Volume decomposition for two-piece rigid casting

THOMAS ALDERIGHI, University of Pisa and ISTI-CNR, Italy
LUIGI MALOMO, ISTI-CNR, Italy
BERND BICKEL, IST, Austria
PAOLO CIGNONI, ISTI-CNR, Italy
NICO PIETRONI, University of Technology Sydney, Australia

Fig. 1. Given a closed 2-manifold mesh, we split it automatically into double height field (DHF) components that can be fabricated using two-piece rigid casting and assembled afterwards.

We introduce a novel technique to automatically decompose an input object's volume into a set of parts that can be represented by two opposite height fields. Such decomposition enables the manufacturing of individual parts using two-piece reusable rigid molds. Our decomposition strategy relies on a new energy formulation that utilizes a pre-computed signal on the mesh volume representing the accessibility for a predefined set of extraction directions. Thanks to this novel formulation, our method allows for efficient optimization of a fabrication-aware partitioning of volumes in a completely automatic way. We demonstrate the efficacy of our approach by generating valid volume partitionings for a wide range of complex objects and physically reproducing several of them.

CCS Concepts: • **Computing methodologies** → **Mesh geometry models**; *Shape analysis.*

Additional Key Words and Phrases: double height field, volumetric decomposition, casting

ACM Reference Format:
Thomas Alderighi, Luigi Malomo, Bernd Bickel, Paolo Cignoni, and Nico Pietroni. 2021. Volume decomposition for two-piece rigid casting. *ACM Trans. Graph.* 40, 6, Article 272 (December 2021), 14 pages. https://doi.org/10.1145/3478513.3480555

Authors' addresses: Thomas Alderighi, University of Pisa and ISTI-CNR, Italy, thomasalderighi@gmail.com; Luigi Malomo, ISTI-CNR, Italy, luigi.malomo@isti.cnr.it; Bernd Bickel, IST, Austria, bernd.bickel@ist.ac.at; Paolo Cignoni, ISTI-CNR, Italy, paolo.cignoni@isti.cnr.it; Nico Pietroni, University of Technology Sydney, Australia, nico.pietroni@uts.edu.au.

0730-0301/2021/12-ART272 $15.00
https://doi.org/10.1145/3478513.3480555

1 INTRODUCTION

Due to its efficiency, versatility, and scalability in cost and production speed, molding is a de facto standard for mass production. Numerous common products, ranging from plastic parts to edible items, are brought into shape using molds. Molds are usually composed of multiple shell components surrounding a cavity representing the volume of the desired object. By filling the cavity with a liquid that solidifies, they can be used to materialize the target shape. In this paper, we focus on reusable rigid molds that can be employed in numerous fabrication processes, such as casting, injection molding, or blow molding. For reusable rigid molds, the complexity of the to-be-fabricated object's shape is directly related to the complexity of the mold itself, such as the number of required mold pieces and their geometry. When considering only rigid, linear mold extraction processes, a mold piece can be removed only if the contact surface with the object corresponds to a height field and there are no collisions with other parts. While articulated molds composed by multiple pieces allow the reproduction of sophisticated shapes, *two-piece molds* are, in contrast, significantly simpler to design and operate. However, their applicability is restricted to shapes that can be represented as double height fields (DHFs).

In practice, to benefit from the advantages of two-piece molding, shapes are often explicitly designed to fulfill the double height field constraint, or their shape is modified [Stein et al. 2019] to ensure castability as a single piece. An alternative approach for extending the method to more general shapes and preserving their original geometry is to split the object into multiple parts that individually fulfill the moldability criteria and then assemble them afterward. However, this approach comes with a demanding design problem: how can we segment the volume into a *small number of parts* while ensuring that each part can be represented as a double height field? As parts also have to be assemblable, the surfaces of two parts that

are in contact with each other need to match. These constraints restrict potential double height field configurations between neighboring parts and create global dependencies, making the problem challenging to solve.

In this paper, we address this problem by proposing a novel optimization-based approach to decompose a volumetric object into multiple parts, guaranteeing that each part is manufacturable with a two-piece mold. Unlike traditional shape segmentation approaches, which rely on surface representations, our system exploits a volumetric representation of the object. We show an overview of the pipeline in Figure 2. Given a tetrahedralization of the object to be cast, we first compute a casting feasibility value (*castability*) for each volumetric element, for a set of predefined directions. Then, exploiting these values, we derive an optimized decomposition of the volume by solving the partitioning problem using graph-cut. We employ a novel energy formulation that produces a decomposition into double height field pieces that are compact and easy to fabricate and assemble. The contact surfaces between parts are generated starting from the volume partitioning with a new approach that ensures smooth contact surfaces while preserving castability and assemblability.

Our fully automatic approach offers advantages with respect to intelligent, assisted tools that help the user to create sound decompositions like the one proposed by Nakashima et al. [2018]. While manual-assisted methods can lead to visually appealing shape decompositions, this process still requires a substantial effort from the user. The problem might quickly become hard to manage for significantly complex geometries with a high genus or high-frequency details. Furthermore, Nakashima et al. [2018], when decomposing, do not offer a complete, explicit treatment of contact surfaces between pieces and, even if they work on hollow thin-shell objects where this problem is less critical, they cannot guarantee that parts match perfectly.

We demonstrate the reliability of the method by showing several fabricated examples. As illustrated in the results section, our method can automatically decompose complex shapes and thereby creates a new avenue towards more efficient and automatic reproduction of digital artifacts with rigid two-piece molding.

2 RELATED WORK

There is a vast literature on rigid mold generation in the computer-aided design and mechanical engineering fields [Chakraborty and Reddy 2009; Lin and Quang 2014; Zhang et al. 2010]. Most of these methods focus on how to obtain a proper mold or generate an object decomposition that targets mechanical objects. More generally, the task at hand is to custom design molding mechanisms to enable rigid casting for very specific examples or to automatize the casting process for geometries that are already designed for manufacture by this process. For this reason, these approaches do not work in the general case (e.g., generating the mold for casting organic free-form shapes). The task of designing a casting process with rigid molds is non-trivial, as it involves multiple necessary conditions in the form of simultaneous geometric constraints, which must be rigorously satisfied for feasibility. For this reason, within industrial applications this task is manual and allocated to skilled experts.

In recent years, many methods have been proposed to partition a shape into multiple pieces in order to simplify its fabrication process. A recent survey [Wang et al. 2021] provides an overview of the techniques for designing and assembling rigid assemblies. A class of these techniques uses mesh partitioning to overcome the limitations and the lack of additive manufacturing. Another substantial part of the research focuses on shape decomposition for subtractive manufacturing (i.e., CNC milling fabrication); other methods are explicitly dedicated to the molding process. While these three scenarios are substantially different from a practical perspective, they share similarities regarding the geometric constraints involved. Most of these works, including those by Araújo et al. [2019], Herholz et al. [2015], and Muntoni et al. [2018], address the decomposition as a multi-label segmentation problem defined over either the object surface or its volume, and focus on tailoring the energy formulation and constraints to the problem at hand. Such formulations are commonly solved using known optimization techniques like integer linear programming or graph-cut optimization. In Sections 3 and 4 we will describe the constraints related to our scenario and how we model them within the graph-cut optimization framework.

2.1 Segmentation for Additive Manufacturing

In the context of additive manufacturing, a fabricated object might also be decomposed to accommodate a different class of practical problems. Chopper [Luo et al. 2012], for example, decomposes the mesh into parts that individually fit with the 3D printer's workspace. Other decomposition methods include the minimization of supports [Hu et al. 2014], the optimization of packing [Chen et al. 2015], or the reduction of the visible artifacts of 3D printed supports in the final assembly [Filoscia et al. 2020].

Decomposing an object into multiple pieces also involves the related assembly problem. In that case, relying only on the surface might be insufficient to model the involved physical constraints. The volumetric decomposition strategy proposed by Araújo et al. [2019] analyzes the spatial movement of pieces to guarantee the existence of an assembly sequence. This technique focuses on assembling multi-material parts. Contrary to constrained fabrication techniques (like molding or subtractive manufacturing), the components are fabricated using additive manufacturing; hence, they are not required to be height fields. Moreover, the method takes the initial surface decomposition as a user-given input. In contrast, in the context of mold generation, the definition of such decomposition is the actual problem that needs to be solved.

