











Continuous Toolpath Optimization for Simultaneous Four-Axis Subtractive Manufacturing

Zhenmin Zhang,¹  Zihan Shi,¹  Fanchao Zhong,¹  Kun Zhang,¹  Wenjing Zhang,¹  Jianwei Guo,²  Changhe Tu¹ 
and Haisen Zhao¹ 

¹School of Computer Science and Technology, Shandong University, Qingdao, China
zhenminzhang.cn@gmail.com, szh211227@163.com, fanchaoz98@gmail.com, kun_zhang2004@foxmail.com, wenjingzhang4af@outlook.com, {chtu, haisenzhao}@sdu.edu.cn
²MAIS, Institute of Automation, Chinese Academy of Sciences, Beijing, China
jianwei.guo@nlpr.ia.ac.cn

Abstract

Simultaneous four-axis machining involves a cutter that moves in all degrees of freedom during carving. This strategy provides higher-quality surface finishing compared to positional machining. However, it has not been well-studied in research. In this study, we propose the first end-to-end computational framework to optimize the toolpath for fabricating complex models using simultaneous four-axis subtractive manufacturing. In our technique, we first slice the input 3D model into uniformly distributed 2D layers. For each slicing layer, we perform an accessibility analysis for each intersected contour within this layer. Then, we proceed with over-segmentation and a bottom-up connecting process to generate a minimal number of fabricable segments. Finally, we propose post-processing techniques to further optimize the tool direction and the transfer path between segments. Physical experiments of nine models demonstrate our significant improvements in both fabrication quality and efficiency, compared to the positional strategy and two simultaneous tool paths generated by industry-standard CAM systems.

Keywords: digital geometry processing, toolpath planning, four-axis machining

CCS Concepts: • Computing methodologies → Mesh geometry models

1. Introduction

CNC subtractive manufacturing (SM), a cornerstone of modern industry, has continually evolved to meet the growing demands for precision and complexity in the fabrication of various components and products [LXG10, SKM*22]. In this context, the utilization of four-axis machining has emerged as a crucial and cost-effective technique, bridging the gap between the accessibility of three-axis CNC machines and the intricate capabilities of five-axis CNC machines. The significance of four-axis machining lies in its ability to strike a delicate balance between shape complexity and machine accessibility. In contrast to three-axis machining, the additional axis in four-axis CNC machines enables the creation of intricate and multifaceted designs, such as side drilling or surface cylinder drilling. Moreover, it improves productivity by allowing multiple operations to be performed in one setup. In contrast to five-axis machining, the four-axis CNC machines are much more cost-effective while realizing complex geometries, resulting in much more accessibility

in the manufacturing field (According to Statistics MRC [Gii23] and a research report [Wic24], the global four-axis CNC Machining Center Market and the global five-axis CNC Machining Centers market were valued at \$34,012.22 million and \$4,119.9 million in 2023, respectively.). As a result, four-axis machining has found a wide range of applications, especially in aerospace, automotive and medical industries, where the fusion of precision and artistry is paramount [JLZ*21, ZRZ23]; see Figure 2.

A four-axis CNC machine has three degrees of translation and one degree of rotation. This rotational capability enables the machine to perform complex and versatile machining operations, creating intricate and precise designs. See Figure 4 for the machine setup. To perform SM using a four-axis CNC machine, we need to determine tool direction and its movement on the fine-machining surface of the target 3D shape. The movement of the cutter refers to the machining sequence, which indicates the next machining points after carving the current one. Four-axis machining presents

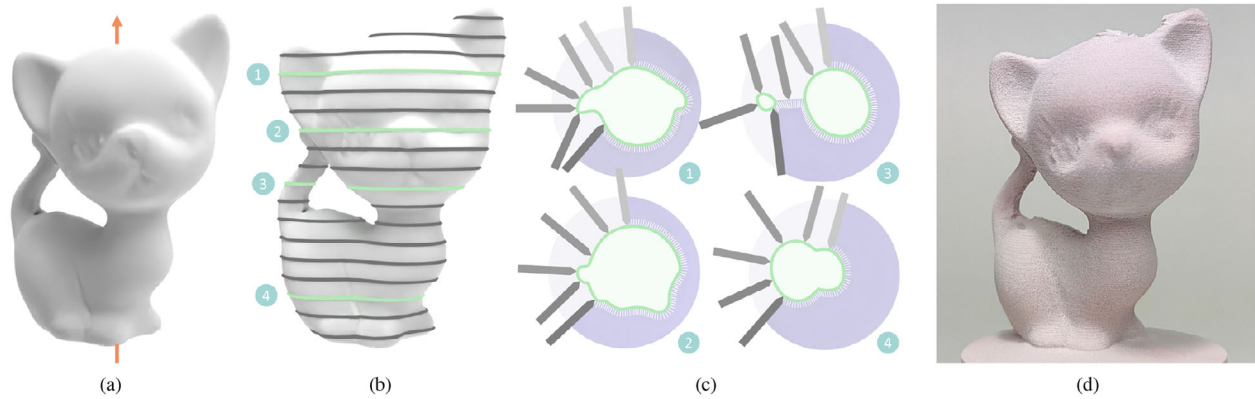


Figure 1: Our algorithmic pipeline in brief. This work introduces a general computational framework for simultaneous four-axis CNC machining, which aims to minimize the variation of tool direction while machining continuously as much as possible, and to ensure that the fabrication process is always collision-free. This figure shows the Kitten model (a). We uniformly slice it along the rotation axis after determining its object orientation (b). Next, we apply tool path optimization to each layer to generate a simultaneous four-axis tool path. This aims to maximize geometric continuity and minimize variations in the machining directions (c). (d) displays the physical fabrication outcome of the Kitten model. It indicates that the proposed framework can be successfully used for simultaneous four-axis subtractive manufacturing.

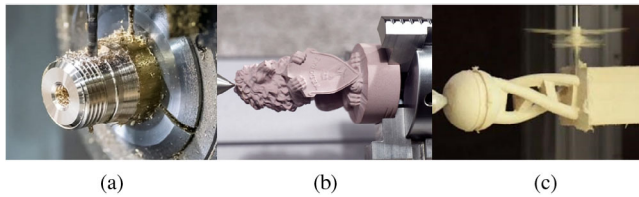


Figure 2: Typical application products of four-axis machining. Four-axis subtractive manufacturing is widely used in metal machining of solids of revolution (a), wooden crafts (b) and prototypes with high-genus (c).

a critical decision point with two primary strategies: positional and simultaneous machining strategies.

