VDAC: Volume Decompose-and-Carve for Subtractive Manufacturing

ALI MAHDAVI-AMIRI, Simon Fraser University
FENGGEN YU, Simon Fraser University
HAISEN ZHAO*, University of Washington, Shandong University
ADRIANA SCHULZ, University of Washington
HAO ZHANG, Simon Fraser University

Fig. 1. Carvable volume decomposition computed by our algorithm for the high-genus Fertility model, with 6 carving directions (indicated by the yellow arrows) and a total of 10 carvable volumes (one carving direction may yield multiple volumes, e.g., 3 volumes for the second direction). Three insets show physical outputs produced by CNC rough machining. Each carvable volume is continuously carved following a connected Fermat spiral toolpath.

We introduce *carvable volume decomposition* for efficient 3-axis CNC machining of 3D freeform objects, where our goal is to develop a fully automatic method to jointly optimize setup and path planning. We formulate our joint optimization as a volume decomposition problem which prioritizes minimizing the number of setup directions while striving for a minimum number of *continuously carvable* volumes, where a 3D volume is continuously carvable, or simply carvable, if it can be carved with the machine cutter traversing a *single continuous* path. Geometrically, carvability combines visibility and monotonicity and presents a new shape property which had not been studied before. Given a target 3D shape and the initial material block, our algorithm first finds the minimum number of carving directions by solving a *set cover* problem. Specifically, we analyze cutter accessibility and select the carving directions based on an assessment of how likely they would lead to a small carvable volume decomposition. Next, to obtain a minimum decomposition based on the selected carving directions efficiently, we narrow down the solution search by focusing on a special kind of points in the residual volume, *single access* or SA points, which are points that can be accessed from *one and only one* of the selected carving directions. Candidate carvable volumes are *grown* starting from the SA points. Finally, we devise an energy term to evaluate the carvable volumes and their combinations, leading to the final decomposition. We demonstrate the performance of our decomposition algorithm on a variety of 2D and 3D examples and evaluate it against the ground truth, where possible, and solutions provided by human experts. Physically machined models are produced where each carvable volume is continuously carved following a connected Fermat spiral toolpath.

*Corresponding author: Haisen Zhao (haisen@cs.washington.edu)

Authors' addresses: Ali Mahdavi-Amiri, Simon Fraser University; Fenggen Yu, Simon Fraser University; Haisen Zhao, University of Washington, Shandong University; Adriana Schulz, University of Washington; Hao Zhang, Simon Fraser University.

0730-0301/2020/12-ART203 $15.00
https://doi.org/10.1145/3414685.3417772

CCS Concepts: • **Computing methodologies → Computer-aided manufacturing**; **Shape analysis**.

Additional Key Words and Phrases: subtractive manufacturing, rough-stage CNC machining, volume decomposition, setup planning, toolpath planning

ACM Reference Format:
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 (December 2020), 15 pages. https://doi.org/10.1145/3414685.3417772

1 INTRODUCTION

As computer graphics research continues to push the envelope in additive manufacturing, in particular, 3D printing, some recent works have studied compelling geometry problems related to *subtractive* manufacturing such as CNC machining [Hattab and Taubin 2019; Muntoni et al. 2018; Rivers et al. 2012; Zhao et al. 2018]. Today, CNC machining remains the dominant manufacturing process for high-volume production of finished 3D products. Compared to 3D printing, CNC machining is typically faster and capable of producing larger and stronger parts, with smoother surface finishes. However, subtractive manufacturing has a complicated setup and planning process, which is typically done manually by experts [Al-wswasi et al. 2018]. Automation of this step is fundamental to inspire a new age of batch-of-one production of customized designs.

In this paper, we study the problem of *rough-stage*[1] 3-axis CNC machining of *freeform 3D objects*. As the most commonly applied subtractive manufacturing technology today, a 3-axis CNC machine moves with three degrees of freedom: two horizontal axes and the depth axis aligned with the orientation of the machine's cutter. We address the geometry problem of controlling the movement of the cutter around a block of material (typically a cuboid to start with) to carve out the *residual* volume, defined as the difference

[1]In this stage, a rough 3D object is produced by the CNC machine; this is followed by fine-stage machining to finalize the subtractive manufacturing.

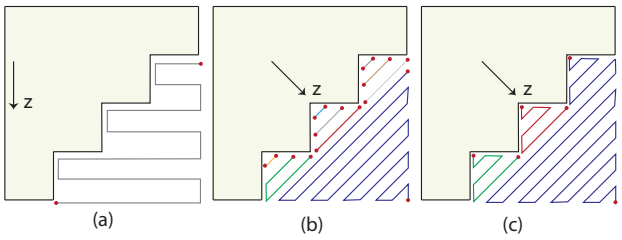


Fig. 2. The same shape with different carving directions (dark arrow marked by z) and carving paths: (a) a continuous zigzag; (b) zigzag with many path transitions (red dots); (c) a path with less number of transitions.

between the input material block and the target 3D shape. Due to self-obstructions by the shape parts (e.g., see the high-genus Fertility model in Figure 1) and physical limitations of the CNC machine, e.g., cutter length, it is often not possible to complete the carving of an arbitrary shape from a single direction. *Setup planning* is the task where a set of orientations for setting up the block of material is defined. This task is followed by *toolpath planning*, which optimizes the cutter movement paths for efficient machining.

Finding the minimum number of setup directions is typically the most important consideration, given the time and precision the manual setup would require. However, there are often many solutions with the same number of setups, with some solutions significantly better than the rest, if we take into account *how* the residual volume is carved. To illustrate, consider the example in Figure 2. All three paths shown can carve the full residual volume with a single setup. But solution (a), with its setup direction z, induces a *continuous* motion which would minimize the overall carving time. On the other hand, the other setup direction in (b) and (c) does not allow a continuous path; the cutter must be repositioned multiple times (red points) during machining. Among the two remaining solutions, a straightforward zigzag path (b) would incur many cutter transitions (i.e., stop-n-go's) than a more sophisticated path (c).

Hence, there is an inherent *interdependence* between setup and path planning, leading to a challenging computational problem which is compounded by the geometric complexity of the target 3D object.

The goal of our work is to *jointly* optimize setup and path planning by focusing on minimizing both the number of setup directions and the number of carving path transitions/repositions, while giving priority to the former. To formulate the problem, we make two key observations both illustrated by Figure 2. First, the total number of path transitions is precisely the number of regions of the residual volume each of which can be carved with one continuous motion, e.g., 1 in Figure 2(a) and 3 in 2(c). In other words, these paths induce a *decomposition* of the residual volume. Second, continuity of the carving depends on the choice of toolpaths; see Figure 2(b) vs. (c). It is known that any connected 2D region can be traversed continuously using a connected Fermat spiral [Zhao et al. 2016], but not zigzag. These observations allow us to formulate our joint optimization as a single *decomposition* problem of the residual volume, which prioritizes minimizing the number of setup directions while striving for a minimum number of *continuously carvable*, or simply

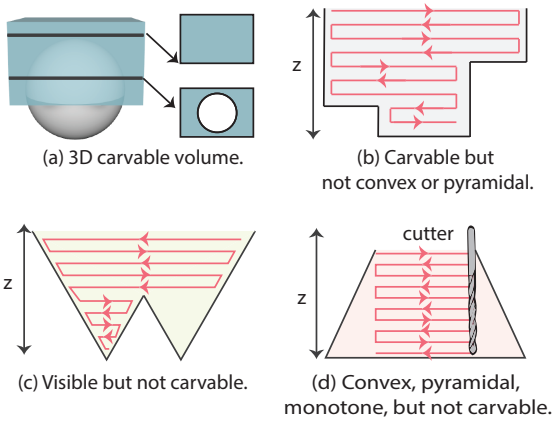


Fig. 3. Illustration of carvability, convexity, visibility, pyramidality, monotonicity, and a comparison between them. (a) A 3D carvable volume with two layers shown. (b-c) There is no causal relation between carvability and convexity, or pyramidality. While visibility (c) and monotonicity (d) are both necessary for carvability, neither is sufficient. We also show one *single path* of top-down, layer-by-layer carving of the shapes (b-d).

carvable, volumes. We refer to this problem as *carvable volume decomposition*. Clearly, all volumes that are carvable with respect of a setup direction should be carved sequentially to avoid switching setups. Optionally, we can also account for *cutter length*, i.e., the length of the CNC cutter that carves the residual volume, where shorter cutters are preferred to minimize vibration during carving and layers with larger areas induce less bending along the cutter paths, preventing slowdowns. Both of these considerations motivate minimizing the depth of carvable volumes.

Motivated by the continuous carving capabilities of connected Fermat spirals, we define a volume to be carvable, with respect to a depth/vertical direction z, if it is entirely *visible* from the direction z, and every horizontal cross-section, called a *layer*, of the volume is a *connected* region (possibly with holes); see Figures 3(a) and (b). While carvability is related to convexity, visibility, pyramidality [Hu et al. 2014] (i.e., the volume is a height field over a flat base), and monotonicity[2], it is a *new and different* geometric property from all of them, as illustrated in Figures 3(b-d). Specifically, our carvability definition is equivalent to requiring both visibility and monotonicity, while allowing the cross-sections to have holes.

To the best of our knowledge, carvable volume decomposition has never been studied before. Providing an NP-hardness proof for the problem is beyond the scope of our work, but it is quite conceivable that the problem is hard, given that related decomposition problems including exact convex decomposition, monotone polygon decomposition for polygons with holes [Keil 1985], as well as exact 3D pyramidal decomposition [Fekete and Mitchell 2001] are all provably NP-hard. Hence, as a first attempt, we present a novel, heuristic solution, which is outlined in Figure 4.

[2]A 2D polygon P is *monotone* with respect to a straight line L, if every line orthogonal to L intersects P at most twice. A 3D volume is *weakly monotonic* in direction L if all cross-sections orthogonal to L are simple polygons [Toussaint 1985].

