 1995, 852–857. DOI: 10.1109/ICCV.1995.466848 4.

[YAV*20] YANG, JINFAN, ARAUJO, CHRYSTIANO, VINING, NICHOLAS, et al. "DHFSlicer: Double Height-Field Slicing for Milling Fixed-Height Materials". *ACM Trans. Graph.* 39.6 (2020), 205:1–205:17. DOI: 10.1145/3414685.3417810 3.

[ZGZJ16] ZHOU, QINGNAN, GRINSPUN, EITAN, ZORIN, DENIS, and JACOBSON, ALEC. "Mesh Arrangements for Solid Geometry". *ACM Trans. Graph.* 35.4 (2016), 39:1–39:15. DOI: 10.1145/2897824.2925901 8.

[ZZX*18] ZHAO, HAISEN, ZHANG, HAO (RICHARD), XIN, SHIQING, et al. "DSCarver: Decompose-and-Spiral-Carve for Subtractive Manufacturing". *ACM Trans. Graph* 37.4 (2018), 137:1–137:14. DOI: 10.1145/3197517.3201338 3, 5.