The *positional strategy* (also known as the positional 4th axis, or 3 + 1 machining) maintains a fixed tool direction during cutting using three translation degrees. The remaining rotational degree involves moving the cutter between cutting material from different directions. To apply this strategy, the external surface of the target 3D shape is first decomposed into height-field patches [NTM*21]. Each patch can be carved with a specific tool direction without any collisions. Then a tool path planning process is applied to determine the cutter movement to carve each patch.

The key benefit of the positional machining strategy lies in its simplicity, which determines tool direction and movement in two independent computational stages, as noted in Nuvoli et al. [NTM*21]. However, it suffers from the presence of boundary artefacts between neighbouring patches, which arise from discontinuous tool paths carved from different directions. As shown in Figure 3, the boundary artefacts require additional manual post-processing work to achieve the desired surface finish. In contrast, these concerns are anticipated to be addressed through the simultaneous machining strategy, which can greatly reduce the occurrence of boundary artefacts.



Figure 3: Manufacturing demonstration of the positional machining strategy. The figure is from Nuvoli et al. [NTM*21], which fabricates the Kitten model by carving it from a set of height-field patches and then applying manual post-processing work.

The *simultaneous strategy* (also called true 4th-axis machining) involves a cutter moving in all four degrees of freedom simultaneously during carving. This indicates that both tool direction (rotational degree) and cutter movement (three translational degrees) should be determined simultaneously during tool path planning. In the simultaneous strategy, the cutting tool gradually changes its direction throughout the machining process. Consequently, the simultaneous strategy gains its primary advantage of superior machining surface quality, without the need for post-processing of the boundary artefacts present in the positional strategy, which arise from discontinuous tool paths and dramatically different tool directions. Therefore, to guarantee high-quality fabrication and efficiency, the simultaneous strategy should ensure two key properties of the generated tool path during its planning phase: directional and geometric continuity. **Geometric continuity** refers to minimizing the number of machining tool paths, as discontinuous paths invariably generate numerous transition paths that can hinder overall machining efficiency. **Directional continuity** refers to the smooth and consistent variation in tool direction, as frequent changes can result in defects in surface finishing and reduced machining efficiency.

However, it is challenging to ensure the proposed directional and geometric continuity during the collision-free tool path planning for the simultaneous SM on four axes. We cannot simply replicate the two independent computational stages of the positional strategy. This is due to the coupling between tool direction and its movement in the simultaneous strategy. Different tool directions can result in different machining sequences, which can affect the optimization of tool direction. Unfortunately, the simultaneous tool path planning problem for four-axis CNC machines remains an open research area. To our knowledge, only a few solutions are available in industry CAM systems. However, these methods are capable of producing objects with only simple geometries. For complex geometries featuring high-genus shapes or numerous branching structures, current solutions fall short and often encounter issues such as overcuts or undercuts, as detailed in related works.

We aim to propose a novel end-to-end framework for producing a collision-free tool path with directional and geometric continuity for simultaneous four-axis machining. This proposed method can be utilized to fabricate complex 3D shapes, including those characterized by high-genus shapes or numerous branching structures; see Figure 1. We target the finishing (fine-machining) stage, which is performed using ball-end mills or straight groove pointed tools, where we assume that only the spherical and conical parts of the tool have cutting capability. The specific tool shapes can be found in the results section. Our approach addresses this tool path planning problem in two ways. First, we propose simplifying the problem by transforming the 3D tool path issue into a 2D planning problem. We accomplish this using a layer-based fabrication approach. By dividing the target object into slices, we tackle tool path planning for a simultaneous machining strategy for each individual layer. This allows us to break down the problem into an appropriate level of complexity for initiating research in simultaneous four-axis SM. We refer to the boundary of each connected component in each slicing layer as a **contour**. There may be one or more contours within a layer. Second, we propose an over-segmentation and then a bottom-up merging process to co-optimize the tool direction and its movement. Specifically, we break down tool path planning for each layer into three computational stages. In the over-segmentation stage, each layer contour is uniformly decomposed into atomic segments, and accessibility analysis is then applied to each segment. The bottom-up merging aims to generate an as-continuous-as-possible machining tool path by merging these segments with a back-and-forth traverse procedure, a graph-cut-based overlapping resolving procedure and a TSP connection procedure. The third stage involves a post-processing optimization to further enhance the directional continuity and shorten the transition path.

In summary, our key contributions include the development of the first general computational framework for simultaneous four-axis SM, focusing on continuous tool path generation with minimized directional variation and a minimized number of transition paths. Nine fabrication and three ablation experiments have been conducted to confirm the effectiveness of our technology. Furthermore, three comparison experiments have been performed using the industry-standard CAM systems of Snapmaker [Sna23] and Autodesk [Wor23], as well as the positional four-axis machining introduced by Nuvoli *et al.* [NTM*21] to demonstrate significant improvements in manufacturing quality and efficiency.

2. Related Works

Tool path planning is a classical research topic in the CNC machining domain. It has been addressed by a broad range of approaches, such as the parameterization method, which maps a curved surface to the plane [RSG09], the drive surface-based method, which generates iso-planar tool paths by intersecting parallel planes [CJ12, HBA13], and iso-scallop tool paths which pursue the uniform scallop distribution [CÜ10, LKLF21], particularly studied in the context of five-axis CNC machining [MPE17, EE18, BBR*21]. Rather than aiming to conduct a comprehensive survey of the entire tool path planning field in CNC machining [YJJ*22], we focus specifically on existing work on the tool path planning strategy for four-axis CNC machining. This section initially examines previous work on the positional machining strategy.

In terms of simultaneous strategy, despite the existence of several commercial CAM systems in the industry, we cannot find any research literature addressing the simultaneous tool path planning problem for four-axis CNC machining.

Furthermore, we review works aiming at directional and geometric continuity in different manufacturing domains.