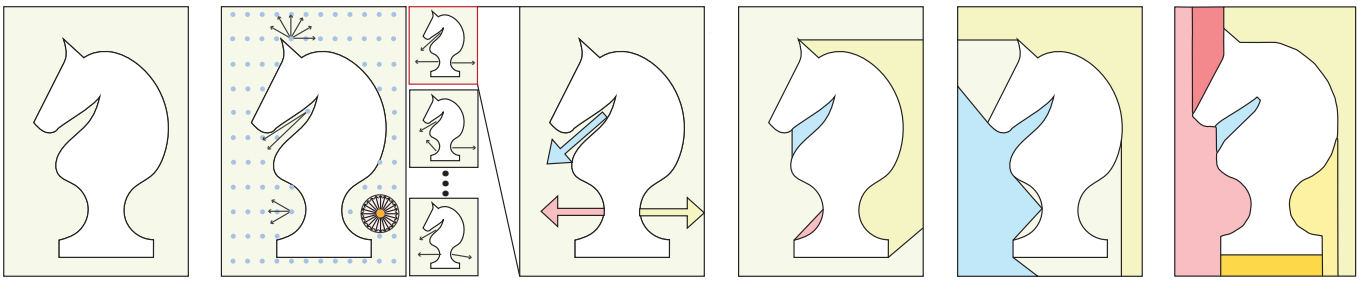


Fig. 4. Overview of our carvable volume decomposition algorithm. Given a target object and the material block (a), our first task is to find a minimum number of carving directions along which the entire residual volume can be accessed by the cutter. To this end, we sample the residual volume spatially and direction-wise (b), perform visibility and accessibility analysis to find a *set* of candidate carving directions with minimum cardinality, which is three in this example, and then select the best candidate (c) from the set. Next, we find a minimal carvable volume decomposition based on (c). Specifically, we identify SA (single access) sets (d) with respect to the selected carving directions, where an SA set consists of all points that can be accessed by one and only one carving direction. Candidate carvable volumes (e) are obtained by a growing process starting from the SA sets. Finally, we devise an energy term to evaluate the carvable volumes and their combinations; results with the lowest energies constitute the final decomposition (f).

Our method works by first finding the minimum number of setup, i.e., carving directions, by solving a *set cover problem*. Specifically, we analyze visibility and accessibility, based on cutter length, and select the carving directions based on an assessment of how likely they would lead to a small carvable volume decomposition; see Figure 4(b-c). In the next and core step of the algorithm, we obtain a minimum decomposition of the residual volume, based on the selected carving directions. Clearly, we prioritize the minimization of the number of setup directions by solving that problem first, while our selection of the *best* candidate does strive for finding a minimum carvable volume decomposition.

Due to the immense solution space, our main strategy is to narrow down the search by first identifying points that can only be carved from one of the selected directions. Clearly, to compute a carvable volume for these points, we only need to consider that one carving direction. Indeed, *our second key observation* is that while most points in the residual volume can be accessed by the cutter from multiple carving directions, there are "special points" that can be accessed from *one and only one* of the directions, as shown in Figure 4(d). We call such a set of points a *single access set*, or SA set, and prove that SA sets must exist for each carving direction. Given the unique association of an SA point with a carving direction, a sensible heuristic for finding a minimum volume decomposition would be to start from a maximal SA set, and grow an as-large-as-possible carvable volume with respect to its associated carving direction. However, due to the extra monotonicity requirement for carvability, we often need to find multiple carvable volumes to cover all SA points for a given direction. To this end, we develop a scheme to find a minimum number of *SA seeds* for each SA set, from which we grow a set of carvable volumes to constitute the final decomposition of the entire residual volume; see Figure 4(e-f).

We demonstrate the performance of our carvable volume decomposition algorithm on a variety of 2D and 3D examples and evaluate it against ground truth results, where possible, and solutions provided by human experts. Our algorithm is quite fast, requiring few minutes to run on moderately complex 3D models. This can significantly reduce human workload during setup planning. Finally, real CNC machining results based on our decomposition and Fermat spiral toolpaths [Zhao et al. 2018] are obtained; see Figure 1.

2 RELATED WORK

In the computer-aided design (CAD), computer-aided manufacturing (CAM), and mechanical engineering domains, there has been extensive literature on CNC machining, but mainly toolpath planning [Choi and Jerrard 1998; Lasemi et al. 2010]. Computer-assisted process (i.e., setup) planning (CAPP) has been a long-standing topic in these areas. However, "due to the enormous complexity of the nature, and the dynamic aspects of the process planning task", it is still predominantly a manual process, relying on expert knowledge and experience of the machinists [Al-wswasi et al. 2018]. We are not aware of any work from the CAD/CAM/manufacturing literature which addressed the volume decomposition problem as posed in our paper. On the other hand, surface and volume decomposition has been a well-studied problem in computer graphics with numerous applications. Our coverage will focus on those most closely related to our work, e.g., height-field surface decomposition. But first, we start with some necessary CNC machining basics.

Rough- vs. fine-stage machining. Our work belongs to *rough-stage* machining where the carving starts from the initial material block and stops "close to" the target object. This is followed by fine-stage machining for final surface finish. Due to accessibility issues, both stages would involve decomposition, with the rough stage operating on the residual *volume*, leading to our volume decomposition problem. In contrast, the fine stage operates close to the target object's surfaces — the ensuing accessibility analysis poses a *surface* decomposition problem, which has been studied by Zhao et al. [2018] recently. There are other differences between the two stages, e.g., the shape and size of the cutters and the fixtures employed for stabilization. Our work does not fully account for all these issues and focuses on the geometric problem of carvable volume decomposition. Rough machining typically takes as input an allowed error threshold that

is determined by the machinists and depends on the properties of the cutters chosen for both stages. Taking this into account when computing the path for rough machining is important to optimize the fabrication time. Not accounting for this allowance would make the rough machining state slower simply to avoid impressions that will be taken care of in a later stage regardless.

Height-field surface decomposition. Decomposing a surface into height-field pieces has seen early applications such as printing cultural art works [Alemanno et al. 2014]. For molding applications, Herholz et al. [2015] studied the problem of decomposing the surface of a 3D object into a set of non-overlappling height-fields. Clearly, each height-field is visible from the outside of the object along the height direction, allowing a mold to cover that surface area. In DSCarver, the work by Zhao et al. [2018] for fine-stage CNC machining, the surface decomposition method of Herholz et al. [2015] was employed as a sub-routine. In FlexMolds, Malomo et al. [2016] develop a greedy heuristic to segment a surface where each segment is almost a height-field so as to allow a flexible mold to be applied. Similarly, Alderighi et al. [2018] use a relaxed visibility constraint to define moldability for making silicone molds.

The key distinction between moldability/visibility and carvability, assuming that the molds are rigid (i.e., not deformable [Malomo et al. 2016]), is that the latter also requires monotonicity; see Figure 3. Figure 2 shows that different visibility-inducing directions, which all represent acceptable solutions to height field surface decomposition, could lead to significant discrepancies in carving efficiency. Furthermore, the decomposition of Herholz et al. [2015] is limited to genus-zero shapes; it cannot handle models such as the Fertility.

Volume decomposition. Volume decomposition has been studied well for computational fabrication and 3D printing [Alderighi et al. 2019; Livesu et al. 2017; Sá et al. 2016; Stein et al. 2019]. Chopper [Luo et al. 2012] decomposes a 3D shape into parts each of which can fit into the printing volume of a 3D printer. Dapper [Chen et al. 2015] goes one step further and optimizes packing of the parts in a compact manner to best utilize the printing volume. Jacobson [2017] solves a nested packing problem to replicate a Generalized Matryoshka. Hu et al. [2014] decompose a 3D object into a few approximately pyramidal parts so that each part can be fabricated using an FDM printer with minimal support waste. While pyramidality and carvability both possess the height-field property, one over a flat base and the other over an arbitrary base, not all pyramidal shapes are carvable; see Figure 3. Monotonicity is a necessary condition for carvability. We are only aware of works from computational geometry which decompose a *planar* polygon into monotone pieces, e.g., [Keil 1985; Wei et al. 2012]. In the 3D case, one could only find the work by Bose and van Kreveld [2005] which studies the *recognition* of weak monotonicity of polyhedra.

Muntoni et al. [2018] seek a volume decomposition of a 3D object into height-field (i.e., pyramidal) blocks. Their method starts with a height-field *surface* decomposition and "intrudes" from these surface segments into the object, along the height direction, to obtain the blocks. While pyramidality ensures that the block can be easily manufactured via 3-axis CNC milling, the machining time is not explicitly optimized for. In contrast, a part obtained by our method is from the residual volume — it is *carved off* by a CNC machine;

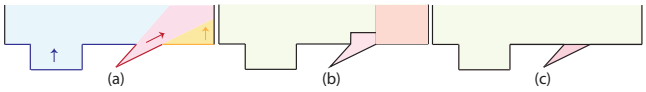


Fig. 5. Extrusion from surface decomposition vs. volume decomposition, illustrated in 2D. An optimal height-field surface decomposition has three segments, which can be extruded to obtain three carvable volumes (a). Surface2Volume decomposition by extruding the surface segments into assemblable volumes generates three carvable volumes (b). The optimal *volume* decomposition has only two carvable volumes (c).

and we do optimize for the carving time by insisting that the part is carved via a single continuous path. Last but not least, the height directions in their work are restricted to be aligned with one of the canonical x, y, and z axes, while in our work, we sample over many directions. Both methods are tied to height-field decomposition, but the decomposition results and associated height directions are completely different. For reference, on the high-genus Fertility and Kitten models, their results contain 63 and 25 blocks, respectively, while our method produces 10 and 5 parts, respectively.

In Surface2Volume, Araujo et. al. [2019] compute a volume decomposition based on a *given* surface segmentation, which may or may not be a height-field decomposition. Similar to [Muntoni et al. 2018], their method works "inward" from the surface segments to produce a set of *assemblable* parts. These parts conform to the surface segments and can be moved apart with no collisions. In general, assemblability has nothing to do with how a part may be carved by a CNC machine, except that geometrically, carvability implies visibility, which in turn, implies assemblability.

From surface to volume decomposition. One possible way to obtain a carvable decomposition of the residual volume is to start with a height-field surface decomposition of the target 3D object and *extrude outward* from the surface segments, along their corresponding height directions. Clearly, we cannot rely on the height-field decompositions obtained by Muntoni et al. [2018], since, as we discussed above, their decompositions can be far from optimal (e.g., for high-genus models such as the Fertility). A better choice is to use an optimal height-field decomposition, such as one obtained by Herholz et al. [2015]. But the first problem is that not all height directions would yield an efficient carvable volume decomposition, as shown in Figures 2(b-c). In contrast, the example in Figures 5(a-b) provides a best-case scenario, where each surface segment can be extruded into a carvable volume. Yet, the final volume decomposition is still sub-optimal, as illuminated by the optimal solution shown in Figure 5(c). For the extrusion approach to work, each volume of an optimal solution must completely cover a continuous area of the target surface. Unfortunately, this is not necessarily the case for a decomposition of the residual *volume*. Moreover, consideration of the depth of the carvable volumes necessitates an analysis of the residual volume, not just the surface of the target 3D object.

Toolpath planning. In 3-axis CNC machining, a carvable volume is typically sliced into a set of planar layers, each carved by the cutter traveling a specific path. There are many possible toolpath patterns including zig-zag [Ding et al. 2014], iso-contours [Yang et al. 2002], and double spirals [Hauth and Linsen 2011]. In our work, we adopt the connected Fermat spirals developed by Zhao et al. [2018] to

execute a top-down, layer-by-layer carving of the carvable volumes obtained by our decomposition method. When choosing a proper toolpath pattern, there are several criteria to take into consideration such as smoothness of the paths which would affect the speed of machining, along with retraction, cutter lifting, and deacceleration that should be avoided as much as possible. Fermat spirals appear to offer various advantages on these accounts and most importantly, it allows traversal of complicated, but connected areas using a single continuous space-filling path.