2.2 Segmentation for Subtractive Manufacturing

Similar to rigid mold casting, the classic 3-axis milling process has critical restrictions in the class of manufacturable shapes. For example, the accessibility requirement for a 3-axis milling tool is similar to the conditions required by rigid casting; that is, the surface to be milled must be accessible to the milling head. A common strategy to overcome these limitations is to decompose the object into multiple height fields. Alemanno et al. [2014] proposed one of the

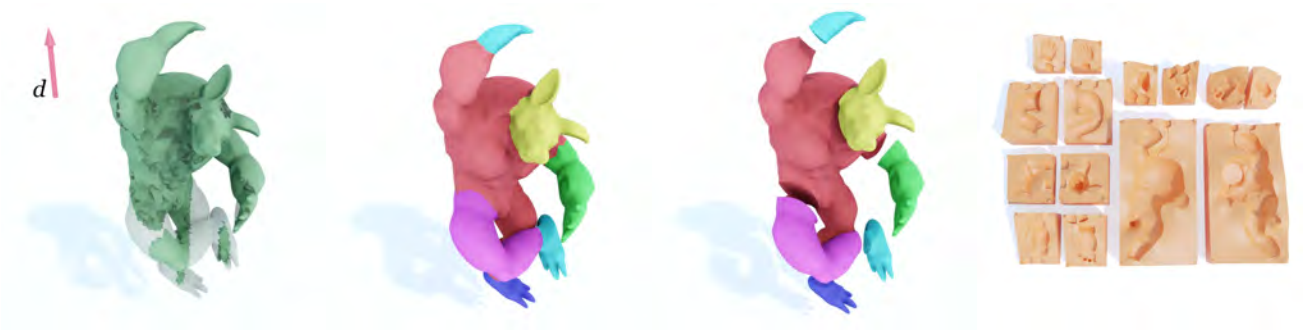


Fig. 2. An overview of the proposed pipeline: (a) the DHF-accessible region (green) for a direction d (red arrow) visualized within the object volume; (b) the DHF segmentation obtained on the mesh tetrahedralization; (c) the final resulting solid parts that can be manufactured using (d) pairs of generated rigid molds for each piece.

first methods to allow the production using 3-axis CNC milling by manual decomposition of shapes into multiple height fields.

Fanni et al. [2018] proposed a polycube decomposition suitable for additive and subtractive techniques. The approach proposed by Muntoni et al. [2018] decomposes 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 milling. This method outputs an axis-aligned block decomposition. While enforcing planar cuts might be a necessity for 3-axis milling, in the case of molding, such constraint is not needed and it might lead to an unjustified high number of parts. Instead, designing the segmentation problem to our specific application domain allowed us to derive decompositions with fewer parts.

DHFSlicer [Yang et al. 2020] replaces the single height-field decomposition with a more general double height field, which allows decomposing objects into small sets of height-bounded 3-axis millable parts (slices). Although we share the ultimate goal of obtaining double height fields, DHFSlicer was designed with a very different application goal, namely reducing milling time and material waste by targeting the processing of slabs of material. Although reasonable for slab milling, this approach, which results in tens of parts, is unfeasible for casting.

Our solution strategy differs significantly and produces results that are much better suited for molding, as shown in the results section (Figure 16). In more detail, DHFSlicer employs a binary-space partition strategy that generates parts that are *local* double height fields. On the other hand, by solving a volumetric labeling problem, our method produces parts with general internal surfaces that fully satisfy the double height field constraint.

DSCarver [Zhao et al. 2018] segments a shape to optimize the different orientations for subtractive manufacturing using a (3+2) axis milling machine. The method considers the volume surrounding the object to estimate the accessibility of the milling tool. Thanks to the extra degrees of freedom offered by the setup, this method can produce significantly more complex, even high-genus, objects as a single piece. The VDAC method [Mahdavi-Amiri et al. 2020], instead, strives 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/repositioning while giving priority to the former. Similarly, the work of Nuvoli et al. [2021] decompose the object surface into portions that can be individually manufactured using a 4-axis milling machine.

2.3 Segmentation for Casting

Height field decomposition is also used in casting to ensure mold extractability. Herholz et al. [2015] automatized this process by proposing an automatic method to segment the surface into different height fields that correspond to the components of a multi-piece rigid mold. A more radical approach, proposed by Stein et al. [2019], bi-partitions an input surface mesh for two-piece rigid casting. While these methods offer a practical fabrication approach, their application is limited to simple surfaces. Moreover, these methods resolve the required height field constraint by deforming the input surface, significantly affecting the fidelity of the manufactured results.

Flexmolds [Malomo et al. 2016] and Metamolds [Alderighi et al. 2018] overcame the rigid molding limitations by using flexible molds, which, thanks to their elasticity, allow a relaxed castability constraint. Composite molds [Alderighi et al. 2019] consider the volume surrounding the fabricated objects to automatically generate silicone molds that include cuts and allow for the casting of highly complex shapes using two-piece molds.

While most of the previous works focus on decomposing a mold that surrounds the cast object, we focus instead on how to automatically subdivide an input object into parts that can be individually cast with an easier setup (a two-piece rigid mold). We automatically derive a set of double height fields by jointly optimizing for the number of parts and their casting directions. In this way, we avoid most of the classic physical limitations involved in multiple-piece casting, extending the range of shapes fabricable using rigid casting.

3 MOTIVATION

The automatic design of rigid molds is a complex problem involving hard geometric constraints that ensure casting feasibility. Compared to flexible molding, the design of rigid molds is much more constrained and therefore generally more difficult. Given an input shape, an effective mold assembly for reproducing it should be composed of rigid mold pieces that satisfy the following constraints:

- **(C1) Surface Partition:** each portion of the cast object surface must correspond to one and only one mold piece.
- **(C2) Height Fieldness:** the surface cast by a mold component must be a height field with respect to a direction d_i to ensure that each mold component can be removed from the cast with a linear translation along a direction d_i.
- **(C3) Extraction Path:** each mold component must be able to be extracted at least along one direction d_i without intersecting either other mold pieces or the cast object.

Deriving an optimal mold assembly satisfying these constraints in order to minimize, for example, the number of mold components, requires exploring a huge solution space. Attempts have been made to solve this problem by relaxing some of the constraints. For example, the approach by Herholz et al. [2015] models the problem as an integer linear program to partition the object surface into a set of height fields, returning a solution that satisfies (C1) and (C2). In order to have a limited number of parts, the method partially neglects the height field constraint in small regions, but it is later enforced in a post-processing step by deforming the object geometry. Moreover, the actual volume of the mold pieces is implicitly derived from the surface partitioning, without any guarantee regarding (C3). Ultimately, this method could fail to produce a usable mold assembly, even for relatively simple shapes. If successful, it may significantly change the geometry of the cast object. Following mold design practices commonly employed for industrial applications, (C3) is usually enforced by carefully planning the paths and the stages of extraction for each piece.

Our goal is to enable the rigid mold casting of generic objects (ranging from free-form to mechanical shapes) and to preserve their original shape. To reach this objective, we shifted the paradigm from finding a mold decomposition to partitioning the object volume into a set of parts that can be easily produced by rigid casting and subsequently assembled to form the input object. This is a commonly used strategy in the industrial setting, where decomposing an object into simpler parts (i.e., double height field parts) makes the fabrication process cheaper and more efficient. While on the one hand, breaking the object into multiple parts could hinder its structural robustness, on the other hand, it dramatically reduces the complexity of the molds. This loss of robustness can be limited using strong glues or more sophisticated binding techniques during assembly. Moreover, since our method natively supports the decomposition of full objects (as opposed to thin shells), larger adhesion areas and more effective binding becomes possible. Additionally, decomposing the object into smaller parts allows us to define a novel criterion for global accessibility, the double height field accessibility. This criterion ensures that the parts can be cast and then assembled using simple linear movements, enabling the fabrication of objects using common plastic injection machinery, which is a very high throughput fabrication technology currently used in mass production of plastic objects.

Multiple two-piece rigid molds. The choice of *to split the cast object, rather than the mold*, renders the problem more manageable and also offers a series of advantages:

- We solve by design (C3), the Extraction path constraint, by decomposing the object into parts that can be individually cast using two-piece rigid molds. By doing so, we avoid any collisions by mold parts along their extraction direction since, pairwise, they always have two opposite, feasible extraction directions. With these simplification, we only aim for optimal decomposition of the volume, where each segment can be represented with two opposite height fields and hence will be moldable using two-piece rigid molds.
- It provides us with the ability to accurately produce intricate shapes that were not achievable with previous methods for rigid casting [Herholz et al. 2015]. As we previously explained, having a single multi-piece mold reduces the solution space sensibly, and so it can practically limit the class of fabricable objects. This is especially true with high-genus shapes, where mold pieces can mutually collide during the extraction process. Those methods are restricted to shapes with relatively simple topology or have to modify the input surfaces to cope with a smaller solution space. Instead, the proposed method can fabricate both objects with a complex topology (see Figure 14) and objects with complex surface detail (see Figure 1), preserving their original shape entirely.
- It can enable the rigid cast manufacturing of objects that are generally not feasible with a single mold assembly, like a bottle with a thin neck or simple hollowed shapes.
- We obtain a set of parts that are generally compact and easy to cast, either manually or using a robotized industrial pipeline.