2.1. Positional machining strategy

The most critical aspect of this strategy is to minimize the use of positional directions to process the entire surface of the target 3D shape. This issue has been well addressed by various methods. In Ding and Jiang [DJ04], an interaction-based method is proposed that involves users in assigning machining orientations for free-form surfaces in electric discharge machining applications. Muntoni *et al.* [MLS*18] propose a method that decomposes a 3D object into height fields, then projects the decomposition toward the interior, covering the entire volume and ensuring that each piece is manufacturable with three-axis CNC machines. For CAD models, which are constructed using precise geometric primitives such as lines, arcs, circles and polygons, Joshi [Jos15] proposes a method to determine orientations based on these primitives. In the case of non-complex parts where all features can be machined from two directions, Zahid *et al.* [ZCW16] aim to find the best pair of two orientations that avoid thin web structures and preserve cutter life. In Fanni *et al.* [FCM*18], a polycube representation of the original shape is used for surface decomposition in four-axis CNC machining.

Researchers also address the positional machining strategy in a surface decomposition process. In this process, the external surface of the target 3D model is decomposed into a minimal number of height-field patches using multi-label graph-cut optimization [STC09]. Each decomposed height field patch is associated with a single tool direction. This graph-cut-based surface decomposition method has proven its effectiveness for three-axis machining [HMA15], four-axis machining [NTM*21], five-axis machining [ZZX*18] and volume decomposition for moulding [AMG*19]. In our approach, we also utilize multi-label graph-cut optimization. However, instead of aiming to reduce the number of tool directions, we focus on minimizing the number of machining segments along the contour of each layer. This approach helps to achieve the desired geometric continuity necessary for the simultaneous SM of the four-axis.

Regardless of whether it is for the positional or simultaneous machining strategy, accessibility analysis is crucial in order to produce a fabrication plan that is free of collisions. For the positional machining strategy, Frank *et al.* [FWJ06] involve slicing the layers of the input 3D model for accessibility analysis. They start by computing 2D visibility maps for the set of sliced contours on each layer. They then determine the minimum number of tool directions based on these 2D visibility maps. However, it is important to note that their approach relies on other CAM software to generate the positional machining tool paths for each direction of the determined tool directions. In our case, we can generate simultaneous machining tool paths using our own algorithm.

2.2. Simultaneous machining strategy

To our knowledge, our research is the first to address the co-optimization of tool direction and movement for the four-axis simultaneous SM. Consequently, most of the relevant literature comes from CAM systems used in the industry domain. However, due to the abundance of CAM systems and the lack of technique reports for commercial purposes, it becomes impossible to explore and fully understand all of them and their algorithms in detail. To provide a brief understanding of the simultaneous four-axis strategy in the industry, let us explore a selection of the options that we currently have access to, including the Luban software attached to our Snapmaker four-axis CNC machine of Snapmaker [Sna23], the Fusion 360 software developed by Autodesk [Wor23] and the Siemens NX [Sie16].

After examining the source code released by Snapmaker [Sna23], we discovered that Luban addresses four-axis machining by utilizing the convex hull of sliced contours. It generates a 360-degree tool path along the boundary of the convex hull for each slicing layer and projects it onto the sliced contours. However, our experiments indicate that it can only handle simple geometric models, which consist of a single contour in each layer, and cannot process models with multiple contours in each layer. Autodesk's Fusion 360 offers a 'rotary' finishing strategy [Wor23] for the simultaneous four-axis machining. Users have the option to select from three rotating tool paths: Spiral, Line or Circular. However, our experiments reveal that Fusion 360 fails to produce a completely collision-free machining path, resulting in numerous undercuts. Siemens NX offers a semi-automatic strategy for generating simultaneous tool paths for CAD models [Sie16], which necessitates manual specification of drive geometry or guiding curves to generate tool paths at a feature-based level, such as a circular pocket or a slot feature. In contrast, our technique provides a fully automatic solution for generating simultaneous tool paths for the entire model. Therefore, it is unnecessary to compare our technique to Siemens NX.

2.3. Continuous tool path planning

Continuity is an important and desirable characteristic for tool path planning in various manufacturing domains. It has a significant impact on the efficiency of manufacturing and product quality. Previous studies have attempted to enhance directional continuity or geometric continuity [MSJ*22], which serves as inspiration for our

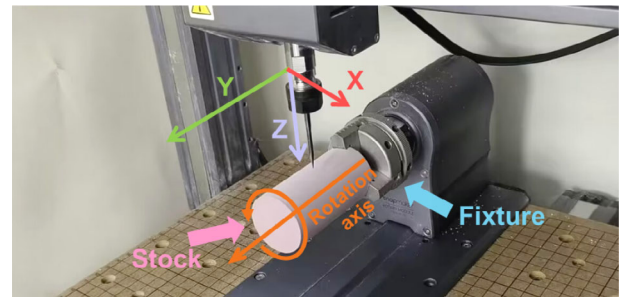


Figure 4: The setup of our four-axis machine. The milling tool has 3 degrees of freedom (DOF), and the rotation axis provides the fourth.



Figure 5: Demonstration of the fabrication process. After assembling the stock on the fixture, we mill the stock layer by layer along the rotation axis using the four-axis machine.

algorithm to optimize the continuity of tool direction and the machining sequence.

First, for directional continuity, we adopt the graph-based representation utilized in our graph-cut step, akin to Plakhotnik and Lauwers [PL14], which determines tool direction through graph-based optimization for five-axis machine tool movements. Second, for geometric continuity, we employ the over-segmentation and merging strategy similar to that employed in Zhao *et al.* [ZGH*16] and Zhong *et al.* [ZXZL23]. In these works, a 2D domain (a surface model) is initially decomposed into several sub-domains (small-scale one-path patches). Subsequently, a single Fermat spiral (a single one-path patch) is generated through a bottom-up merging process. In our case, we decompose short segments of each layer and then connect them into a single tool path.

3. Overview

This section begins by explaining the configuration of our four-axis CNC machine and provides an overview of our fabrication process. Then it re-iterates the basic idea of our computational framework, accompanied by a technique overview.

3.1. Setup and fabrication process

In our experiments, we utilize the four-axis CNC machine: Snapmaker 2.0 A350T, manufactured by Snapmaker [Sna23]. It consists of three linear axes of movement and an additional axis of rotation, as shown in Figure 4. In the fabrication process, we first assemble the stock on the four-axis machine through the fixture so that its axis is aligned with the rotation axis. Then we mill the stock layer by layer, as shown in Figure 5.