3 OVERVIEW AND PROBLEM FORMULATION

In this section we will motivate our approach and describe the key steps at a high level. Throughout the paper, we refer to regions that we grow as *volumes* (Figure 4(e)) and non-overlapping regions (Figure 4(f)) that cover the entire residual volume as *segments*.

3.1 Design Considerations

Our algorithm is based on the following considerations that machinists take into account for CNC machining. This list was developed based on interviews with four expert machinists with 10 to 27 years of professional experience.

- **Setup cost:** Changing the carving direction involves repositioning the part, which is time-consuming and error-prone. Therefore, we should minimize the number of setups.
- **Carving time:** Experts usually prefer long, continuous paths to carve as much material as possible in one pass.
- **Cutter length:** Shorter cutters are preferred during rough machining to minimize vibrations and increase cut accuracy.

Consider a residual volume V that is the subtraction of a target shape A from the initial block B. Our method proposes a plan for carving out V driven by the criteria above. We will do this by decomposing V into a set of *carvable segments*, which, as previously stated, are regions that can be continuously carved from a given direction. We will use Fermat spirals [Zhao et al. 2016] to define cut paths for each of the segments because they provide a continuous path, which will help to minimize the total cut time.

Since setup cost is typically the most crucial, we must minimize the total number of *carving directions* used to carve all the segments. We ensure to minimize the number of setups by solving a set cover on the number of machining directions.

In addition, we must ensure that all segments associated with a given direction can be carved sequentially. Because each segment can be cut continuously, we minimize the cut time by simply minimizing the number of times the cutter needs to travel between segments (i.e., we minimize the number of carvable segments).

The cutter length that is necessary to carve out a segment depends on a complete analysis of collisions, which depends on the shape of the cutter head and the shape and order of segments being carved. While this evaluation can be costly, we provide a simple solution by analyzing accessible points through a given direction under a certain cutter length. In addition, we efficiently prioritize segments that can be carved under fewer number of vertical movements by removing as much material as we can in one path as it creates less blockage for the cutter. This can also have an added benefit for cut time, as for some CNC machining processes, moving along vertical directions is time consuming and inefficient.

3.2 Algorithm Overview

We wish to find a set of non-overlapping segments $\mathcal{S} = \{s_1, ..., s_n\}$ that cover the residual volume V and are carvable using a set of carving directions $\mathcal{D}$. This is not an easy task because there can be infinitely many carving direction sets that have the same cardinality—consider for example, the problem of carving out a perfect sphere. Furthermore, under a given direction set, there are potentially intractable configurations of carvable segments covering the residual volume.

As previously discussed, our key observation is that if $\mathcal{D}$ is a *minimal* set of carving directions, then it will define SA sets. These SA sets can then be used to derive a decomposition algorithm for V. This will be done by the following steps:

Accessibility Analysis: Our first step is to sample both the residual volume and the unit hyper-sphere to generate a discrete set of carving directions. This sampling can be uniform or adaptive to respect shapes' features. Then, we find a set of directions with minimum cardinality through which all points are accessible- i.e., visible along the carving direction and reachable by the cutter considering its length. The direction set should attain minimum cardinality to ensure that we minimize the number of setups.

Carving Direction Selection: After finding a set of machining directions $\mathcal{D}$ that can be used to fully carve the residual volume, we argue why this solution is not unique. We therefore select one of such direction sets by assessing the quality of segments it will produce based on the criteria described above.

Generation of Candidate Carvable Volumes: The next step is to find a set of volumes that can be carved using the selected directions. These are created by analyzing the SA sets and defined such that the union of all candidates span the entire residual volume with potential overlaps.

Carvable Volume Decomposition: To produce the final non-overlapping carvable volume segments, we then find the order of combining the candidate carvable volumes to optimize for the criteria discussed above by minimizing an energy function.

Cutter Path Planning: Finally, we use Fermat spirals to trace a full continuous path to carve each segment.

4 METHODOLOGY

We now describe each step of our algorithm in more detail. In this section, our explanations are for 2D shapes and later in Section 5, we discuss how we can extend it to 3D shapes.

4.1 Accessibility Analysis

First, all input shapes are normalized in scale by setting the longest side of the tightest bounding box of the shape to unit length. We then need to find a set of carvable directions with which all the points in the residual volume are visible. To find these directions, we sample the residual volume with a set of points p_j and then uniformly discretize the hyper-sphere with a set of vectors d_i (see Figure 4(a)).

Each point p_j needs to be reached by the cutter through a direction d_i. Assuming an infinitely long cutter with infinitely small width, a volumetric segment, $s_k \in V$ is carvable in direction d_i if all points $p_j \in s_k$ are visible along d_i and at each layer sampled along d_i, the volume is connected (possibly with holes for 3D shapes; see Figure 3(a)). A point p_j is visible through direction d_i, if when we shoot a ray from p_j along d_i, it has no intersection with the shape. Users can specify cutter length constraint to avoid using long cutters that may produce shaking and artifacts. Here, we also discard points that are not *accessible* by the cutter in a specific direction due to cutter length constraint. A detailed discussion on how we incorporate this constraint is provided in Section 6.

Since each direction is equivalent to an expensive setup, we wish to carve all the points p_j with minimum number of directions. Identifying the minimum number of directions d_i that cover all the points p_j is equivalent to solving a set cover problem for directions d_i. This way, we define a set cover $\mathcal{D}_s = \{d_i, i = 0, \ldots, N\}$ as a collection with the fewest number of directions through which all sampled points are accessible. Since there can be multiple direction sets with the same cardinality we index set covers by $\mathcal{D}_s$.

4.2 Carving Direction Assessment

While solving set cover gives us all the sets, $\mathcal{D}_s$, that result in the minimal number of setups, the fabrication cost will still depend on the number of carvable segments, which can be vastly different even in direction sets with the same cardinality (see Figure 2). The next step is then to choose the direction set that allows the lowest fabrication cost (Figure 4(c)). However, this is an intractable process since it would involve finding the optimal decomposition for each direction set. We therefore propose a method to estimate the fabrication cost allowing us to efficiently assess all directions and choose the best one, $\mathcal{D}_*$. Since our approximation discussion is better understood after the presentation of our decomposition algorithm, we leave its discussion for Section 4.5 and we close this section with an important observation about direction sets.

Proposition 1. *For each direction d_i in $\mathcal{D}_s$, there is at least one sample p_j that is only accessible by d_i and no other direction.*

Proof. We can prove this by contradiction. Let $\{p_j^i, j = 0, \ldots, M\}$ be the M samples that are accessible through $d_i \in \mathcal{D}_s$. Assume that each point is accessible through another direction in $\mathcal{D}_s - \{d_i\}$. If this is the case, then there exists a smaller set of directions that can cover the full samples, namely $\mathcal{D}_s - \{d_i\}$, and therefore, $\mathcal{D}_s$ is not a set cover by definition. Sample points that are accessible through only one vector are called *Single Access* points or SA points.

4.3 Generation of Candidate Carvable Volumes

Given point p_j and direction d_i, we define the *maximal carvable volume* $M(p_j, d_i)$ as the largest volume that can be carved along direction d_i that will include p_j (see Figure 4(e)). As previously discussed, our algorithm is based on generating a set of candidate carvable volumes that will fully span the residual volume and we will then use these candidates to partition the space into a set of non-overlapping segments (Figure 4(f)). The motivation of using maximal volumes for candidates is that carving more from residual volume in one path creates less blockage for moving the cutter; plus

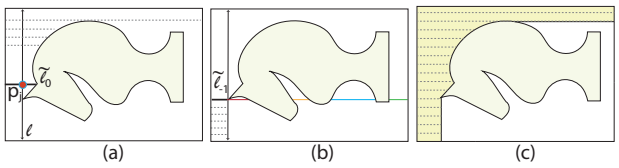


Fig. 6. Starting from a point p_j along line ℓ that is a line-segment parallel to d_i, carvable regions are found by finding the closest intersection of $\tilde{\ell}_t$ with shape or block (a). Moving along ℓ in the opposite direction, we find a set of line-segments by finding intersection of $\tilde{\ell}_t$ with the shape and block and pick the mergable line-segment highlighted in solid black (b). A maximal carvable volume $M(p_j, d_i)$ is found using the end points of dashed lines (c).

candidates are independently carvable meaning that the carvability of candidates does not depend on each other giving us the freedom of choosing any order of volumes to carve. As it would be intractable to consider all maximal carvable volumes, we propose a method for selecting candidate volumes that will produce a final decomposition matching our discussed criteria. In the following, we first discuss how to generate a maximal carvable volume from a point—direction pair and then elaborate how to generate a set of candidates.

Maximal Carvable Volumes. A maximal carvable volume $M(p_j, d_i)$ is generated by growing a *monotone* region around p_j by simulating the CNC machining process that removes layers perpendicular to the machining direction d_i. This is done by moving vertically along machining direction d_i traversing perpendicular to d_i at sample depths to carve out layers of the residual volume. Line ℓ that passes through p_j along d_i is parameterized by $p_t = p_j + t\delta d_i$ (t can be also negative) to sweep line $\tilde{\ell}_t$ perpendicular to ℓ (See Figure 6). Traversing ℓ resembles the vertical movement of the cutter while $\tilde{\ell}_t$ determines side-by-side movements of the cutter at p_t. The line-segments extracted from $\tilde{\ell}_t$ determine movement range of the cutter at different layers and also the boundaries of carvable volumes. Note that the resulting volumes are always monotone.

To compute $\tilde{\ell}_t$, we first find the intersections of the line passing through p_t and perpendicular to d_i with shape A and block B, which will determine all the line-segments that are inside the residual volume V. If t is positive we can simply select the line-segment that contains p_t, because there is no blockage to p_t (Figure 6(a)). For negative t, we choose a line-segment that can be merged with the region grown with positive ts (Figure 6(b)). Two consecutive line-segments can be merged if the projection of one onto the other has an overlap. We stop when there is no mergable line-segment found or $\tilde{\ell}_t$ has no intersection with the residual volume (i.e., it is out of the block). Consecutive endpoints of these line-segments form the boundary of a carvable volume (Figure 6(c)).