Volumetric segmentation. As opposed to most of the segmentation for fabrication methods, we resort to a volume segmentation rather than a surface segmentation. While this choice might considerably increase the complexity of the problem, using a volumetric mesh offers a series of practical advantages:

- Volumetric representation allows us to derive a more compact partitioning. Intuitively, if we consider only the surface of an object, parts that are close in a volumetric sense might be quite distant using the geodesic metric. These assumptions might produce unexpected effects on the final segmentation. As an extreme example, related to the point above, let us consider a hollow cube. In this case, the external and internal surfaces are not even connected, so they will never share the same partition. Instead, relying on a volumetric representation allows us to effectively bypass this problem.
- In contrast with methods that partition the volume by relying only on the information present on the surface [Araújo et al. 2019; Livesu et al. 2020], solving the segmentation as a volumetric labeling problem implicitly provides the geometry of the internal cuts (see Figure 2c).

4 METHOD

Our main goal is to decompose the volume of a watertight 2-manifold surface mesh into a collection of *double height fields*. A double height field (or DHF) is a geometry that can be represented with a pair of opposite height fields. We initially discretize the volume of the input mesh by tetrahedral tessellation. In our work, we used TetWild [Hu et al. 2020, 2018] to produce a tetrahedral mesh from a triangular input mesh.

Our processing pipeline is composed of the following steps (see Figure 2):

DHF-accessibility In the first stage of the algorithm, we estimate the *DHF-accessibility* of the tetrahedral elements for a set of candidate extraction directions. Given a direction d, a binary field expresses the capacity of each tetrahedron t_i to be extracted along that direction d (see Figure 2a).

Volume segmentation We segment the object volume into parts by associating one label with each tetrahedron. Each label corresponds to an extraction direction d_i that determines the double height field representation of the geometry it is assigned to. Each connected set of elements with the same label form a volumetric part that will be cast with a two-piece rigid mold. We derive an optimized labeling by solving a graph-cut problem that considers all the tetrahedrons and all candidate directions. Our graph-cut formulation blends DHF-accessibility and compactness of parts (see Figure 2b).

Internal surfaces regularization As a byproduct of the volumetric segmentation, we obtain new inner surfaces between adjacent parts. Since these surfaces are derived from the tetrahedral mesh partitioning, they contain high-frequency artifacts from the volumetric tessellation. These high frequencies are very likely to invalidate the DHF property of the parts. We therefore smooth the internal surfaces emerging from the segmentation with a novel set of constraints to ensure that the final surfaces of each part is actually a DHF (see Figure 2c).

Mold generation Finally, for each part, we generate a couple of rigid molds using an algorithm similar to Alderighi et al. [2018], using the boolean operations defined in [Zhou et al. 2016]. As proposed by Alderighi et al. [2019; 2018], we also equip the resulting molds with channels and holes to enable the casting of liquid material and allow the air to escape during the manufacturing process (see Figure 2d).

4.1 DHF-Accessibility

To decompose the mesh into multiple volumetric parts that can be successfully cast using two-piece rigid molds, we must guarantee that each part is a double height field (DHF) along two opposite directions. In theory, we can achieve this goal by simply splitting the mesh into countless parts. However, for obvious reasons, we are interested in using as few parts as possible.

Formalizing the space of all the possible segmentations of a closed surface where each portion can be represented as a DHF is a challenging task because of the multiple constraints involved. Checking if a part is a valid height field requires analysis of its entire shape in every possible direction. Consider the example in Figure 3a. In this case, we defined a valid DHF partitioning using a single direction. Yet, adjacent regions of the mesh must be distinct. Modeling such behavior will inevitably produce an enormous solution space that is difficult to explore and optimize efficiently.

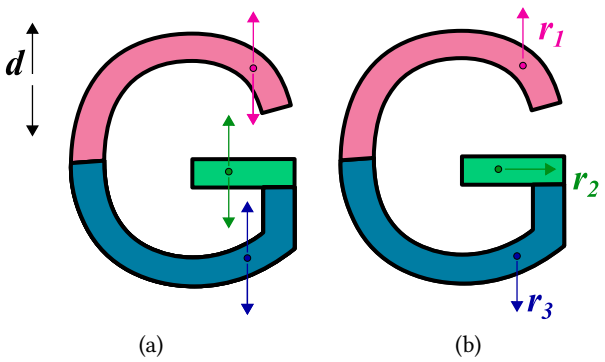


Fig. 3. (a) A valid partitioning of the mesh into double height fields, considering a direction d; (b) The partitioning induced by double height field accessibility and the relative directions.

To make the problem tractable, we reduce the space of possible DHF segmentations to the space of possible *double height field-accessible* segmentations. We can compute if a point p is DHF-accessible along a direction d if shooting a ray along direction d from p does intersects with the mesh surface only once. It is significant that DHF-accessibility depends only on the point, the direction, and the global shape, bypassing the need to consider the shape of the part explicitly. Thanks to this simplification, we can first precompute a set of valid directions for each point and then cluster points with compatible directions into partitions, shifting the focus of the optimization problem to the compactness and the smoothness of the produced partitions. The DHF-accessibility criterion is shown in Figure 3b. Figure 4 shows two examples of DHF-accessible regions in two opposite directions. Notice that, while the region is different, both configurations generate a valid DHF. Hence, if a region is DHF-accessible with respect to a direction, then it will be a double height field with respect to that direction and its opposite and so fabricable with two-piece rigid molds.

Although our simplified DHF-accessibility excludes some valid double height field partitionings, it comes with a great advantage: the direction set defines a sequence of part movements to assemble the shape. So, in other words, DHF-accessibility ensures both double height field decomposition and the existence of a valid assembly procedure. In our formulation, when a portion of the mesh is DHF-accessible along a specific direction, then it is possible to move it along that direction without intersecting any other part of the mesh. It follows that we can simply translate each part along its direction to assemble the final object.

In the discrete setting, we first pre-define a set of uniformly distributed directions $d_1, ..., d_n \in \mathcal{D}$. In our implementation, we sampled 512 directions uniformly distributed on a sphere using

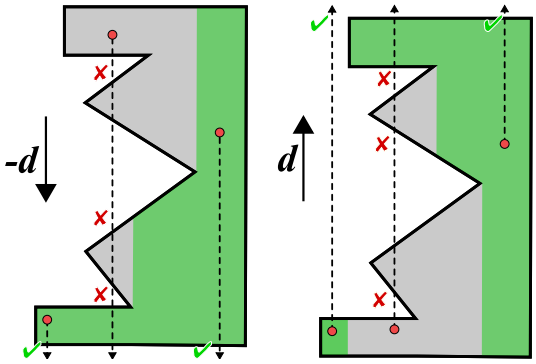


Fig. 4. DHF-accessibility for a point p along a direction d is verified when the ray shot from p along d intersects the surface only once. Notice that two opposite directions can generate different DHF-accessible regions (green), depending on the object shape.

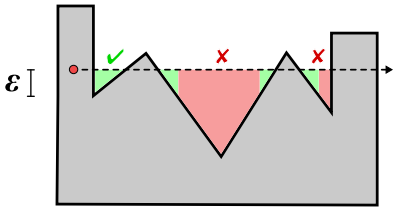


Fig. 5. Illustrating the tolerance on the DHF-accessibility evaluation.

the strategy proposed by Keinert et al. [2015]. Additionally, to express the DHF-accessibility of a tetrahedron, we check it for every point within its volume. Implementing this test might require complex geometric operations, including sweeping volumes and boolean operations. For practical purposes, we only estimate the DHF-accessibility of a set of points carefully placed within a tetrahedron's volume. Therefore, in our setting, we consider a tetrahedron t_i DHF-accessible along direction d_j if and only if it is DHF-feasible on all its sampled points. In our implementation, we sample 15 points for each tetrahedron: the four vertices, the six edges' midpoints, the four faces' barycenters, and the barycenter of the entire tetrahedron.

We add a geometric tolerance to regularize the accessibility field estimation and so the regularity of the extracted parts. This tolerance factor allows for control of the amount of undercut that is considered to be admissible (taking into account that plastic materials like the mold itself and the object can slightly deform during the extraction procedure), allowing the object to snap out of mild undercuts. To do so, we add a tolerance factor, ϵ, in millimeters that is controlled by the user. Given a ray r, for each intersection h_i of r we sample the distance of the ray segment (h_{i-1}, h_i) from the mesh surface. If this distance is below ϵ at every sampled point, the intersection is ignored and the ray passes the accessibility test (see Figure 5). To efficiently compute the ray-surface intersections, we use Embree [Wald et al. 2014].