3.2. Technique overview

Recall that we aim to propose an algorithm that forms an end-to-end framework for simultaneous four-axis machining. Our algorithm takes a 3D object \mathcal{M} , represented by a triangular mesh, as input, and generates a collision-free simultaneous tool path $\mathcal{TP} = \{TP_1, TP_2, \dots, TP_n\}$ for four-axis machining. Here, TP_i represents the tool path of the i th slicing layer. Each TP_i comprises a sequence of machining segments and transfer move paths, denoted as $TP_i = \{S_1^i, T_{1,2}^i, S_2^i, \dots, T_{m-1,m}^i, S_m^i\}$, where S_k^i is a continuous machining tool path segment of the i th slicing layer, $T_{k-1,k}^i$ is a transfer move path between S_{k-1}^i and S_k^i . To ensure directional and geometric continuity of TP_i , our framework is to reduce the sequence length of TP_i and minimize the changes in tool direction within each TP_i . Our algorithm achieves these objectives through three stages:

- (1) In initialization, we first determine an object orientation of \mathcal{M} (Section 4.1), then slice it into n layers, $\mathcal{L} = \{L_1, L_2, \dots, L_n\}$, and uniformly sample sub-segments along each slicing contour C_j^i of layer L_i . Then, we conduct an accessibility analysis for each sub-segment (Section 4.2).
- (2) In the over-segmentation-then-merging process, we first decompose each contour C_j^i into a set of machining segments with a proposed back-and-forth traverse process. Then a graph-cut method is used to resolve the overlapping between these segments, resulting in machining segments $\{S_1, S_2, \dots, S_n\}$ (Section 4.3). Connect the result machining segments into a single tool path TP_i of layer L_i , while considering minimizing the length of transfer moves (Section 4.4). For each TP_i , apply a post-processing to further optimize its connection points and machining directions (Section 4.5).
- (3) Connect tool paths $\{TP_1, TP_2, \dots, TP_n\}$ of each layer into a single tool path \mathcal{TP} by determining one connecting point for each TP_i . As each TP_i is a circuit, one connecting point is both the starting and existing endpoint of TP_i . To simplify the computation, we select the connecting point of TP_i with the maximum Z coordinate. Finally, generate \mathcal{TP} by inserting transfer move paths with tool retracting between two connecting points of adjacent layers.

4. Methodology

This section provides a more detailed description of each step of our algorithm. For every layer, our algorithm aims to produce a tool path with minimized variations of tool directions (directional continuity) and a minimized number of transfer moves (geometric continuity), which can be directly used in simultaneous four-axis CNC machining.

4.1. Object orientation

The object orientation refers to the alignment of the object relative to the rotation axis of the four-axis machine. Before tool path planning, we first determine the orientation of \mathcal{M} using a method similar to the one in Nuvoli et al. [NTM*21]. The only difference is that we do not divide the top area of the model to avoid the seam line caused by decomposition. Our main consideration is to avoid flat areas whose normal direction is nearly parallel to the rotation axis, which are eas-

ily omitted by slicing. To determine the object orientation, we begin by generating a set of candidate orientations $\{\vec{d}_1, \dots, \vec{d}_k\}$. This is achieved by uniformly distributing points in a hemisphere using a Fibonacci sphere algorithm [SJP06] (with $k = 2000$ in our implementation). The best orientation is selected from $\{\vec{d}_1, \dots, \vec{d}_k\}$, which yields the maximum criterion:

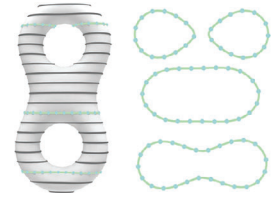
$$A(\vec{d}_i) = \sum_{f_j \in \mathcal{M}} a_j * \left(1 - \left| \vec{n}_j \cdot \vec{d}_i \right| \right) \quad (1)$$

where a_j is the area of the triangle face f_j , \vec{n}_j is the surface normal of f_j and \vec{d}_i is the candidate direction of rotation axis.

4.2. Accessibility analysis

Collision-free is a hard constraint for tool path planning. We tackle this through a pre-computation process during the initialization stage. Here, we pre-compute the machinable direction range for each surface point of \mathcal{M} . This process involves a slicing-and-sampling approach followed by a collision-detection-based accessibility analysis process for each sampled surface point.

Slicing and sampling. We slice \mathcal{M} with flat planes vertical to the rotation axis, resulting in n slicing layers, $\mathcal{L} = \{L_1, L_2, \dots, L_n\}$. Following Lee [Lee03], the dependency between the scallop h and the slicing thickness t between adjacent layers of the path is empirically formulated as



$$h = t^2 / (8 * R) \quad (2)$$

where R is the radius of the tip of the tool. To achieve sufficient machining accuracy, we set the slicing thickness to 0.2 mm and use a tool with a radius of 0.15 mm, which allows us to obtain a scallop height of 0.033 mm; see Figure 13. On each layer L_i , there will be one or multiple slicing contours $\mathbf{C} = \{C_1^i, C_2^i, \dots, C_m^i\}$, where C_j^i denotes the j th contour of the i th layer. Then we uniformly sample atomic segments $\mathcal{A}^{i,j} = \{a_1^{i,j}, a_2^{i,j}, \dots, a_k^{i,j}\}$ along C_j^i , with a spacing of 0.2 mm, where $a_k^{i,j}$ is the k th atomic segment on the j th contour of the i th layer. The inset figure shows the output of slicing and uniform sampling.

Collision detection. This step aims to compute the machinable direction range, abbreviated MDR, for each atomic segment $a_k^{i,j}$. This can be estimated by determining the MDR of the middle point of $a_k^{i,j}$ on the 2D slicing layer, thanks to the layer-based strategy employed in our technique. The MDR of $a_k^{i,j}$ is composed of 2D machinable direction sectors, abbreviated as MDS. The directions within these sectors are collision-free, allowing the CNC tool to carve $a_k^{i,j}$ without interference. The $a_k^{i,j}$'s MDR can include multiple MDS, $\{MDS_1, MDS_2, \dots, MDS_n\}$, in which the t th MDS is defined with a starting boundary angle A^s and an ending boundary angle A^e , indicated as $MDS_t = (A^s, A^e)$.