Selecting Candidate Seeds. Carefully choosing the point—direction pairs to grow maximal carvable volumes $M(p_j, d_i)$ is important to generate an appropriate decomposition. We will take advantage of SA sets to create candidate seeds. Our method for selecting candidates has two steps. First, we select seeds from the set of SA points, as defined by Proposition 1. These are called *SA seeds* from which we grow maximal carvable volumes to cover the residual volume. Second, as it is possible that the regions grown from SA seeds do

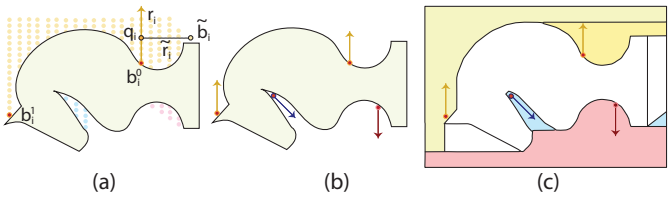


Fig. 7. SA points of the same directions are in the same solid color. Seeds are the bottom points along each direction that are depicted by points filled by red (e.g, ●). Ray r_i is shot from seed b_i. An SA point $\tilde{b}_i$ (○) is mergable with seed b_i^0, if the line-segment between q_i (○) and $\tilde{b}_i$ has no intersection with the object. q_i is the intersection of ray $\tilde{r}_i$, shot perpendicular to r_i, with r_i (a). SA seeds and their directions (b). Candidate carvable volumes grown from the SA seeds (c). Pink has been depicted over the blue volume.

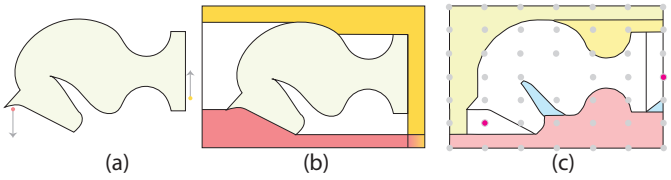


Fig. 8. Complementary seeds are selected (a) to cover the entire residual volume (b). To estimate the number of complementary volumes, the residual volume is coarsely sampled and the points that are not accessible by SA seeds are chosen to form complementary volumes in the same fashion (c).

not cover the entire residual volume (see Figure 7(c)), we select a set of *complementary seeds* to cover the rest of the space.

For each direction d_i, we define SA regions as a set of SA points $\mathcal{B}_i$, which is always non-empty and typically includes many points (see Figure 7(a)). Our goal is to select the smallest number of seeds that when grown under direction d_i will cover all of these points. Multiple seeds will be necessary when not all points can be carved by a single carvable volume. This happens when the SA region associated with $\mathcal{B}_i$ is not monotone. Therefore, we break it into monotone regions. To do this, we start by selecting the first seed as the bottom point $b_i^0 \in \mathcal{B}_i$ along direction d_i. If we shoot a ray from b_i^0 along d_i, all the points in $\mathcal{B}_i$ that are accessible through this line can be carved from the volume grown from b_i^0 and we therefore add them to C along with b_i^0; C is now a monotone region. We select the next seed by repeating this process for $\mathcal{B}_i - C$ until this set is empty, defining the set of SA seeds for direction d_i, $\mathcal{I}_i = \{b_i^0, b_i^1, \ldots\}$.

Next, we propose a method for selecting *complementary seeds* that, when combined with SA seeds, will cover the entire residual volume. This is done by first identifying initial sample points (p_j) that have not yet been covered by the grown regions and a similar process that is used for finding SA seeds. Since complementary seeds are from points that are reachable through multiple directions in $\mathcal{D}_*$, we choose a direction with lower energy (provides shallower carvable volumes; see Section 4.4 for our energy function). We repeat this process until the entire residual volume is covered by at least one carvable volume.

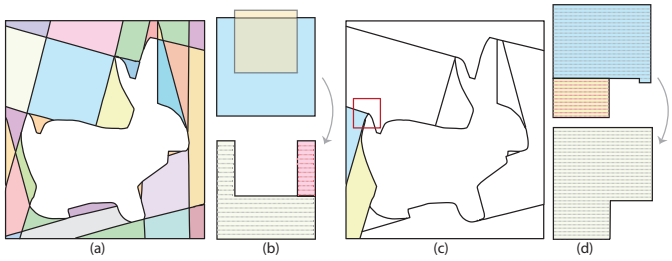


Fig. 9. Producing all possible sub-volumes results in many small volumes that may not be necessarily carvable (a). Subtracting two carvable volumes (b. top), is not carvable in a single path (b. bottom). Small details (highlighted in red box) produce an extra segment (c). In a post-processing phase, two consecutive segments carvable along the same direction (d. top) are combined in a single carvable volume by removing a small detail.

4.4 Carvable Volume Decomposition

The previous step generates a set of maximal carvable volumes $\mathcal{M} = \{M_1, M_2, ..., M_n\}$, which can be carved using directions in $\mathcal{D}_*$ and span the entire residual volume V. Our next step is to use these candidate volumes to create a non overlapping decomposition $\mathcal{S} = \{s_1, s_2, ..., s_n\}, s_d \cap s_e = \emptyset, \forall d \neq e$ to stop the cutter from going over an already carved region for multiple times. We will first define an energy function that allowing us to assess decomposition results and then describe a method for finding a decomposition that minimizes this energy.

Energy Function. As discussed earlier, we are looking for a small number of shallow segments. Therefore, if we have carvable segments $\mathcal{S} = \{s_1, ..., s_n\}$, our energy function is composed of $E_{sh}(s_k)$ that measures the shallowness of each segment s_k, along with an additional term $E_{\mathcal{S}} = |\mathcal{S}|$ that is the number of carvable segments. $E_{sh}(s_k)$ is defined as follows:

$$E_{sh} = \frac{h(s_k)}{A(s_k)} \tag{1}$$

where $h(s_k)$ and $A(s_k)$ are respectively the height and area (volume for 3D) of each segment. We normalize the heights to promote wider segments under the same height as carving such segments removes more blockage against the cutter. Plus, with the same depth, more material is carved in wider segments. The total energy of a decomposition, therefore, is defined as:

$$E_{total} = \alpha \sum_i E_{sh}(s_i) + \beta |S| \tag{2}$$

Smaller values for α promote fewer number of carvable segments while smaller β promotes shallower segments. See Section 6 for a more detailed discussion about the effects and settings of these parameters.

Decomposition. Having candidate carvable volumes $\mathcal{M}$, one may attempt to find all possible combinations including intersection, union, and subtraction to partition the residual volume into a set of non-intersecting carvable regions. Then by combining these carvable regions, one may find a carvable decomposition that covers the entire residual volume with no overlaps and minimizing the energy function. We have tried this approach and several problems

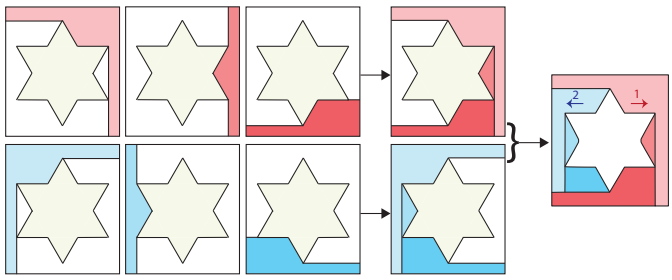


Fig. 10. Maximal carvable volumes of each direction are combined individually and the results are integrated to obtain the final decomposition.

may occur. First, even for relatively simple shapes, many small carvable regions might be produced (Figure 9(a)). Second, intersection, subtraction, or union of these carvable regions is not necessarily carvable in a single path (Figure 9(b)). As a result, further assessment of regions after such tasks is necessary. Third, finding the smallest set with no intersection covering the residual volume is equivalent to exact cover problem [Garey and Johnson 1983]. Known algorithms to solve this NP-complete problem such as Algorithm X [Knuth 2000], which has been previously used for segmentation [Hu et al. 2014], fails to deliver near optimal results for large sets in a reasonable time.

We therefore propose an alternative technique that tries to utilize the candidate carvable volumes as much as possible and find their best combination. In this method, two candidate carvable volumes M_f and M_e are combined in a given order. If M_f is supposed to be carved first, M_f is preserved in our decomposition and then $M_e - M_f$ is carved. Note that $M_e - M_f$ might not be carvable, in this case, we split it into carvable volumes. As a result, finding a decomposition amounts to selecting an *order* of candidate carvable volumes. Having n candidates carvable volumes, there exist $n!$ possible decompositions, which would be intractable to search. However, we have an additional constraint that results from our algorithm design criteria: volumes that are carvable in the same direction must be carved together to avoid unnecessary set-up changes. Therefore, we combine all the volumes of each direction d_i and then integrate the results to find the final decomposition $\mathcal{S}$ attaining the lowest cost with no overlap (see Figure 10). In addition, to shrink the search space even more, if s_d and s_e have no overlaps, their order does not matter, therefore, we remove such repetitions. When the number of volumes that are going to be combined is less than or equal to four which is the case for most of our results, finding all the possible combinations is fast. For combinations with larger number of volumes (e.g., ✲ in Figure 20), we use beam search with beam width equal to four [Lowerre 1976]. We prioritize the set of candidate volumes based on the energy function in Equation 1 and assessment of the final result is done according to Equation 2.

4.5 Estimation to Find D_*

Since growing candidate carvable volumes for all $\mathcal{D}_s$ to find $\mathcal{D}_*$ is inefficient, the number of SA and complementary seeds are considered as an estimation for the number of carvable segments. We then estimate the cost of each direction set according to our energy function in Equation 2 to find $\mathcal{D}_*$. Therefore, the number of SA seeds for each direction set $\mathcal{D}_s$ is first counted. Then the residual volume is evaluated to determine whether more seeds (i.e., carvable volumes) are needed to cover the residual volume. Here, to quickly evaluate the residual volume, we coarsely sample it with a few number of samples (e.g., a seven by seven grid) and estimate the number of complementary seeds. Complementary seeds are the points that do not belong to SA regions and from them all the other complementary points are accessible (see Figure 8).

4.6 Post-processing

Small-scale details sometimes may result in additional segments or directions (Figure 9(c)). In rough machining, however, it might be more desired to avoid more carvable segments, sacrificing tiny details. To avoid such issues, consider two neighbouring segments s_i and s_j that are carvable in the same direction and s_i is going to be carved first. We generate a single carvable segment s_k by removing the edge connecting s_i and s_j. If the remaining segment $(s_i - s_k)$ is too small, we remove it and keep s_k.

4.7 Cutter Path Planning

For a 2D case, the most suitable paths to fill carvable volumes are direction-parallel patterns in which the line-segments identified from each layer can be connected in a zig-zag manner. However, cutter paths for 3D are different as we employ Fermat spirals to generate smooth and continuous paths for each carvable region. In the next section, we discuss how to extend all the steps in our algorithm to 3D including cutter path planning.

5 EXTENSION TO 3D

Extending our algorithm from 2D to 3D is very straightforward as most of the steps in both cases are fairly similar. In the following, we discuss the extension of each step in detail.

Accessibility Analysis and Carving Direction Selection. Finding and assessing machining direction in 3D and 2D are similar with the only difference that the sample points are in a 3D residual block and vectors are sampled over a unit sphere. However, finding SA seeds and complementary seeds are exactly the same.