4.2 Volume Segmentation

Given a tetrahedral mesh $\mathcal{T}$, for each tetrahedron t we want to assign one direction $d \in \mathcal{D}$. In other words, we look for a labeling function $\ell : \mathcal{T} \rightarrow \mathcal{D}$ that assigns a direction (label) to each tetrahedron in $\mathcal{T}$. Intuitively, we want to associate to each tetrahedron with a valid direction with respect to the DHF-accessibility constraint while also encouraging solutions with a small number of parts and smooth boundaries.

Solving a multi label assignment problem is known to be NP-Hard. To solve it efficiently, we use the graph-cut optimization framework (Section 4.3.3). The graph-cut algorithm derives an approximation of the optimal labeling by minimizing an energy function in the form of

$$\arg\min_{\ell} \sum_{t \in \mathcal{T}} D\left(t, \ell\right) + \sum_{(t_i, t_j) \in \mathrm{Adj}_{\mathcal{T}}} S\left(t_i, t_j, \ell\right) + L(\ell)\,, \qquad (1)$$

where $D(t, \ell)$ is the data term and describes the cost of assigning the direction $d = \ell(t)$ to the tetrahedron t (Section 4.2.1). $\mathrm{Adj}_{\mathcal{T}}$ is the set of face-adjacent tetrahedra pairs in the tesselation $\mathcal{T}$. The energy term $S(t_i, t_j, \ell)$ is a smoothness term that penalizes the overall boundary area between adjacent parts and determines their shape and locations (Section 4.2.2). We use this term to support the formation of compact and smooth parts. The smoothing term also avoids the energy minimization to converge to the extreme situation where each tetrahedron is associated with a different direction; hence it implicitly favors solutions with a smaller number of parts. We will describe this term in Section 4.2.2. The rightmost term L is a label cost term that penalizes the use of many labels in the segmentation (Section 4.2.3).

4.2.1 Data Cost. To obtain a fabricable solution we must associate to each tetrahedron one direction for which it is DHF-accessible. Since accessibility is a discrete boolean function, the data cost will generate intervals with constant energy:

$$D(t, d) = \begin{cases} 0 & \text{if } t \text{ DHF-accessible for } d \\ +\infty & \text{otherwise.} \end{cases} \qquad (2)$$

Similarly to the approach by Herholz et al. [2015] (which segments the external surface only), $D(\cdot)$ does not provide a smooth signal. In our approach, to improve the minimization of the data cost and regularize the problem, we propose a smooth data cost function that favors direction assignment to tetrahedra that are organized in large clusters.

Therefore, we introduce a penalty term, per tetrahedron per label, for all tetrahedra that result DHF-accessible from a direction (the others keep the infinite penalty term). Given a direction d, we define the function $dist_d(t)$ as the distance from t to the closest tetrahedron that is not feasible (DHF-accessible) for d. When $dist_d(t)$ is large, it means that direction d is a good choice for tetrahedron t, because all nearby tetrahedra are also feasible for that direction. Therefore, we define the penalization term for tetrahedron t and a direction d as:

$$p(t, d) = \alpha \cdot e^{-\frac{dist_d(t)^2}{2 \cdot c^2}}\,.$$

This cost smoothly converges to α as we get closer to the unfeasible region and approaches 0 around $4 \cdot c$, following a Gaussian bell shape.

The parameter c controls the width of the Gaussian bell, which in our context is related to the spatial influence of the penalization term. The data cost then becomes:

$$D(t,d) = \begin{cases} \text{Volume}(t) \cdot p(t,d) & \text{if } t \text{ DHF-accessible for } d \\ +\infty & \text{otherwise,} \end{cases} \quad (3)$$

where the volume of the tetrahedron weights the resulting data cost. In our experiments we set the parameter $\alpha = 5$ and the parameter $c = 10$.

4.2.2 Smoothing Cost. We introduce a smoothing term to penalize the formation of irregular borders between parts. This term supports the formation of compact and smooth parts. Also, it avoids the energy minimization process to converge to the extreme situation where each tetrahedron is associated with a different direction while implicitly favoring solutions with a smaller number of parts.

For each pair of adjacent tetrahedra $(t_i, t_j) \in \text{Adj}_{\mathcal{T}}$, assigned with directions $(d_i, d_j) = (\ell(t_i), \ell(t_j))$, we define the energy term

$$\sigma(t_i, t_j) = \begin{cases} 0 & \text{if } d_i = d_j \\ \beta \cdot Area(t_i, t_j) & \text{if } d_i \neq d_j \,, \end{cases}$$

where $Area(t_i, t_j)$ is the area of the triangular face shared between t_i and t_j. When multiplying an area by β, we transform the whole term into a volume quantity, which is comparable with the data cost (also expressed as a weighted volume). In our setup we use *mm* as base unit and therefore β represents a length which was set to 6 *mm*.

Additionally, to penalize solutions with longer seams on the surface, we define an additional smoothness term penalizing surface edges shared between tetrahedrons having different labels:

$$\overline{\sigma}(e_{i,j}) = \begin{cases} 0 & \text{if } d_i = d_j \\ \beta^2 \cdot (1 + AO(e_{i,j}) \cdot \text{len}(e_{i,j})) & \text{if } d_i \neq d_j \,, \end{cases}$$

where $e_{i,j} \in \Omega_{\mathcal{T}}$ is an edge on the mesh surface shared between two tetrahedra t_i and t_j, which have at least one face on the object surface. This penalization term also accounts for the visibility of the surface seams, similarly to the approach in [Filoscia et al. 2020]. To favor the formation of surface seams on less visible edges, we add a normalized term $AO(e_{i,j})$ that is equal to 0 where the ambient occlusion is maximum and 1 on the least occluded edges.

Eventually, our smoothing cost becomes:

$$S\left(t_i, t_j, \ell\right) = \sigma(t_i, t_j) + \overline{\sigma}(e_{i,j}) \,.$$

4.2.3 Label Cost. In our implementation, a constant penalization factor is added for any label assigned in the solution; in particular, the cost is computed as

$$L(\ell) = \lambda \cdot \text{Volume}(\mathcal{T}) \cdot |\{\ell(t) : t \in \mathcal{T}\}| \,,$$

where we set as weighting factor $\lambda = 0.01$.

4.3 From Segmentation to Solid Parts

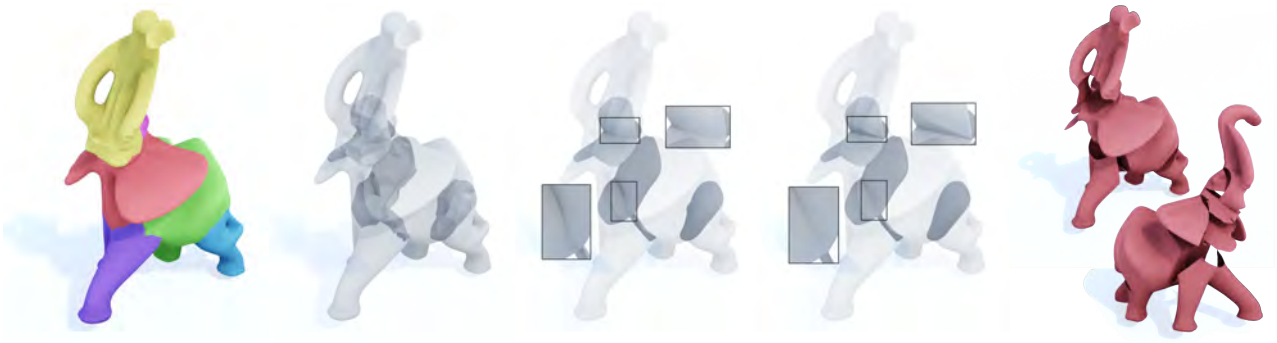
Once we have derived an optimized labeling, the object volume can be decomposed into solid parts represented by a double height field. Note that, at this stage, the optimized labeling could assign the same label, and thus direction, to multiple connected components of the volume. Whenever this is the case, we consider each connected component of the labeling as a separate solid double height field part.

Fig. 6. Boundary seams smoothing. Left: the surface seams polyline (green), derived from the DHF partitioning, can freely move within a tolerance region (gray) while preserving DHF-accessibility of the surface regions it delimits. Right: the same polyline is smoothed using the tolerance region as smoothing domain.

4.3.1 Internal surfaces smoothing. When we partition the volume into multiple components, we also create new boundary surfaces between all pairs of adjacent parts. Unfortunately, due to the internal tetrahedral tessellation, these boundaries are not smooth (Figure 7b). Also, our DHF-accessibility is computed using ray-intersections only against the external surface of the object (see Section 4.1). For these two reasons the inner boundaries of the parts, defined by the raw tetrahedral decomposition, can exhibit some undercuts and are not guaranteed to be double height fields. We therefore smooth the boundaries between parts and enforce the double height field property using a two-step approach.