We calculate MDR of $a_k^{i,j}$ mainly based on a sampling-based method. We first uniformly sample the candidate machining

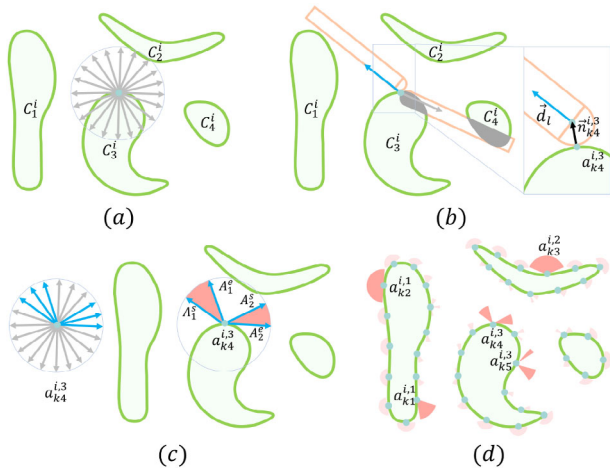


Figure 6: Demonstration of accessibility analysis. (a) Uniform sampling of machining directions. (b) Collision detection for sampling direction, showing examples with and without collisions. (c) Results of collision detection, and the produced MDR from the accessible machining directions. Red sectors denote MDS of each atomic segment. (d) MDR results of the layer L_i . $a_{k2}^{i,1}$ has an almost 180° machinable range. $a_{k3}^{i,2}$ is located in a concave area, with a smaller MDS. $a_{k4}^{i,3}$ and $a_{k5}^{i,3}$ have two divided MDS, due to the occlusion of contour C_4^i .

direction at 5° intervals, resulting in $\{\vec{d}_1, \dots, \vec{d}_n\}$. Let $\vec{n}_k^{i,j}$ be a normalization vector to the surface normal of $a_k^{i,j}$. For each candidate direction \vec{d}_i , we designate $a_k^{i,j}$ as the cutter contact (CC) point. Next, we position the cutter location point (CL), which in this paper is the centre of the sphere for the ball-end mill or the straight groove-pointed tool, at $a_k^{i,j} + R * \vec{n}_k^{i,j}$ (R indicates the radius of the sphere). We then align the cutter's direction with \vec{d}_i . After that, we detect the collision between the cutter and \mathcal{M} by checking if any of the sampled atomic segments on L_i lie inside the cutter. If none of the sampled atomic segments are inside the cutter, then no collision occurs, and the \vec{d}_i is machinable. Finally, we group all the machinable directions into the machinable sectors of $a_k^{i,j}$, assigning the starting and ending boundary angle A^s and A^e for each MDS $_i$, as shown in Figure 6(a)–(c). Figure 6(d) shows an example of the accessibility analysis result. Note that while tool posture can be uniquely determined by the machining direction for the two tools above, custom-shaped tools like the ellipsoid need an additional variable. Hence, the above method does not support these tools.

4.3. Path segments decomposition

This step aims to decompose each contour C_j^i of i th layer into the minimum number of continuous machining tool paths, $\{S_1, S_2, \dots, S_n\}$. We define such a continuous path as a **path segment** S_k , which can be machined continuously with minimal variation of tool directions. A path segment can be seen as a basic item with the desired directional and geometric continuity. For the simultaneous SM of each layer, the fewer path segments a layer contains, the better its directional and geometric continuity. Our pri-

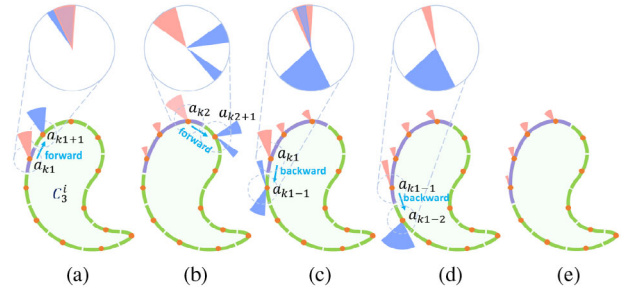


Figure 7: Path segment generate by back-and-forth traverse. Starting from atomic segment a_{k1} , (a) we first traverse its adjacent atomic segment a_{k1+1} on contour C_3^i in a clockwise direction. With the traverse MDS overlapping, a_{k1+1} is included in the path segment. (b) The clockwise traversal terminates at a_{k2+1} , with no MDS overlap between a_{k2} and a_{k2+1} . (c) We then traverse the contour counter-clockwise from a_{k1} , merging a_{k1-1} and terminate at a_{k1-2} . (e) Finally, we generate a path segment that contains four atomic segments.

mary idea to the problem of path segments decomposition is the over-segmentation-and-then-merging strategy.

With the technique described in Section 4.2, each contour C_j^i has been ‘over-segmented’ into atomic segments $\mathcal{A}^{i,j} = \{a_1^{i,j}, a_2^{i,j}, \dots, a_n^{i,j}\}$. Each of these points can be considered an initial path segment. The next step is to further merge these initial path segments. We attempt an iteratively greedy-based method, in which each iteration produces one path segment from C_j^i . However, this approach often fails to achieve the minimum number of path segments. As a result, a graph-cut based method is proposed. Before delving into the two methods, it is important to first describe a back-and-forth traverse process, which will be utilized in both methods.

Back-and-forth traverse. This process aims to generate the longest path segment (S_k) from C_j^i , starting from one of its atomic segment $a_k^{i,j}$. When calling the back-and-forth traverse process, a specific MDS (MDS_t) of $a_k^{i,j}$ must be input, referred to as the **traverse MDS** of $a_k^{i,j}$, to generate $S_k^{i,j}$. Starting from $a_k^{i,j}$, we traverse C_j^i both forward and backward. Initially, we traverse the backward atomic segment $a_{k-1}^{i,j}$ and the forward atomic segment $a_{k+1}^{i,j}$. During this traversal, if one of the MDS of the encountered atomic segment $a_{enc}^{i,j}$ overlaps with the traverse MDS of the current atomic segment $a_{cur}^{i,j}$, we include it in the path segment and designate that MDS as the traverse MDS of $a_{enc}^{i,j}$; see Figure 7. If $a_{enc}^{i,j}$ is successfully included, it indicates that the machining tool can continue machining between $a_{cur}^{i,j}$ and $a_{enc}^{i,j}$ using any tool direction within the overlapping sectors of their traverse MDS.

Greedy-based path segments decomposition. This method uses a heuristic-driven greedy strategy. The key heuristic rule is to call back-and-forth traverse to generate as few path segments as possible while determining a traverse MDS for each atomic segment. First, we randomly select an atomic segment $a_k^{i,j}$ from all the atomic segments of C_j^i and randomly determine its traverse MDS. Then, we call back-and-forth traverse to generate the longest path segment from $a_k^{i,j}$. Next, we repeat the previous operation to generate the longest path segment from the remaining atomic segments of C_j^i .