Carvable Segments. Although the basic idea of growing carvable segments in 2D and 3D is very similar. For 3D shapes, we use a discrete approach with the main reason being to simplify finding a continuous carvable segment. Similar to the 2D case, we first find SA seeds and sample on line ℓ along direction d_i and find the intersection of plane ρ with normal d_i with the object and bound it with the block. We uniformly sample ρ with points q_w, if q_w is at the boundary, in the object or in the residual volume it respectively receives tags two, one, or zero. We use points along line ℓ as a seed for the 2D region growing algorithm on samples on ρ and find a carvable connected 2D layer, possibly with holes, by merging all connected points with zero tags. Figure 11 illustrates this process. The stacks of these 2D layers form a 3D carvable segment. To generate the surface for each volume, we calculate signed distances for points near the shape and reconstruct a surface using marching cubes.

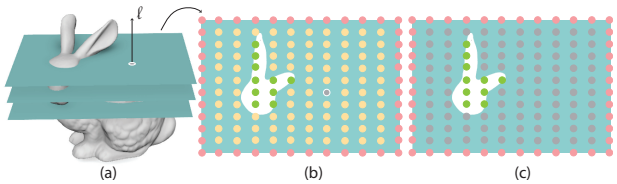


Fig. 11. Sampling along line ℓ, we find the intersection of plane ρ with the object (a). Point q_w at the boundary (●), in the object (●) or in the residual volume (●) respectively receives tag two, one, or zero (b). From a seed point (●), a 2D region growing is performed to find a carvable connected layer (c). Stacks of these layers form the candidate carvable volumes.

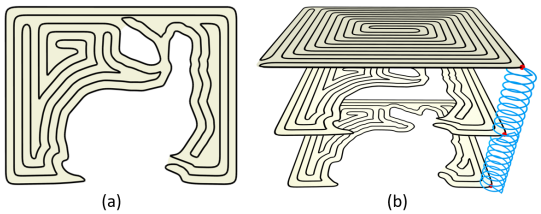


Fig. 12. Connected Fermat spirals for a single layer (a). Helix lines for connecting two neighboring layers (b).

Carvable Volume Segmentation. We use the same energy functions and optimization approach for finding the carvable volume segmentation and use libigl [Jacobson et al. 2018] to perform boolean operations to find the intersection between volumes. As discussed above, it is possible that the results of a subtraction operation, $M_e - M_d$, is not carvable in one continuous path and must be further segmented. To split a non-carvable segment into carvable segments, we first group the sample points of $M_e - M_d$ into different slicing layers with respect to the cutting direction d_e. A tree structure is built whose nodes are the connecting layers and whose edges denote the inclusion relationship between two layers. A decomposition into carvable segments can be driven by decomposing such tree structure into a minimal number of sub-trees.

Surface Dilation/Erosion due to Cutter Width. In practice, the CNC cutter has a physical width ω, which implies that it is unable to access regions of the target surface that are narrower than ω. In addition, the cutter length is measured up to the *center* of the cutter head, which is about $\omega/2$ away from the cutter tip. Due to these two reasons, the actual residual volume that is accessible by the cutter (up to the center of the cutter head) is not exactly the difference between the initial block and the target shape — there is an offset. We compute the accessible volume via mathematical morphology, where we dilate the target surface by $\omega/2$, which is equivalent to an erosion, with the same amount, of the difference volume between the block and the target shape. Hence, the *virtually carved* volume is a shrunk version of the actual residual volume. This explains why the virtual Fertility models shown in Figure 1 appear to be inflated compared to the target shapes. In contrast, the results from physical CNC machining, shown in the insets in Figure 1, appear "thinner", as they are closer approximations to the target shapes.

Tool Path Planning. For each slicing layer of a carvable segment, connected Fermat spirals (CFS) could be generated as a fully closed tool path (Figure 12 (a)). While any CFS point can be selected as the entry point at each layer, entry point selected with a random process could lead to a long cutter transition between each layer (Figure 12 (b)). In order to minimize the cutter transitions, we apply an iterative optimization method that first generates random entry points, and then updates their positions to minimize the length of the transitions between layers. With the determined entry points, we generate a set of helix connecting lines to connect two neighboring layers in order to reduce the cutting force when the cutter touches the material of the next layer (Figure 12 (b)).

6 RESULTS AND EVALUATION

In this section, we show a variety of results produced by our method, starting with a gallery of 2D and 3D decompositions in Figures 13 and 15, respectively, and an attempt to carve out a text SIGGRAPH logo, in Figure 14. We evaluate our method against baselines, the ground truth (if we could find it), and results provided by machining experts. But first, we describe several implementation details related to sampling, cutter length, and parameters. We have implemented all the steps of our algorithm in C#, where polyshape functionalities of Matlab have been used for 2D segmentation and libigl [Jacobson et al. 2018] is utilized for 3D segmentation.

For physical fabrication of the Kitten, Bunny, Fertility, Rocker-arm, Horse, and Unicorn models (see Figure 27), we account for the cutter width $\omega = 1.5875mm$ to compute the dilation and our algorithm operates on the resulting residual volumes. For 2D decomposition and results on other 3D models, e.g., the Max-Plank and Bimba in Figure 15, we do not apply dilation so that the final carved volumes would be more faithful to the target shapes.

Sampling. For our 2D results depicted in Figures 13 and 14, the samples p_j are taken from a uniform 20×20 grid on the bounding box of the image. Vectors are also sampled on a unit sphere on every 15 degrees. We have experimented on the number of samples and their influence on the results. As shown in Figure 16, denser samples on the grid produce smoother final results but possibly more number of carvable segments. In addition, having a denser sampling might result in $\mathcal{D}_*$ with larger cardinality since detailed points on the object may need extra directions to be carved.

While uniformly sampling the residual volume is easy, efficient, and likely sufficient for rough machining, it may miss small details on the target object. A possible remedy is to apply an adaptive sampling that is shape-dependent. To obtain such a sampling, we form a set of offsetting "buffers" around the target shape at multiple distances, e.g., 0.1, 0.15, 0.3; see Figure 17(a) and recall that all shapes have been scale-normalized. In addition, to make sure that the residual volume is sufficiently covered, we also coarsely sample the volume and then remove points that are either too close to each other or fall inside the shape. Figure 17(b-d) show decomposition results obtained from this sampling, which can be contrasted to results in Figure 13. Clearly, this simple adaptive sampling allows more surface details to be carved out. One should decide for rough machining to have more segments but a finer result or less segments and a rougher result. This sampling also gives us the possibility of

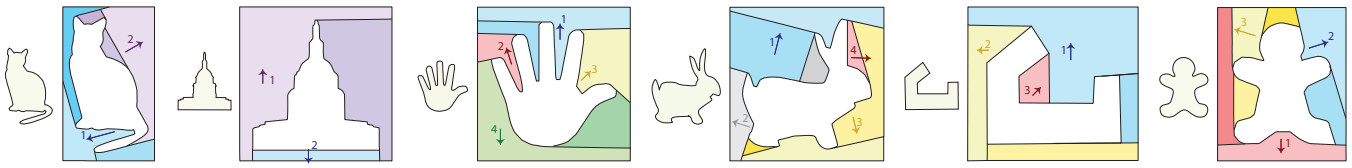


Fig. 13. A gallery of 2D shapes and their carvable decomposition. Segments that are carved under the same direction attain the same color but different shades. The direction numbers indicate carving order; segments with brighter colors are carved earlier.

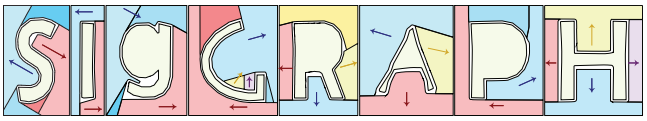


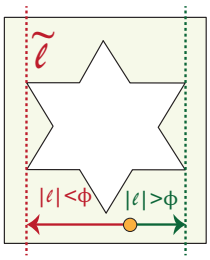
Fig. 14. SIGGRAPH text is carved using our method. We have added a buffer around letters to avoid collision with the text.

controlling the roughness of of the final outcome by having a buffer with a certain distance to the actual object (see Figure 14).

For our 3D results in Figure 15, we have used adaptive sampling with buffer size 0.1 and 0.2mm, with 8192 points sampled on each buffered surface along with a $20 \times 20 \times 20$ grid with sampled vectors every 45 degrees. Note that although our direction sampling is rather coarse, we guarantee accessibility of all sample points by adaptively adding extra directions for inaccessible points. To do this, we double the samples and only add vectors that make inaccessible points accessible. We repeatedly increase sampling until no inaccessible point is present.

Cutter length. Based on our formative study in Section 3, it is desired to use shorter cutters to avoid vibration and cutter breakage. Although we have designed the energy to promote shallower segments, sometimes, simultaneously optimizing the number of directions and carvable segments may result in deep segments. As a result, considering cutter length as a hard constraint is desired.

We have taken a simple approach to incorporate constraints on cutter length ϕ. In the Accessibility Analysis stage, we only consider points p_j as visible along direction d_i, if $\tilde{\ell}$ passing through $p_j + \phi d_i$ and perpendicular to d_i has no intersection with the object. In fact, we have simplified the shape of a cutter as a line ℓ with length ϕ and an infinite base $\tilde{\ell}$ that cannot be obstructed by the shape. Figure 18 shows the effect of decreasing the cutter length on the final decomposition. Having a shorter cutter may result in more number of directions since a point might be visible through many directions but carvable only along a few directions offering a shorter distance to the cutter base. As illustrated in the in-line figure, a point is visible through the red direction but it is not reachable. Therefore, the additional green direction is needed offering shorter distance to the base within the cutter length constraint.



Parameter setting. α and β in our energy function in Equation 2 are set to $\alpha = 1$ and $\beta = 5$, since a fewer number of carvable segments is desired. Larger values for α promotes shallower segments while larger values for β promotes a smaller number of carvable segments. Figure 19 shows the example when $\alpha = 0$ and $\beta = 1$, it is apparent that in comparison with the result of Figure 4(f), deeper segments have been produced. If we change the setting to $\alpha = 1$ and $\beta = 0$, we may get more number of segments in sacrifice of having shallow segments. This setting is shown in Figure 19(d) that serves as a ground truth since the decomposition is obvious. If we use our default setting for the same shape, the result will be the optimal segmentation with less number of carvable segments (Figure 19(c)) but they are deeper than Figure 19(d).

Timing. Table 1 provides the timing of the key steps in our algorithm. As apparent in Table 1, the most time consuming steps in our algorithm are direction assessment and generating carvable volumes since solving set cover and growing carvable regions are relatively more expensive. We have also provided the timing and statistics for 3D objects in Table 2. Tests were run on an Intel i7-8700, 3192 Mhz, six core processor under Windows 10.

Table 1. Algorithm running times (in seconds) and other statistics about some 2D shapes shown in the paper. $|D_*|$ is the number of directions and $|S|$ is the number of segments needed to cover the residual volume. AA, DS, CCV, and DEC are respectively running times for accessibility assessment, carving direction selection, generating candidate carvable volumes, and carvable volume decomposition.