In the first step, we smooth the seams on the mesh surface by iteratively performing laplacian smoothing of the seams polyline, and reprojecting it on the mesh surface. Unfortunately, smoothing the partitioning borders (the seams) might invalidate the double height field property. Therefore, to avoid this undesired effect, we create a tolerance region on the mesh surface where the boundary polyline can freely move without invalidating the accessibility (see Figure 6) and use it to constrain the smoothing and reprojection of the boundary polyline. To obtain this region, after the DHF-accessibility computation step, we perform a morphological erosion of all feasible tetrahedral regions for every direction. The effect of such morphological erosion is to consider as not feasible any tetrahedron that is vertex-adjacent to some not DHF-accessible tetrahedron. This procedure guarantees that for any point in the volume, assigned to a direction d, there is a neighborhood that can be also associated to d retaining the DHF-accessibility property.

In the second step, once we have smoothed the seams on the mesh surface, we smooth the internal surfaces keeping their border fixed (Figure 7c). In this step, we enforce the double height field property by adopting an iterative approach that smooths the surface and reprojects it outside a volume where the DHF-accessibility will



(a) (b) (c) (d) (e)

Fig. 7. Processing of internal surfaces: (a) starting from the tetrahedral segmentation, (b) the inner surfaces are defined as the boundaries between parts; (c) a regular smoothing can violate the double height field constraint; (d) the constraints is enforced through reprojection outside the "infeasible volumes", red in (e).

be not preserved for the chosen directions (Figure 7d). For each part, we compute the volumetric region that all inner boundaries must avoid to preserve the DHF-accessibility. Let's consider a single component of the optimal segmentation (Figure 8a); we compute its "infeasible volume" as the intersection of the sweeping volumes of the object external surface of the part, as seen from direction assigned to it d and $-d$ (Figure 8b). Intuitively, given a direction d and a surface $\mathcal{S}$ to be extracted along d, any part of $\mathcal{S}$ that is inside the sweeping volume of $\mathcal{S}$ with respect to d won't be extractable along that direction. Thus, for a double height field, any point in $\mathcal{S}$ must be outside the intersection of both its sweeping volumes with respect to d and $-d$. This represents a *necessary condition* to ensure the DHF-accessibility. Hence, we move each point of the surface which falls within this volume on the volume boundary (Figures 8c,d), by iteratively cycling between laplacian smoothing and damped projection steps.

Notice that while each part is guaranteed to be a DHF with respect to the object's surface, some portions of the internal surfaces might be occluded with respect to both d and $-d$, as the condition explained above is not sufficient. This situation is quite rare in practice but we still solve it using a conservative strategy. To accommodate the DHF condition we carve out a portion of the volume from the boundary surface, including the occluded region. As a result, this strategy generates one or more holes inside the assembled full object and, while we sacrifice the perfect match across adjacent parts, the process has no impact on the object's external surface.

4.3.2 Mold generation. Given the set of solid parts, we proceed to generate the mold geometry for each ones. Each part is a double height field with respect to the direction assigned to it by the volume segmentation (Section 4.2). At this point, given a direction d, we need to sample the actual surface accessibility along direction d and $-d$ for all the faces on the part surface. For every triangle we compute the ray accessibility using the same strategy and tolerance we defined in Section 4.1. This will partition the surface into three sets: the faces accessible along d, the ones accessible along $-d$, and the faces accessible for both directions. To assign each face to its mold piece (corresponding to d or $-d$), we solve a small graph cut problem where each face that is shared by the two opposite directions is assigned to the one that is better aligned to its normal, also considering a smoothness term to keep the boundary as smooth as possible. The decomposition boundary is usually jagged, as it is bound to lie on the discrete triangular edges. We smooth the boundary polyline using a laplacian smoothing on the tangent plane with respect to the object surface, where the vertices of the polyline can move only within the regions of the surface that are accessible by both d and $-d$, while parts of the polyline where this region is empty are fixed. This smooth polyline is the starting point for generating the two-piece mold parting surface using the approach described in [Alderighi et al. 2018] based on Poisson Disk Reconstruction.

Given the mold parting surface geometry, one key detail for a successful mold design is the placement of the air vents for material casting and for the release of the air that would eventually get trapped inside the mold volume during the pouring [Alderighi et al. 2019, 2018; Malomo et al. 2016]. The generation of air vents for a rigid mold poses two main problems:

- the geometry of the air vent itself needs to be extractable along the two height field directions;
- air vents should be located at all local maxima within the mold cavity, where air could be trapped.

Thanks to the double height field property of our decomposition, we can prove that it is always possible to design a mold where both the aforementioned requirements are satisfied. In particular, we can prove that, if we carefully choose the gravity direction for material casting, local maxima will always be located on the boundary between the two height fields composing the object part. As a consequence, the air vents geometry can be generated such that the vent completely lies on the mold parting surface, and thus their double height field property can be enforced. This concept is illustrated in Figure 9. Given a double height field surface along a direction d, for any direction y orthogonal to d all local maxima are located on the boundary between the d *region* and the $-d$ *region* (Figure 9 left). Otherwise, the part would violate the height field property with respect to d (Figure 9 right) or $-d$.

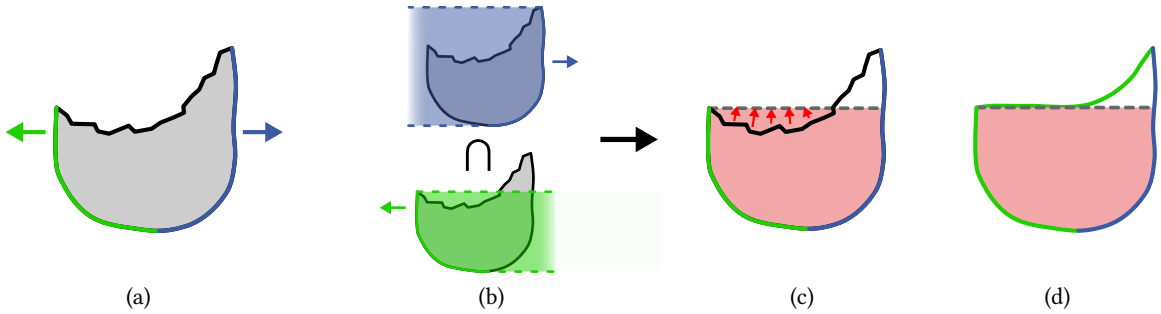


Fig. 8. (a) To make the inner boundary surface accessible (black), we first compute two volumes by sweeping the segmented surface regions (green and blue) along the opposite of their extraction direction (b); we then compute their intersection (red), and we force the inner boundary surface to stay outside this volume (c), obtaining a new surface that does not collide with the part external boundary during the extraction (d).

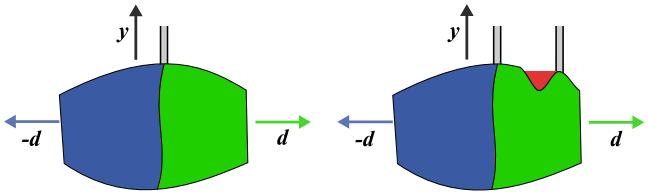


Fig. 9. Left: Air vents must be placed on the boundary between the two height field composing the mold (respectively from d and $-d$). Right: the presence of a local maximum (with respect to y) outside the boundary violates the height field property with respect to d (b). The undercut is highlighted in red.

Fig. 10. For a simple object, our method can derive a single two-piece rigid mold as in [Stein et al. 2019].

4.3.3 Solving Using Graph-cut. In our pipeline we solve the partitioning problem using the graph cut optimization framework (*gco-v3.0* [Boykov et al. 2001; Delong et al. 2012]), which relies on the α-expansion algorithm.

The iterative nature of the α-expansion algorithm makes it sensitive to the initialization of the optimization algorithm and the ordering in which the labels are evaluated during the α-expansion cycles. To reduce the influence of these two factors on the quality of the solutions, we repeat the minimization process multiple times, randomizing both the algorithm initialization and label orders, as suggested in [Herholz et al. 2015]. In our setup, the graph-cut minimization algorithm is ran 15 times, and the best solution is selected, among the resulting minimizers, to be the one that minimizes the number of parts in the decomposed object. In case of a tie, the algorithm selects the solution that minimizes the variance of parts volume to favour the more balanced solutions.