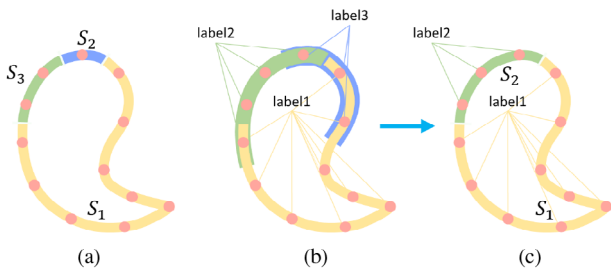


Figure 8: Two strategies for generating path segments. (a) Greedy-based method. The contour is decomposed into three non-overlapping path segments. (b) Graph-cut method: generate candidate path segments on the contour. Then, we use the graph-cut algorithm to resolve the overlapping between path segments. (c) Finally, we get two path segments, which is fewer than with the greedy method.

Here, the remaining atomic segment refers to those not yet included in the already generated path segments. The iterations continue until the generated path segments include all the atomic segments, thereby also determining the traverse MDS for each atomic segment, as shown in Figure 8(a). A complex scenario applying the greedy method can be found in Figure 14.

Obviously, this method does not produce overlapping path segments. However, such a greedy-based method could easily overlook the optimal solution. This is because an atomic segment can have multiple MDS. Merging the atomic segment into a path segment based on different MDS may yield a varying number of path segments.

Graph-cut-based path segments decomposition. This method initially generates a set of potential path segments to machine C_j^i . It does this by running the back-and-forth traverse, starting from each MDS of every atomic segment of C_j^i . Because a unique path segment can be obtained if the back-and-forth traverse starts from any atomic segment within it and its traverse MDS, therefore, we do not need to re-run the back-and-forth traverse if the MDS of one atomic segment has been included in the produced path segment. However, the resulting path segments may overlap (see Figure 8b), causing multiple machining passes when we take them as the tool path directly. Therefore, we need to resolve these overlaps while minimizing the number of resulting path segments. To achieve this, we apply a multi-label graph-cut algorithm [STC09].

We first associate a label to each path segment and then assign this label to all points that the segment contains. With the overlaps between path segments, an atomic segment may have multiple labels. We seek a label assignment l that minimizes the energy function:

$$E(l) = \sum_{a_i \in \mathcal{A}} D(l(a_i)) + \alpha \sum_{(a_i, a_j) \in \mathcal{A}} S(l(a_i), l(a_j)) \quad (3)$$

where D is the data term, S is the pairwise smooth term, \mathcal{A} is the atomic segments of a contour, α is a trade-off parameter between D and S ($\alpha = 2000$ in our implementation). The data term D estimates the cost of assigning a path segment (label) $l(a_i)$ to an atomic segment a_i . We define $Angle(l(a_i))$ to measure the angular magnitude

of the traverse MDS of atomic segment a_i in path segment $l(a_i)$. We define D as follows:

$$D(l(a_i)) = \begin{cases} 185 - Angle(l(a_i)), & \text{if } a_i \text{ in segment } l \\ \infty, & \text{otherwise} \end{cases} \quad (4)$$

The above formula tends to choose a larger traverse MDS when assign a_i into a segment. A larger MDS implies a safer machining direction and provides a broader range of tool direction options, which is conducive to finding a smoother direction in subsequent post-processing in Section 4.5. The smooth term S measures the cost of assigning path segments (labels) to two adjacent a_i and a_j . We define S as follows:

$$S(l(a_i), l(a_j)) = \begin{cases} 1, & \text{if } l(a_i) \neq l(a_j) \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

where $l(a_i)$ and $l(a_j)$ represent the path segments (labels) assigned to a_i and a_j , respectively.

Figure 8(c) shows two non-overlapping selected path segments from the input path segments of Figure 8(b). Compared to the greedy method shown in Figure 8(a), our method yields a smaller number of path segments. A quantitative comparison is provided to compare the performance of both methods in handling more complex scenarios, as shown in Figure 14.

4.4. Path segments connection

This step aims to connect all non-overlapping path segments on L_i into a single tool path TP_i by generating a machining sequence and transfer move paths between adjacent path segments. We seek to minimize the length of the transfer move paths to reduce machining time. This can be regarded as the classic Travelling Salesman Problem (TSP), which is NP-hard [HPR*13]. First, we construct a weighted complete graph G , where each node corresponds to the endpoints of the path segments. If two nodes belong to the same segment, we set the weight of the edge between them to 0. Otherwise, we set the weight to the length of the transfer move path. If the tool can move along a straight line between two endpoints without collision, the transfer move path can be the line segment connecting them. If there is a collision, we generate a transfer move path using additional paths, known as retract paths (see Figure 9b). Generating retract paths involves retracting the tool to a safe intermediate point, known as the retract point, which requires retracting a certain distance, known as the retract distance. Suppose e_k and e_{k+1} are endpoints of two path segments, with the middle directions of their traverse MDS being \vec{d}_k and \vec{d}_{k+1} , respectively. To generate such a retract path, first calculate the retract points $wd_k = e_k + W * \vec{d}_k$ and $wd_{k+1} = e_{k+1} + W * \vec{d}_{k+1}$, where W is the retract distance ($W = 35$ mm in our implementation). Thus, the retract path between e_k and e_{k+1} consists of three straight segments connecting the four points e_k, wd_k, wd_{k+1} and e_{k+1} .

To balance algorithm performance and efficiency, we propose two TSP solvers for G , depending on the number of nodes. Within 60 nodes, we run an exhaustive method, which starts from a randomly selected node, then traverses G using the Depth-First-Search (DFS) strategy. In the search tree, the path from the head node to the leaf node is taken to the TSP path. For graphs with more than 60 nodes,

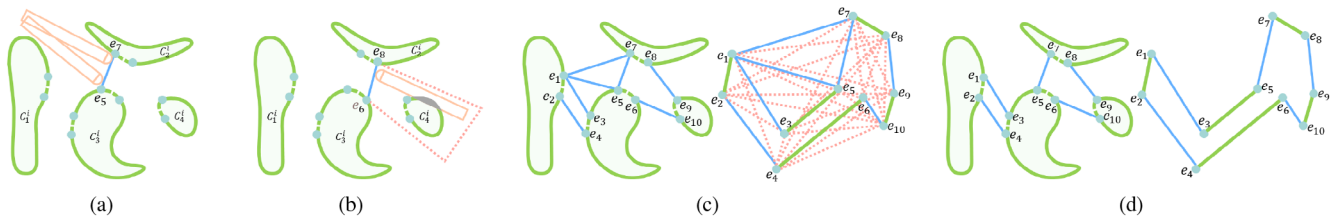


Figure 9: Path segments connection. (a) Two path segments can be connected by a straight transfer move (blue line). (b) Collision happens if two segments are connected by straight transfer moves (blue line). We avoid collisions by retracting the tool (red lines), which inevitably prolongs the path. (c) The complete graph built by our method. Blue lines indicate that the machining tool can move from one endpoint to another following a straight line. Red lines indicate that the connection needs a retract operation. Green lines connect nodes that are endpoints of the same path segment. (d) The resulting path is calculated by the exhaustive method to solve the TSP.