Shapes	$\lvert D_*\rvert$	$\lvert S\rvert$	AA	DS	CCV	DEC	Total
	3	5	0.89	1.80	1.01	0.21	3.91
	3	6	0.63	1.86	1.31	0.61	4.41
	3	7	1.01	2.10	1.12	0.32	4.55
	4	6	1.02	2.12	1.01	1.20	5.35
	2	6	0.91	1.83	2.51	1.15	6.40
	2	3	1.11	5.12	1.39	0.20	7.82
	4	8	1.09	8.72	2.82	1.26	13.89

Comparison with ground truth. To evaluate the performance of our algorithm, we have tested it on a few simple objects depicted in Figure 20. Our algorithm seems to work efficiently for all the cases providing the optimal number of carvable segments. Note that there are some deep segments if we want to optimize the number of carving directions. However, considering cutter length, these segments are avoided. We have provided the result of considering cutter length for some of our shapes in Figure 21. As apparent, for some objects, more directions are needed but segments are shallower.

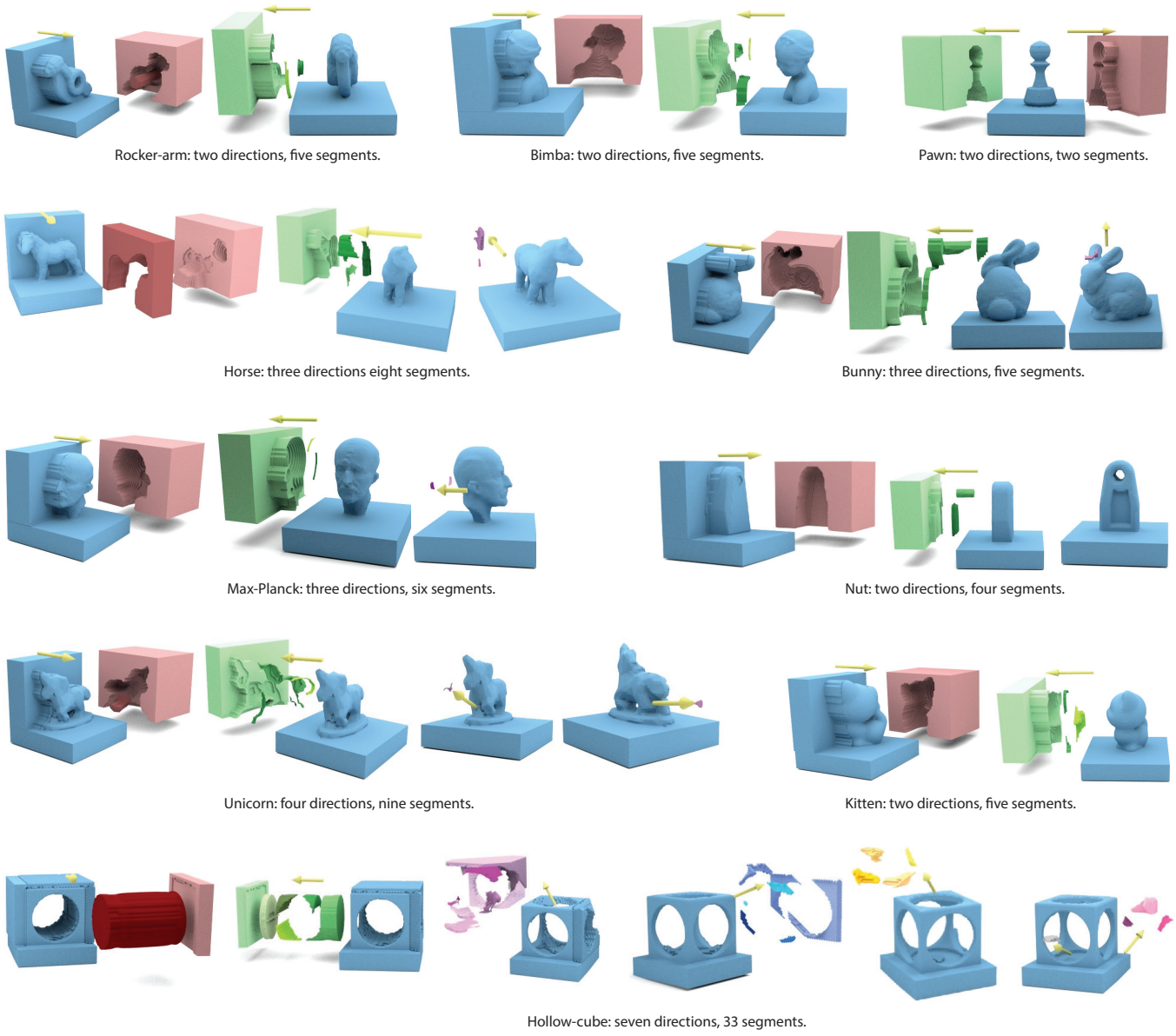


Fig. 15. A gallery of 3D shapes and their carvable volume decompositions visualized step-by-step. Carving directions are indicated by yellow arrows. The square base is part of all the target 3D shapes, as input to the decomposition algorithm.

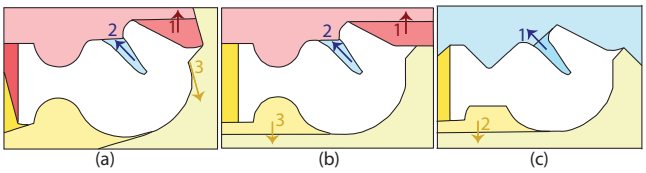


Fig. 16. Results of 30×30 (a), 20×20 (b) and 10×10 (c) uniformly sampling.

Comparison with expert's decomposition. To evaluate our results, we have asked a machining expert to segment a sampler of our objects (see Figure 22). It seems that the experts cannot simultaneously consider the number of directions and carvable segments or they have some intuitions that lead them to specific decomposition results. In all cases, the result of our algorithm provides a smaller or equal number of directions and comparable number of carvable segments. Note that not all the segments that the expert provides are carvable in one pass, therefore more number of segments is needed to fully respect the target shape or a coarser shape is produced. For instance, if you carve the top left figure in Figure 22 (the Gingerman), the red bottom segment becomes the same as our result in Figure 13. In addition, experts implicitly consider shorter cutter

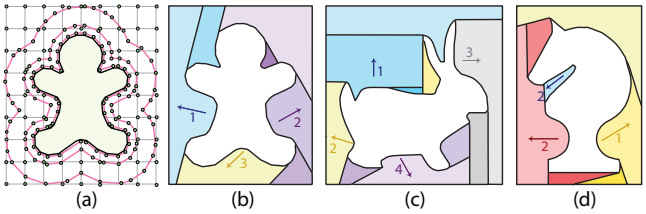


Fig. 17. Adaptive and shape-dependent sampling of the residual volume (a) using a few buffer of the shape along with a coarse uniform sampling to cover the entire residual volume. The resulting decomposition leads to a more faithful reproduction of the target objects, but at the expense of more volumes. Please contrast results shown in (b-d) to those from Figure 13.

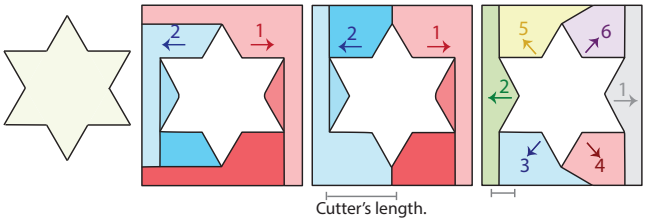


Fig. 18. Cutter length is decreased from left to right. Having a longer cutter results in fewer number of segments to cover the residual volume.

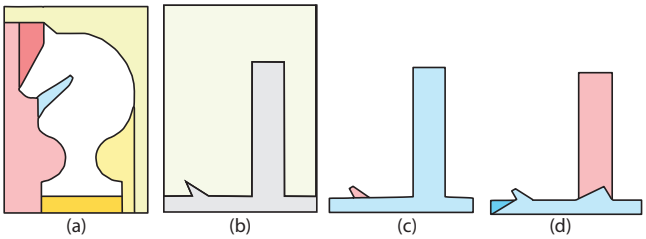


Fig. 19. Changing the default setting to $\alpha = 0$ and $\beta = 1$ produces deeper segments (a) in comparison with Figure 4(f). Given a simple shape (b), the decomposition of the residual volume may promote fewer carvable segments but deeper (c) or more carvable segments but shallower (d).

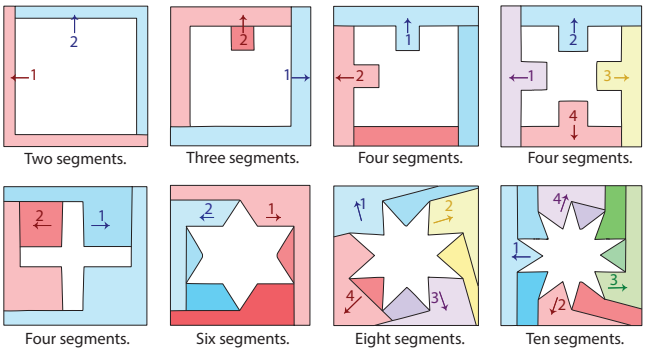


Fig. 20. Results produced by our method on simple 2D shapes for which we were able to determine the optimal number of carvable segments. In each case, our method was able to match the optimal segment count, while not necessarily able to find the most "natural" or obvious decompositions, e.g., in terms of symmetry for the last three examples.

Table 2. Algorithm running times (in seconds) and other statistics about 3D shapes shown in the paper. $|D_*|$ is the number of directions and $|S|$ is the number of segments needed to cover the residual volume. AA, DS, CCV, DEC, and PG are respectively running times for accessibility assessment, carving direction selection, generating candidate carvable volumes, carvable volume decomposition, and cutter path generation.

Shapes	$\|D_*\|$	$\|S\|$	AA	DS	CCV	DEC	PG
Pawn	2	2	6.82	0.01	520	12.81	34.47
Nut	2	4	9.75	0.02	1,353	27.07	24.32
Bimba	2	5	9.40	0.02	1,157	48.55	38.62
Rocker-arm	2	5	10.10	0.01	1,410	28.65	50.29
Kitten	2	5	10.71	0.01	591	14.84	35.78
Bunny	3	5	10.20	0.02	1,070	30.78	41.61
Max-Planck	3	6	11.10	0.02	739	44.49	32.97
Horse	3	8	11.5	0.01	1,543	35.86	38.43
Unicorn	4	9	9.90	0.02	1,365	78.19	34.53
Fertility	6	10	24.01	3.91	2,432	133.54	48.49
Hollow-cube	7	33	30.04	16.65	4,556	302.90	74.35

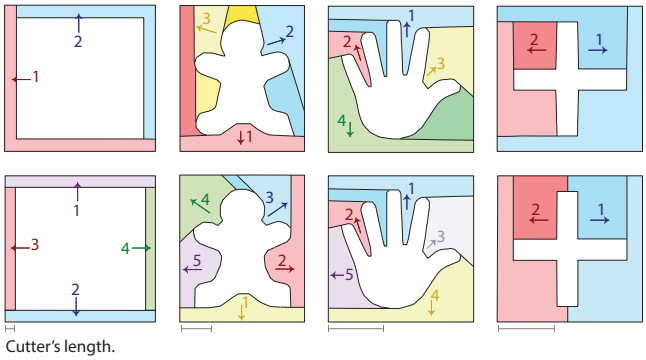


Fig. 21. Constraining cutter length (bottom) can lead to different decomposition results. Top: no constraint on cutter length.