5 RESULTS

We tested our method on a series of 3D models with varying geometric complexity. For simple objects our method can successfully derive a single two-piece rigid mold (e.g., the cactus model in Figure 10). We tested our method for the fabrication of shapes featuring complex geometric features (Figure 11) or topology (Figure 14). Our method is able to decompose the tested shapes into a manageable number of parts, and we demonstrated the effectiveness of the resulting two-piece molds by physically fabricating the parts using casting. We experimented with our pipeline by manufacturing the set of two-piece molds for eight models in total (see Table 1). For practical reasons, we fabricated the molds using additive manufacturing technology. We used two different 3D printers: a Utimaker S5 and Stratasys J750. Our test models were manufactured by liquid casting all parts using a variety of materials, including white resin, gypsum and translucent resin. Moreover, since our molds are height fields, they can also be manufactured using a 3-axis milling, enabling the casting of a wider range of materials (e.g., metal).

Actually, since the parts resulting from our decomposition can be represented by double height fields, they could be fabricated using subtractive manufacturing, as in DHFSlicer [Yang et al. 2020]. However, to effectively manufacture either mold or object parts through subtractive technologies, our method would also need to consider the shape of the milling tool when evaluating the DHF-accessibility.

An additional advantage of decomposing the object into parts, rather than generating multi-piece molds, is that it enables the generation of valid molds for objects featuring inaccessible cavities that are impossible to cast as a single piece using a reusable mold. In Figures 12 and 14 (bottom) we show two examples of such shapes and the resulting decomposition obtained with our method. Defining a optimal DHF decomposition ground truth to compare against would be challenging since the optimality criteria are often related to the considered shape and its final purpose. However, in the context of an automatic, general-purpose decomposition algorithm, Figure 13 shows how our method can output optimal or near-optimal solutions.

Fig. 11. Segmentation and fabrication result: for each model we show the tetrahedral decomposition, the final DHF parts, the fabricated molds with their respective cast part, and the final assembled results.

We report some statistics of the processing time in Table 1. Computation has been performed on a machine equipped with a 8-core AMD Ryzen 7 2700X CPU clocked @ 3.7GHz and 32 GB of RAM. We can observe that the computation time for DHF-accessibility is proportional to the number of tetrahedra used to tessellate the volume of the object, while the derivation of the segmentation, on the other hand, depends on the overall complexity of the shape and the structure of the DHF-accessibility fields.

5.1 Parameters

Figure 17 shows the impact of the the undercuts tolerance ϵ (Section 4.1), as well as the impact of the regularized data cost and smoothing factor (Section 4.2), on the quality of the segmentation results. Using a smaller undercut tolerance will make the DHF-accessibility estimation more sensitive to high-frequency surface details and discretization noise. This results in partitions with a higher number of parts (see Figure 17, top), that better adhere to the height field constraint. Increasing the allowed undercut depth will result in a lower number of parts. Indeed, some casting (and mold) materials can bear the relatively small deformations that occur during the extraction process, guaranteeing extractability of the parts for small values of ϵ. We used PLA and ABS for the molds, a shore 70D urethane resin for the casts, and set $\epsilon = 0.4$ mm for all our experiments. In Figure 17 (middle) we show the impact of using a smooth data cost with respect to a binary data cost (see Section 4.2.1). When

Table 1. Statistics for the performed experiments: for each model, we report the size, the number of tetrahedra, and the computational time (in seconds) for the main steps of our pipeline: DHF-accessibility computation for 512 directions, the solution of the graph-cut partioning problem, boundary smoothing and generation of the solid DHF parts, and generation of the mold pieces. Finally, we show the resulting number of DHF parts.

Model	bbox size (mm)	#tets	DHF time	Solve time	Partsgen time	Moldgen time	#parts	real cast	Fig.
Elephanticus	115 × 80 × 106	88928	232	609	335	2113	9	✓	1
Dancing Armadillo	98 × 199 × 95	57875	136	466	175	1628	7	✗	2
Elephant	143 × 199 × 120	50589	114	264	273	1629	7	✗	6, 7
Duck	70 × 80 × 83	94056	157	1911	189	433	2	✓	11
Bunny	95 × 94 × 74	82234	156	611	130	1248	4	✓	11
Cactus	49 × 82 × 12	3006	5	2	19	242	1	✓	10
DualDodec	47 × 47 × 147	47042	122	218	257	818	3	✓	14
Bottle	87 × 350 × 88	13364	36	51	230	909	4	✗	12
Fertility	173 × 129 × 67	24215	41	147	104	545	2	✓	14
Magalie	53 × 120 × 66	114546	320	927	208	634	3	✓	11
Table	80 × 68 × 79	85024	266	606	188	1295	5	✓	11
Julia	117 × 119 × 150	42562	144	1255	107	1255	4	✗	16
Bimba	87 × 100 × 72	59077	236	329	209	795	5	✗	17
Armadillo	83 × 99 × 75	57681	123	331	124	1624	6	✗	17
Pulley	98 × 100 × 48	50066	127	139	368	566	2	✗	13
DoubleTorus	166 × 138 × 112	19527	28	45	5	224	1	✗	13
Ujoint	154 × 77 × 77	56507	92	448	685	423	2	✗	13
DragonStand	99 × 70 × 44	75500	203	484	355	1610	7	✗	17

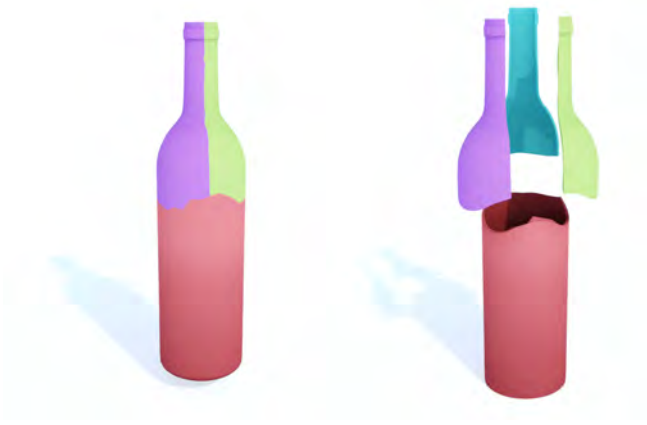


Fig. 12. A model with a complex occlusion configuration, like a bottle, can be successfully automatically decomposed and fabricated with the proposed method.

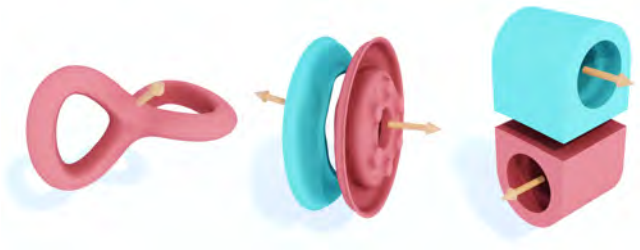


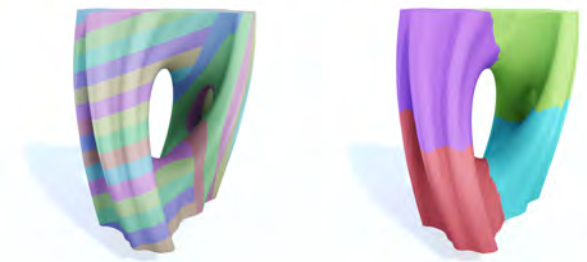
Fig. 13. Outputs of our method on simple shapes. Our solutions closely match optimal DHF manual segmentations.

using a binary data cost, the solver tends to output unbalanced decompositions. This is due to the fact that the minimized energy only depends on the smoothness cost, and thus on the size of boundaries involved in the segmentation. In this setting, if the space of feasible assignments includes a partitioning with a few large parts and a set of smaller ones to fill out the inaccessible volume, the minimizer will pick that solution. Enforcing a smooth data cost, penalizing the assignment of a direction near the boundaries of its feasible region, favors more balanced solutions. This can sometimes result in a slightly larger number of parts, but at the same time, favors the generation of more *fabricable* parts. Finally, in Figure 17 (bottom) we show the effect of the smoothing factor described in Section 4.2.2 on the output decomposition. The image shows the result of our DHF decomposition using different β values: on the left, with $\beta = 0$ the output features high fragmentation counting 469 parts; in the middle, with $\beta = 1$ the resulting decomposition is made up of 13 parts; finally, on the right, for $\beta = 6$ (the value used for the results in the paper), the decomposition results in 7 parts. The comparison shows how adding a penalization term on the size of the segmentation boundaries that considers both the internal surfaces and the length of the surface seams, has an impact on the regularity of the decomposition boundaries and also on the fragmentation. In fact, given our definition of DHF-accessibility and the nature of the problem we solve, it is hard to establish a penalization or reward factor for assigning a given direction label to a specific tetrahedron. This make it crucial to introduce a regularization factor (i.e., a smoothing

Fig. 14. Segmentation and fabrication of shapes with complex topology.