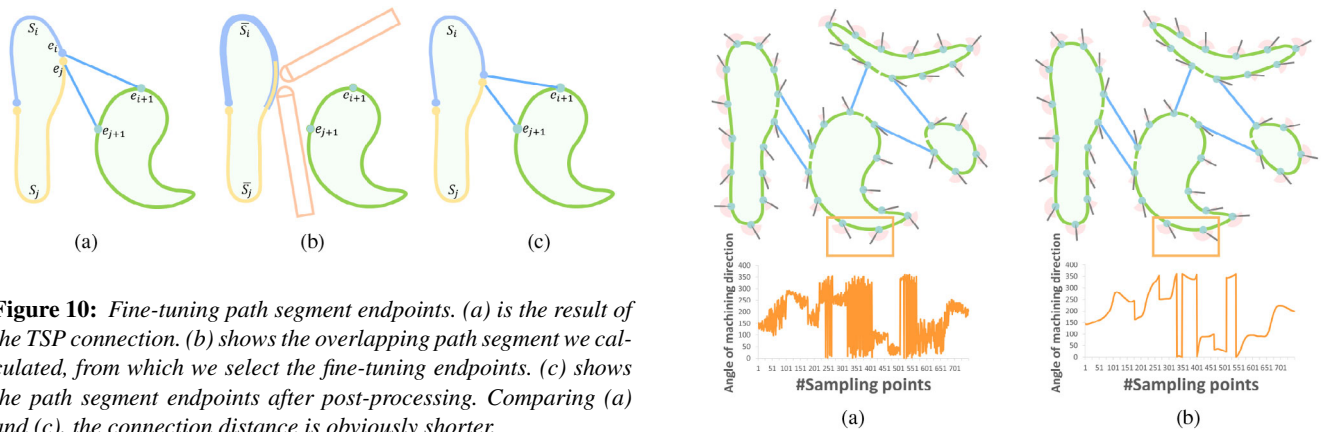


Figure 10: Fine-tuning path segment endpoints. (a) is the result of the TSP connection. (b) shows the overlapping path segment we calculated, from which we select the fine-tuning endpoints. (c) shows the path segment endpoints after post-processing. Comparing (a) and (c), the connection distance is obviously shorter.

we propose an iteratively greedy-based TSP solver. Starting from a randomly selected path segment, each iteration seeks the nearest path segments to the two endpoints of the produced TSP path. See Figure 9 for an example of the proposed path segments connection method. Additionally, we propose a heuristic rule to speed up the TSP solvers. Essentially, when an untraversed path's endpoint is included in the search, the other endpoint's node is set as the next traverse node.

4.5. Post-processing optimization

Up to this point, we have obtained a single tool path TP_i for layer L_i , represented as $TP_i = \{S_1^i, T_{1,2}^i, S_2^i, \dots, T_{m-1,m}^i, S_m^i\}$. This subsection presents a post-processing approach to fine-tune the endpoints of path segments locally and determine the machining direction for each atomic segment of TP_i .

Fine-tuning path segment endpoints. This step aims to slightly shorten the transfer move paths by adjusting the location of the endpoints of transfer move paths. These endpoints are indeed the endpoints of path segments. We apply the fine-tuning process to each endpoint of the path segments, one-by-one, along the sequence of TP_i . Each endpoint e_i of path segment S_i coincides with an endpoint e_j belonging to its adjacent path segment S_j , as shown in Figure 10(a). We first generate two path segments \bar{S}_i and \bar{S}_j starting from e_i and e_j and their traversal MDS, respectively. Then, identify

Figure 11: Smoothing of machining directions. The horizontal axis of the line chart represents the index of the atomic segments, and the vertical axis represents the angle of the machining directions. (a) The initial machining direction of each atomic segment is randomly selected in their MDR, with a notable and abrupt change of machining directions. (b) Results after the Laplacian smoothing for the machining directions, where the machining direction of adjacent atomic segments undergoes a smooth transition.

the overlapping path segment between \bar{S}_i and \bar{S}_j . Next, update e_i and e_j to the atomic segment in the overlapping path segment that is nearest to e_{i+1} and e_{j+1} ; see Figure 10(c). Here, e_{i+1} and e_{j+1} are the other endpoints of the transfer move paths connecting to S_i and S_j .

Smoothing of machining directions. For simultaneous four-axis machining, we need to determine the machining direction for each atomic segment of the final TSP path. We initialize the direction by randomly selecting a direction within the traverse MDS of each atomic segment (see Figure 11a). Then, we apply the Laplacian smoothing method iteratively for all adjacent points [SCOL*04]. For each Laplacian iteration, we update the machining direction of an atomic segment by first taking the average direction of its pre-order, post-order and the atomic segment itself, then setting the machining direction to updated one, as long as it belongs to the traverse MDS of the atomic segment. The smoothing continues until the sum