Table 3. Comparing decomposition results between expert solutions and our results, with vs. without constraining cutter length (CL).

shapes	Alg		Alg+CL		Expert	
	$\|D_*\|$	$\|S\|$	$\|D_*\|$	$\|S\|$	$\|D_*\|$	$\|S\|$
	2	2	4	4	4	4
	4	4	4	4	4	4
	2	6	6	6	6	6
	3	7	5	6	6	6
	4	6	5	6	7	7
	3	5	4	7	6	6

lengths for a decomposition. Therefore, for comparison, we also consider cutter length as a constraint and include its relative statistics in Table 3. The results show that, even with these constraints, our solution outperforms the expert's.

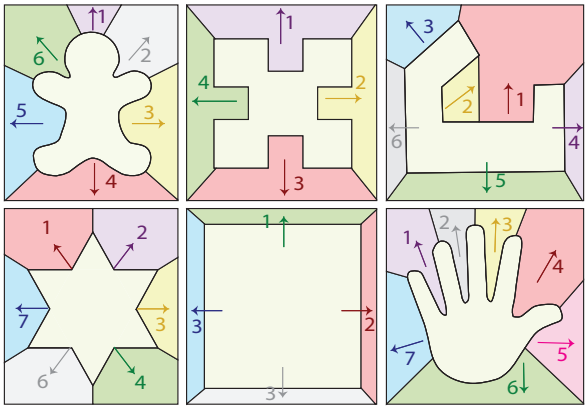


Fig. 22. Results of expert decomposition on a few sample 2D shapes.

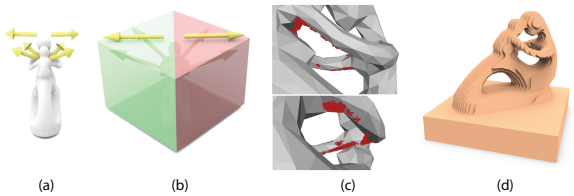


Fig. 23. Experts provide a set of carving directions (a) along with bounding boxes confining the movement of a cutter (b). Using these directions, many points (shown in red) in the residual volume are inaccessible (c). We have simulated the fabrication process using expert's solution (d).

We have also asked an expert to provide the best setup directions for 3D objects (see Figure 23 (a)), but not full volume decompositions, as the latter task is too daunting. Experts also define bounding boxes confining the cutter movement (see Figure 23 (b)). For freeform 3D objects, even finding few setup directions is difficult for humans, as experts' directions often leave many points of the residual volume inaccessible (see Fig 23 (c)). We simulated the machining process based on the expert's solutions and compared with ours (Figure 23 (d)). Respectively, for the Fertility, Bunny, and Kitten models, the fabrication times of our decomposition were 663, 672, and 713 seconds while expert's solutions required 989, 1392, and 1187 seconds. This shows that our fabrication time is respectively about 49%, 107% and 66% shorter than the expert's solutions for the aforementioned models. We generated the tool paths and produced the simulation using Siemens NX 12.0 [Siemens Industry Software Inc. 2020]. Contour-parallel toolpaths were used for our comparison with the same parameters as our actual fabrication.

Comparison with height-field decomposition. Our method is not meant to replace available surface or volume decompositions for molding [Herholz et al. 2015] or CNC machining [Muntoni et al. 2018], since we are solving an entirely different and novel problem, carvable volume decomposition. As already discussed before, an optimal height-field surface decomposition cannot be easily extended to an optimal carvable volume decomposition without considering carvability and other issues related to setup planning.

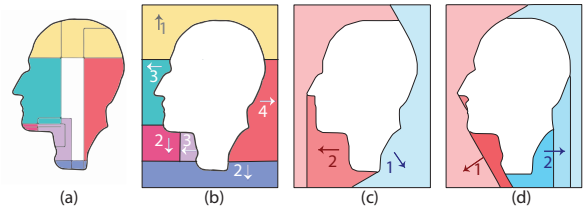


Fig. 24. Segmentation results produced by Muntoni et al. [2018] for 2D MaxPlank (a). Extruding the segmentation of (a) to obtain a volume decomposition (b) results in four directions and six segments. Our method with uniform sampling (c) produces two directions and three segments, and with adaptive sampling (d) produces two directions and six segments.

Segmentations produced by Muntoni et al. [2018] are composed of surface pieces that if assembled, they reconstruct the given shape as a hollow object whose segments can be carved vertically. Therefore, they do not provide a recipe to decompose the residual volume of a given shape. However, 2D segmentation results of Muntoni et al. [2018] can be easily extruded to a valid volume decomposition and our method outperforms their results. In their method, each segment must be height-field individually and carved vertically. Therefore, when segments are combined to form a shape, more directions are needed in comparison to our method as illustrated in Figure 24. In Figure 24 (c) and (d), we illustrate two versions of our algorithm using uniform and adaptive sampling. Since the method provided by Muntoni et al. [2018] is designed for fine machining, we have also provided a finer decomposition using our adaptive sampling. Please also note that, as opposed to our approach, the volumes generated by the method of Muntoni et al. [2018] are not necessarily continuously carvable.

For 3D objects, extruding segments is not trivial as it results in self-intersections specially for high-genus objects. In addition, the outputs of Muntoni et al. [2018] consist of several segments for high-genus models, e.g., 63 for the Fertility model while our method yields 10 segments (see Figure 25 (a)). For genus-zero objects such as the Max Planck, the result generated by Muntoni et al. [2018] attains more directions to cover the entire residual volume (see Figure 25 (b)). Although all the detached segments are height-field along the vertical direction, once they are assembled, they pose arbitrary directions on the model. For instance, our method needs three directions while the method of Muntoni et al. [2018] needs five (removing the directions needed to access the base to match it with our fabricated object). We have simulated and compared the fabrication process for directions provided by Muntoni et al. [2018] against ours (see Figure 25 (c,d)). For the Max Planck model, our method needs 659 seconds while their method needs 893 seconds. Therefore, our method performs 35% better. For Bimba, our method needs two directions while their method requires five; our fabrication time is 690 seconds against 879 seconds (ours is 27% better). However, we must point out that since our decomposition of the residual volume is for rough CNC machining, the reproduced 3D shapes do not approximate the intended target shapes as closely as the method of Muntoni et al. [2018].

Fabrication. Our real machining has been conducted on a 5-axis CNC machine of Pocket NC V2-50, which is *operated as a 3-axis*

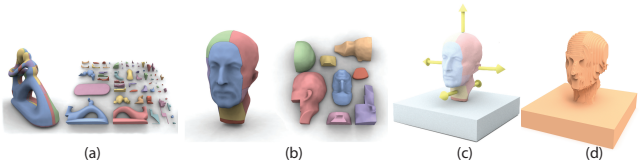


Fig. 25. The results of Muntoni et al. [2018] for the Fertility (a) and Max Planck (b). Carving directions produced by Muntoni et al. [2018] (c) is used to simulate the fabrication process (d).

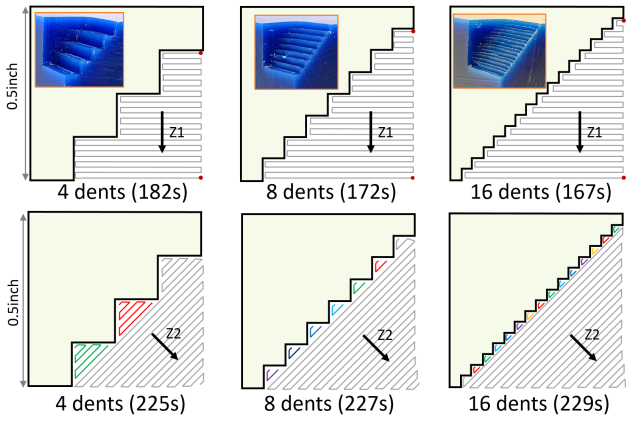


Fig. 26. Comparison on real machining times: continuous vs. non-continuous carving for the same 3D objects.

machine by constraining the degrees of freedom of the cutter. The testing material is machinable square wax (2.0^3 inch). CNC cutting results and analysis are based on the default machine setting: cutter diameter at 1/8 inch, cutter length at 1.5 inch, maximal velocity at 30 inch per minute, feed-rate at 5000*mm* per minute, step over at 2*mm*, step down depth at 1*mm*, and spindle speed at 12,000r per minute. G-code is used to transfer the tool paths. Some outputs of our CNC rough machining are photographed in Figure 27.

Note that all the machined outputs contain a square base, which is part of the target 3D shapes for our volume decomposition algorithm. This base can be easily clamped for the CNC machine we use, while designing fixtures over freeform surfaces is a delicate task which would require human expertise. In our current work, we avoid having to deal with the fixture design problem. Adding the square base to the freeform 3D objects does not compromise the difficulty of the decomposition problem; it may even make it harder, e.g., for the Horse model in Figure 15, since it could increase the topological complexity of a model.

Real machining time: continuous vs. non-continuous carving. To validate the reduction in machining time using continuous carving, we produced a comparison with non-continuous carving of the same target 3D shapes. The tested 3D shapes contain multiple dents, as shown in Figure 26; they are essentially 3D realizations of the 2D examples shown in Figure 2. Six models with different number of dents (4, 8, and 16) have been fabricated from two carving directions. In all cases, each continuous carving follows the connected Fermat spiral

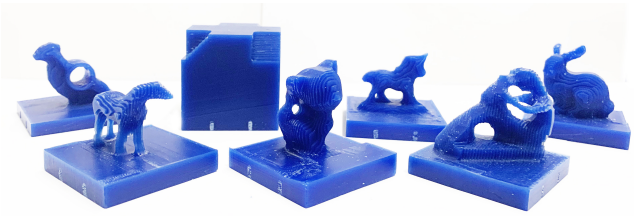


Fig. 27. Rough CNC machining results of some of the models shown in the paper. The CNC machine was set to carve the carvable volumes produced by our decomposition algorithm, following the carving directions computed. Each volume is carved using a connected Fermat spiral toolpath.