(a) CoreCavity (2 pieces) (b) Ours ($\epsilon = 1.5mm$) (c) Ours ($\epsilon = 0.4mm$)

Fig. 15. Comparison of the proposed method with CoreCavity [Nakashima et al. 2018]. The zoomed-in areas highlight large deformations introduced by CoreCavity to ensure fabricability of parts with larger violations of the height field constraint.



DHFSlicer (26 pieces) Ours (4 pieces)

Fig. 16. Comparison of the proposed method with DHFSlicer [Yang et al. 2020].

cost) that helps the solver to converge to solutions that favor the usage of a smaller number of parts and that feature boundaries that are as smooth as possible.

5.2 Comparisons

We also compared the proposed method with the double height field decomposition produced by DHFSlicer [Yang et al. 2020] (Figure 16). While both methods succeed in deriving a valid partitioning for two-piece rigid mold fabrication, our approach produces a significantly lower amount of pieces, resulting in a more practical fabrication process.

Figure 15 shows a comparison between our method and CoreCavity [Nakashima et al. 2018] on a shelled version of the bunny model obtained from the paper supplemental material. The first difference between the methods is that, differently from CoreCavity, our

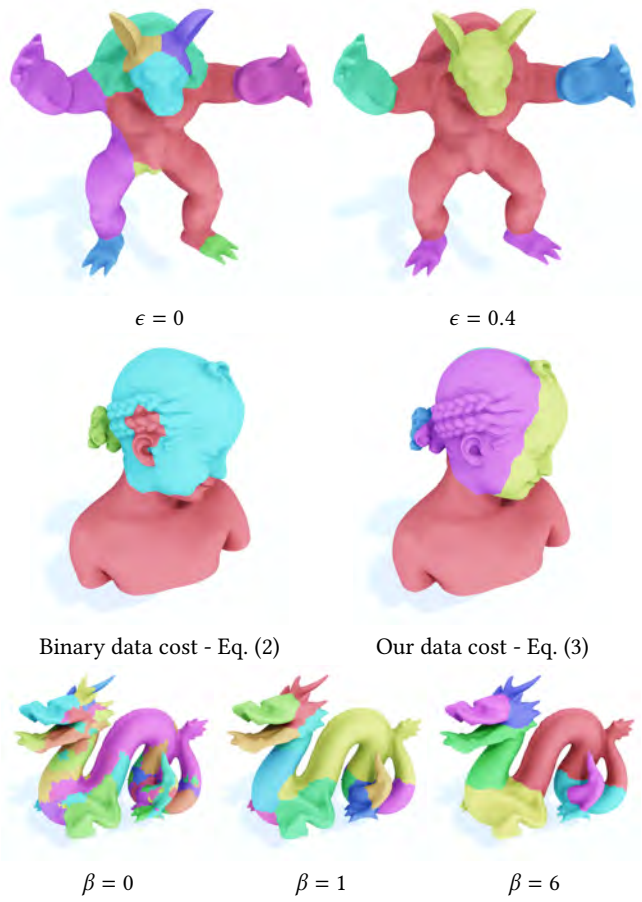


Fig. 17. The impact of the different components on DHF-decomposition output. Top: the impact of the undercut tolerance ϵ (Section 4.1). Middle: The improvement provided by regularized data cost compared to the simple binary DHF-accessibility value (Section 4.2.1). Bottom: The effect of the smoothing factor controlled by the β parameter (Section 4.2.2).

method is completely automatic and natively supports full volumetric decomposition. In some cases, thanks to the feasibility relaxation, CoreCavity might produce decompositions with fewer parts. Unfortunately, since CoreCavity does not allow for direct control of the introduced deformations, the sweeping volume strategy (used by CoreCavity to correct the undercuts) might introduce visible artifacts. This is obvious in the case of the bunny model (shown in Figure 15.a), in which the introduced deformations are larger than 2mm. In contrast, we allow the user to control the amount of tolerated undercut by tuning the ϵ parameter. We show two segmentations, one with an undercut tolerance $\epsilon = 1.5mm$ (Figure 15.b) and one with $\epsilon = 0.4mm$ (Figure 15.c). By introducing larger tolerance, our method can reach similar decompositions to the output of CoreCavity, though deeper undercuts would require special treatment such as surface deformation (similar to CoreCavity) to ensure fabricability. Using a lower tolerance instead results in a larger number of parts whose undercuts are within the tolerable limits of the physical molding process and do not require any deformation, as demonstrated by our fabricated results.

Another major limitation of CoreCavity is its lack of applicability for thick shells. CoreCavity requires a bijective mapping between the interior and exterior surfaces of the shell. However, this mapping can not exist when the external and internal surfaces have different topologies, which might be the result of an increase in the shell thickness. Moreover, CoreCavity constrains the internal cut geometry to the side facets of the prisms induced by the mapping. This poses a major limitation to the method in case of thick shells, where it would make it harder to enforce the double height field constraint. Our method tackles this problems by explicitly managing the internal cutting surface geometry.

6 CONCLUSIONS

We proposed a new method to decompose a generic input shape into a set of double height fields that can be singularly fabricated with two-piece rigid casting and assembled afterwards. Our formulation enables the fabrication of complex shapes with significantly fewer pieces than previous methods. While we used the proposed approach to produce a few physical copies in an experimental setup, our pipeline can be adapted for large-scale industrial production. Indeed, automatic decomposition for two-piece rigid casting can cope significantly well with robotized fabrication in the industrial context. In addition, our pipeline ensures the existence of a correct assembly procedure.

Our methods suffer from a few limitations. First of all, our DHF-accessibility formulation does not guarantee feasibility in the general case. For a very complex shape, some volumetric elements might have no DHF-accessible direction. For example, let us imagine an object composed of multiple nested components. The internal ones might have no accessible DHF direction. In such a case, it is sufficient to isolate that partition and decompose it in a separate step.

Moreover, since our DHF-accessibility sampling strategy and our partitioning formulation are directly related to the tetrahedral tessellation of the object volume, the discretization resolution can affect the quality of the generated results. In each of our experiments we found that a manageable number of tetrahedra were sufficient to obtain valid fabricable results. See Table 1 for details about the number of tetrahedra for all the presented examples.

Concerning limitations, while it is true that small undercuts can be tolerated, in practice, due to the generated friction, large vertical walls can make it very difficult to successfully extract the object from the mold. In such cases it is a suggested best practice to ensure a small draft angle when designing the molds. For future work, such criterion could be embedded in our DHF-accessibility formulation. Finally, as any other shape decomposition approach, breaking the object into parts can reduce its final robustness. Adding a fragility minimization factor into the proposed optimization framework is a interesting research challenge that would require further investigation. Another direct improvement of the proposed method would be including global constraints in the segmentation process, such as symmetry. Symmetric segmentation can significantly improve the aesthetic quality of the final result. One might also use methods for quadrilateral patch layout generation [Pietroni et al. 2016] to

orient the partition along curvature lines or use a precomputed volumetric block decomposition [Livesu et al. 2020] to generate a more compact partitioning. For future work, an interesting extension of our method would be to allow user-guidance for aesthetic choices, as well as investigating methods that can explore and characterize the design space of potential decompositions.

ACKNOWLEDGMENTS

The authors thank Marco Callieri for all his precious help with the resin casts. The models used in the paper are courtesy of the Stanford 3D Scanning Repository, the AIM@SHAPE Shape Repository, and Thingi10K Repository. The research was partially funded by the European Research Council (ERC) MATERIALIZABLE: Intelligent fabrication-oriented computational design and modeling (grant no. 715767).

REFERENCES

Thomas Alderighi, Luigi Malomo, Daniela Giorgi, Bernd Bickel, Paolo Cignoni, and Nico Pietroni. 2019. Volume-Aware Design of Composite Molds. *ACM Trans. Graph.* 38, 4, Article 110 (July 2019), 12 pages. https://doi.org/10.1145/3306346.3322981

Thomas Alderighi, Luigi Malomo, Daniela Giorgi, Nico Pietroni, Bernd Bickel, and Paolo Cignoni. 2018. Metamolds: Computational Design of Silicone Molds. *ACM Trans. Graph.* 37, 4, Article 136 (July 2018), 13 pages. https://doi.org/10.1145/3197517.3201381

G. Alemanno, P. Cignoni, N. Pietroni, F. Ponchio, and R. Scopigno. 2014. Interlocking Pieces for Printing Tangible Cultural Heritage Replicas. In *Proceedings of the Eurographics Workshop on Graphics and Cultural Heritage* (Darmstadt, Germany) *(GCH '14)*. Eurographics Association, Goslar, DEU, 145–154.