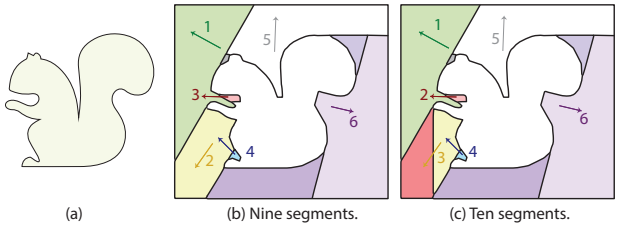
toolpath. With the same machining parameters listed above except for the step down depth at 0.3*mm*, the machining time and fabricated results are marked in Figure 26. We observe that carving the same volume with continuous paths is faster than non-continuous paths and machining time reduction with the continuous paths will increase by adding more dents to the model: about 19% for 4 dents, 24% for 8 dents, and 27% for 16 dents.

7 CONCLUSION AND FUTURE WORK

We present a method to automate setup and path planning for rough-stage CNC machining of free-form 3D objects, including those with high genuses — such models are becoming prevalent as generative design and topology optimization techniques are gaining traction. Our goal is to simplify and accelerate the machining process according to criteria learned from CNC experts. We show how this task can be expressed as a volume decomposition, which jointly optimizes for the number of setup directions and the number of carving path transitions. However, it is important to note that our method is not designed to find the minimum number of carvable volumes. Our joint optimization of setup and path planning *prioritizes* minimizing the number of setup directions. For example, between the choices of 2 setup directions yielding 8 carvable volumes and 3 setup directions yielding 7 volumes, our algorithm would choose the former.

The decomposition problem is challenging and has not been studied before; it introduces (continuous) carvability, a new geometric property which combines visibility and monotonicity. Our proposed solution is based on the observation that minimal sets of carvable directions will generate SA (single access) regions that can assist to optimize the number of carvable volumes. We have applied this insight to the design of an end-to-end pipeline and validated our results through extensive tests, comparisons to ground truth, if available, expert input, as well as physically machined models.

As the first fully automatic solution to carvable volume decomposition, our method still has several limitations which point to potential directions for future work. First, we employ a greedy beam search, which is not guaranteed to find the optimal number of segments, especially when the number of segments or directions is high (i.e., more than four); see Figure 28 for a failure case. Also due to efficiency considerations, we do not account for tiny segments on the surface during decomposition search. Hence, some small artifacts may exist on the results, e.g., see the Fertility model in Figure 1; these can be removed at the fine machining stage. Second,



(a) (b) Nine segments. (c) Ten segments.

Fig. 28. With a greedy search, our method failed at finding a 9-segment decomposition (b); our solution is close but produces 10 segments (c).

we only have a rudimentary solution to incorporate cutter length and width. The erosion and dilation discussed in Section 5, which are based on cutter widths, may lead to loss of small features and changing the hole sizes of high-genus models. Having a model of the actual cutter shape and its base to determine the accessibility of points can be an interesting path to follow. Third, we did not provide any solution for designing fixtures to clamp the 3D objects during machining. While re-configurable pin-type fixtures are adaptable to any shape, it is an important and interesting research direction to provide carving instructions that simplifies clamping. Fourth, our algorithm does not take into account objects symmetry as shown in Figure 20. Therefore, the artifacts produced by rough machining on the fabricated objects including the Fertility in Figure 1 are different from one side to another. An interesting path to follow could be to evaluate shapes more thoroughly including their symmetry and incorporate such properties in an algorithm designed for volume decompositions.

In addition, our segmentation results are dependent on the direction sampling. Although, using adaptive sampling, we can guarantee the accessibility of all points, we may not get the global optimal segmentation under a specific direction set. It might be possible to change the direction set during the segmentation optimization. However, it is not readily obvious how to do this and it will remain as a future work. Our method does not provide visually pleasing segmentations considering symmetric properties of the model as it is shown in Figure 20. In addition, for now the orientation of the object in the block is given and it affects the final segmentation results. It would be interesting to consider the initial orientation as part of the search for optimal carvable segmentation.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their valuable comments. This work was supported in part by NSERC (611370), NSF China (61772318), and gift funds from Adobe and Autodesk.

REFERENCES

M. Al-wswasi, A. Ivanov, and H. Makatsoris. 2018. A survey on smart automated computer-aided process planning (ACAPP) techniques. *The International Journal of Advanced Manufacturing Technology* 97, 1 (2018), 809–832.

T. Alderighi, L. Malomo, D. Giorgi, B. Bickel, P. Cignoni, and N. Pietroni. 2019. Volume-Aware Design of Composite Molds. *ACM Trans. Graph.* 38, 4, Article 110:1–110:12 (2019).

T. Alderighi, L. Malomo, D. Giorgi, N. Pietroni, B. Bickel, and P. Cignoni. 2018. Metamolds: computational design of silicone molds. *ACM Trans. Graph.* 37, 4 (2018), 136:1–136:13.

G. Alemanno, P. Cignoni, N. Pietroni, F. Ponchio, and R. Scopigno. 2014. Interlocking Pieces for Printing Tangible Cultural Heritage Replicas. In *Eurographics Workshop on Graphics and Cultural Heritage*, Reinhard Klein and Pedro Santos (Eds.). The Eurographics Association.

C. Araújo, D. Cabiddu, M. Attene, M. Livesu, N. Vining, and A. Sheffer. 2019. Surface2Volume: Surface Segmentation Conforming Assemblable Volumetric Partition. *ACM Trans. Graph.* 38, 4 (2019), 80:1–80:16.

P. Bose and M. van Kreveld. 2005. Generalizing monotonicity: on recognizing special classes pf polygons and polyhedra. *International Journal of Computational Geometry & Applications* 15, 6 (2005), 591–608.

X. Chen, H. Zhang, J. Lin, R. Hu, L. Lu, Q.-X. Huang, B. Benes, D. Cohen-Or, and B. Chen. 2015. Dapper: decompose-and-pack for 3D printing. *ACM Trans. Graph.* 34, 6 (2015), 213:1–213:12.

B. Choi and R. Jerrard. 1998. *Sculptured Surface Machining: Theory and Applications.* Kluwer Academic Publishers.

D. Ding, Z. S. Pan, D. Cuiuri, and H. Li. 2014. A tool-path generation strategy for wire and arc additive manufacturing. *The international journal of advanced manufacturing technology* 73, 1-4 (2014), 173–183.

S. P. Fekete and J. S. Mitchell. 2001. Terrain decomposition and layered manufacturing. *Int. J. Computational Geometry & Applications* 11, 06 (2001), 647–668.

M. R. Garey and D. S. Johnson. 1983. Computers and Intractability. A Guide to the Theory of NP-Completeness. (1983).

A. Hattab and G. Taubin. 2019. Rough Carving of 3D Models with Spatial Augmented Reality. In *ACM Symp. Computational Fabrication.*

S. Hauth and L. Linsen. 2011. Double-spiral tool path in configuration space. *Int. J. Advanced Manufacturing Technology* 54, 9-12 (2011), 1011–1022.

P. Herholz, W. Matusik, and M. Alexa. 2015. Approximating Free-form Geometry with Height Fields for Manufacturing. *Computer Graphics Forum* 34, 2 (2015), 239–251.

R. Hu, H. Li, H. Zhang, and D. Cohen-Or. 2014. Approximate Pyramidal Shape Decomposition. *ACM Trans. Graph.* 33, 6, Article 213 (Nov. 2014), 213:1–213:12 pages.

A. Jacobson. 2017. Generalized matryoshka: Computational design of nesting objects. *Computer Graphics Forum* 36, 5 (2017), 27–35.

A. Jacobson, D. Panozzo, et al. 2018. libigl: A simple C++ geometry processing library. https://libigl.github.io/.

J. M. Keil. 1985. Decomposing a Polygon into Simpler Components. *SIAM J. Comput.* 14, 4 (1985), 799–817.

D. E. Knuth. 2000. Dancing links. *arXiv preprint cs/0011047* (2000).

A. Lasemi, D. Xue, and P. Gu. 2010. Recent development in CNC machining of freeform surfaces: a state-of-the-art review. *Computer-Aided Design* 42, 7 (2010), 641–654.

M. Livesu, S. Ellero, J. Martínez, S. Lefebvre, and M. Attene. 2017. From 3D models to 3D prints: an overview of the processing pipeline. *Computer Graphics Forum* 36, 2 (2017), 537–564.

B. T. Lowerre. 1976. *The HARPY speech recognition system.* Technical Report. CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER SCIENCE.

L. Luo, I. Baran, S. M. Rusinkiewicz, and W. Matusik. 2012. Chopper: Partitioning models into 3D-printable parts. *ACM Trans. Graph.* 31, 6 (2012), 129:1–129:9.

L. Malomo, N. Pietroni, B. Bickel, and P. Cignoni. 2016. FlexMolds: automatic design of flexible shells for molding. *ACM Trans. Graph.* 35, 6 (2016), 223:1–223:12.

A. Muntoni, M. Livesu, R. Scateni, A. Sheffer, and D. Panozzo. 2018. Axis-aligned height-field block decomposition of 3D shapes. *ACM Trans. Graph.* 37, 5 (2018), 169:1–169:15.

A. Rivers, I. E. Moyer, and F. Durand. 2012. Position-correcting Tools for 2D Digital Fabrication. *ACM Trans. Graph.* 31, 4 (2012), 88:1–88:7.

A. Sá, K. Rodriguez-Echavarria, N. Pietroni, and P. Cignoni. 2016. State of the art on functional fabrication. In *Proceedings of the Eurographics Workshop on Graphics for Digital Fabrication.* Eurographics Association, 1–9.

Siemens Industry Software Inc. . 2020. NX software. www.plm.automation.siemens.com/global/en/products/nx/.

O. Stein, A. Jacobson, and E. Grinspun. 2019. Interactive design of castable shapes using two-piece rigid molds. *Computers & Graphics* 80 (2019), 51–62.

G. Toussaint. 1985. Movable separability of sets. *Comput. Geom.* (1985), 335–375.

X. Wei, A. Joneja, and D. M. Mount. 2012. Optimal uniformly monotone partitioning of polygons with holes. *Computer-Aided Design* 44 (2012), 1235–1252.

Y. Yang, H. T. Loh, J. Fuh, and Y. Wang. 2002. Equidistant path generation for improving scanning efficiency in layered manufacturing. *Rapid Prototyping* 8, 1 (2002), 30–37.

H. Zhao, F. Gu, Q.-X. Huang, J. Garcia, Y. Chen, C. Tu, B. Benes, H. Zhang, D. Cohen-Or, and B. Chen. 2016. Connected Fermat spirals for layered fabrication. *ACM Trans. Graph.* 35, 4 (2016), 100:1–100:10.

H. Zhao, H. Zhang, S. Xin, Y. Deng, C. Tu, W. Wang, D. Cohen-Or, and B. Chen. 2018. DSCarver: Decompose-and-Spiral-Carve for Subtractive Manufacturing. *ACM Trans. Graph.* 37, 4 (2018), 137:1–137:14.