Chrystiano Araújo, Daniela Cabiddu, Marco Attene, Marco Livesu, Nicholas Vining, and Alla Sheffer. 2019. Surface2Volume: Surface Segmentation Conforming Assemblable Volumetric Partition. *ACM Trans. Graph.* 38, 4, Article 80 (July 2019), 16 pages. https://doi.org/10.1145/3306346.3323004

Yuri Boykov, Olga Veksler, and Ramin Zabih. 2001. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23, 11 (2001), 1222–1239. https://doi.org/10.1109/34.969114

Pritam Chakraborty and N. Venkata Reddy. 2009. Automatic determination of parting directions, parting lines and surfaces for two-piece permanent molds. *Journal of Materials Processing Technology* 209, 5 (2009), 2464 – 2476. https://doi.org/10.1016/j.jmatprotec.2008.05.051

Xuelin Chen, Hao Zhang, Jinjie Lin, Ruizhen Hu, Lin Lu, Qixing Huang, Bedrich Benes, Daniel Cohen-Or, and Baoquan Chen. 2015. Dapper: Decompose-and-Pack for 3D Printing. *ACM Trans. Graph.* 34, 6, Article 213 (Oct. 2015), 12 pages. https://doi.org/10.1145/2816795.2818087

Andrew Delong, Anton Osokin, Hossam N. Isack, and Yuri Boykov. 2012. Fast approximate energy minimization with label costs. *International journal of computer vision* 96, 1 (2012), 1–27. https://doi.org/10.1007/s11263-011-0437-z

Filippo A. Fanni, Gianmarco Cherchi, Alessandro Muntoni, Alessandro Tola, and Riccardo Scateni. 2018. Fabrication oriented shape decomposition using polycube mapping. *Computers & Graphics* 77 (2018), 183–193. https://doi.org/10.1016/j.cag.2018.10.010

Irene Filoscia, Thomas Alderighi, Daniela Giorgi, Luigi Malomo, Marco Callieri, and Paolo Cignoni. 2020. Optimizing Object Decomposition to Reduce Visual Artifacts in 3D Printing. *Computer Graphics Forum* 39, 2 (2020), 423–434. https://doi.org/10.1111/cgf.13941

Philipp Herholz, Wojciech Matusik, and Marc Alexa. 2015. Approximating Free-form Geometry with Height Fields for Manufacturing. *Computer Graphics Forum* 34, 2 (2015), 239–251. https://doi.org/10.1111/cgf.12556 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12556

Ruizhen Hu, Honghua Li, Hao Zhang, and Daniel Cohen-Or. 2014. Approximate Pyramidal Shape Decomposition. *ACM Trans. Graph.* 33, 6, Article 213 (Nov. 2014), 12 pages. https://doi.org/10.1145/2661229.2661244

Yixin Hu, Teseo Schneider, Bolun Wang, Denis Zorin, and Daniele Panozzo. 2020. Fast Tetrahedral Meshing in the Wild. *ACM Trans. Graph.* 39, 4, Article 117 (July 2020), 18 pages. https://doi.org/10.1145/3386569.3392385

Yixin Hu, Qingnan Zhou, Xifeng Gao, Alec Jacobson, Denis Zorin, and Daniele Panozzo. 2018. Tetrahedral Meshing in the Wild. *ACM Trans. Graph.* 37, 4, Article 60 (July 2018), 14 pages. https://doi.org/10.1145/3197517.3201353

Benjamin Keinert, Matthias Innmann, Michael Sänger, and Marc Stamminger. 2015. Spherical Fibonacci Mapping. *ACM Trans. Graph.* 34, 6, Article 193 (Oct. 2015), 7 pages. https://doi.org/10.1145/2816795.2818131

Alan C. Lin and Nguyen Huu Quang. 2014. Automatic generation of mold-piece regions and parting curves for complex CAD models in multi-piece mold design. *Computer-Aided Design* 57 (2014), 15 – 28. https://doi.org/10.1016/j.cad.2014.06.014

Marco Livesu, Nico Pietroni, Enrico Puppo, Alla Sheffer, and Paolo Cignoni. 2020. LoopyCuts: Practical Feature-Preserving Block Decomposition for Strongly Hex-Dominant Meshing. *ACM Trans. Graph.* 39, 4, Article 121 (July 2020), 17 pages. https://doi.org/10.1145/3386569.3392472

Linjie Luo, Ilya Baran, Szymon Rusinkiewicz, and Wojciech Matusik. 2012. Chopper: Partitioning Models into 3D-Printable Parts. *ACM Trans. Graph.* 31, 6, Article 129 (Nov. 2012), 9 pages. https://doi.org/10.1145/2366145.2366148

Ali Mahdavi-Amiri, Fenggen Yu, Haisen Zhao, Adriana Schulz, and Hao Zhang. 2020. VDAC: Volume Decompose-and-Carve for Subtractive Manufacturing. *ACM Trans. Graph.* 39, 6, Article 203 (Nov. 2020), 15 pages. https://doi.org/10.1145/3414685.3417772

Luigi Malomo, Nico Pietroni, Bernd Bickel, and Paolo Cignoni. 2016. FlexMolds: Automatic Design of Flexible Shells for Molding. *ACM Trans. Graph.* 35, 6, Article 223 (Nov. 2016), 12 pages. https://doi.org/10.1145/2980179.2982397

Alessandro Muntoni, Marco Livesu, Riccardo Scateni, Alla Sheffer, and Daniele Panozzo. 2018. Axis-Aligned Height-Field Block Decomposition of 3D Shapes. *ACM Trans. Graph.* 37, 5, Article 169 (Oct. 2018), 15 pages. https://doi.org/10.1145/3204458

Kazutaka Nakashima, Thomas Auzinger, Emmanuel Iarussi, Ran Zhang, Takeo Igarashi, and Bernd Bickel. 2018. CoreCavity: Interactive Shell Decomposition for Fabrication with Two-Piece Rigid Molds. *ACM Trans. Graph.* 37, 4, Article 135 (July 2018), 13 pages. https://doi.org/10.1145/3197517.3201341

S. Nuvoli, A. Tola, A. Muntoni, N. Pietroni, E. Gobbetti, and R. Scateni. 2021. Automatic Surface Segmentation for Seamless Fabrication Using 4-axis Milling Machines. *Computer Graphics Forum* 40, 2 (2021), 191–203. https://doi.org/10.1111/cgf.142625 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.142625

Nico Pietroni, Enrico Puppo, Giorgio Marcias, Roberto Roberto, and Paolo Cignoni. 2016. Tracing Field-Coherent Quad Layouts. *Comput. Graph. Forum* 35, 7 (Oct. 2016), 485–496.

Oded Stein, Alec Jacobson, and Eitan Grinspun. 2019. Interactive design of castable shapes using two-piece rigid molds. *Computers & Graphics* 80 (2019), 51 – 62. https://doi.org/10.1016/j.cag.2019.03.001

Ingo Wald, Sven Woop, Carsten Benthin, Gregory S. Johnson, and Manfred Ernst. 2014. Embree: A Kernel Framework for Efficient CPU Ray Tracing. *ACM Trans. Graph.* 33, 4, Article 143 (July 2014), 8 pages. https://doi.org/10.1145/2601097.2601199

Ziqi Wang, Peng Song, and Mark Pauly. 2021. State of the Art on Computational Design of Assemblies with Rigid Parts. *Computer Graphics Forum* 40, 2 (2021), 633–657. https://doi.org/10.1111/cgf.142660 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.142660

Jinfan Yang, Chrystiano Araujo, Nicholas Vining, Zachary Ferguson, Enrique Rosales, Daniele Panozzo, Sylvain Lefevbre, Paolo Cignoni, and Alla Sheffer. 2020. DHFSlicer: Double Height-Field Slicing for Milling Fixed-Height Materials. *ACM Trans. Graph.* 39, 6, Article 205 (Nov. 2020), 17 pages. https://doi.org/10.1145/3414685.3417810

Chunjie Zhang, Xionghui Zhou, and Congxin Li. 2010. Feature extraction from freeform molded parts for moldability analysis. *The International Journal of Advanced Manufacturing Technology* 48, 1 (01 Apr 2010), 273–282. https://doi.org/10.1007/s00170-009-2273-7

Haisen Zhao, Hao Zhang, Shiqing Xin, Yuanmin Deng, Changhe Tu, Wenping Wang, Daniel Cohen-Or, and Baoquan Chen. 2018. DSCarver: Decompose-and-Spiral-Carve for Subtractive Manufacturing. *ACM Trans. Graph.* 37, 4, Article 137 (July 2018), 14 pages. https://doi.org/10.1145/3197517.3201338

Qingnan Zhou, Eitan Grinspun, Denis Zorin, and Alec Jacobson. 2016. Mesh Arrangements for Solid Geometry. *ACM Trans. Graph.* 35, 4, Article 39 (July 2016), 15 pages. https://doi.org/10.1145/2897824.2925901