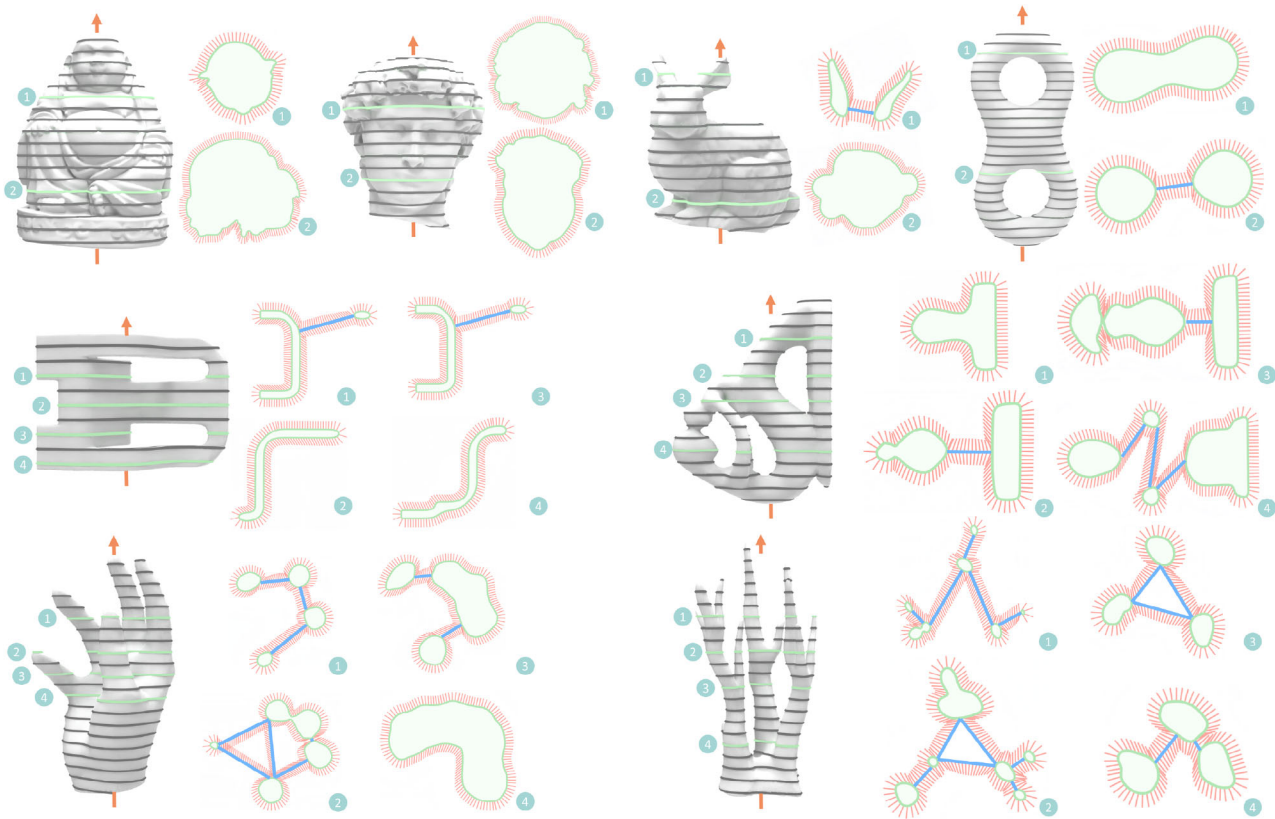


Figure 12: Tool paths results gallery generated by our method. The models are arranged in the order of Buddha, David, Bunny, Eight, Chair, Fertility, Hand and Coral. For each model, we first show the determined orientation of the model and the layer-by-layer slicing, where the two or four layers are picked out and presented. Red lines represent the tool directions of the atomic segments, and blue lines represent the transfer moves between path segments. Note that we apply a larger slicing thickness for visualization, and the intersection of tool directions (red lines) does not imply collisions as the tool moves linearly.

of angle change in all directions is $<1^\circ$ (Figure 11b). Finally, we sample atomic segments uniformly along all transfer move paths, using the same spacing of 0.2 mm, as referenced in Section 4.2. The machining direction of the new atomic segments is determined by a linear interpolation between the machining directions of its two endpoints. The line chart in Figure 11(b) shows the machining direction of each atomic segment, whose transition is significantly smoother than before optimization in Figure 11(a).

5. Results

This section demonstrates the tool path planning and production of 3D models with varying degrees of topological complexity. We conduct a thorough evaluation of our algorithm's efficiency, generality and effectiveness. We also compare our method with existing approaches. Finally, we discuss the limitations of our method.

5.1. Implementation and parameters

Our algorithm was implemented in C++, using CGAL [FP09] and Libhgp [Zha24] for geometric processing, Eigen [GJ*10] to solve linear equations and gco-v3.0 [VD15] for graph cut optimization.

We run the programme on a PC equipped with an Intel Core i7-13700 CPU running at 2.1 GHz and 32GB of memory. In determining object orientation, we sample 2000 candidate orientations in the Gaussian sphere. We set the slicing thickness to 0.2 mm and the sampling interval to 0.2 mm to uniformly re-sample atomic segments on each contour. In collision detection, we sampled 72 machining directions uniformly, sampling every 5° . In path segment connection, we set retract distance to 35 mm. Among these hyper-parameters, the slicing thickness would be the most crucial as it directly determines the number of layers. A smaller thickness means more layers, increasing the fabrication time, but improving the surface quality.

5.2. Simultaneous four-axis toolpath

We assess our algorithm's efficiency and path planning results in this subsection. Figure 12 depicts path planning for 8 models, each with two or four visualized tool paths. Figure 13 shows the surface scallop height analysis results of three models simulated machining in Siemens NX, based on the paths generated by our algorithm. Table 1 provides statistical data, while Table 2 details the algorithm running time for each step. We also conducted a physical

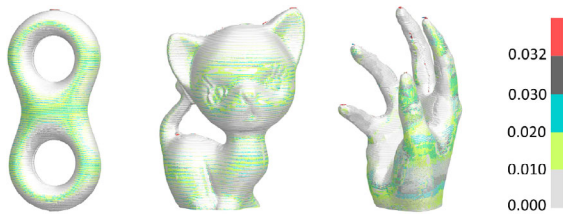


Figure 13: Surface scallop height analysis results. This figure shows surface scallop height analysis results of three models. They are Eight, Kitten and Hand. From the surface scallop height analysis results, the surface scallop height is less than the maximum scallop height of 0.033 mm.

Table 1: Statistics of the results. R_s is the surface-area-to-volume ratio. H is the height (millimetres) of the model along the object orientation. The number of slices is determined by dividing the model height by the layer thickness. $\#S$ is the total number of atomic segments across all layers. $\#C$ is the average number of contours per layer. $\#P_C$ and $\#P$ are the average number of input and output atomic segments to graph-cut, separately. E is the average number of endpoints per layer to TSP. D is the average length of final tool path after post-processing. A is the average transition of machining direction of adjacent atomic segments after post-processing. T is the total fabrication time (minutes).

Model	R_s	4.1		4.2		4.3		4.4		4.5	
		H	$\#S$	$\#C$	$\#P_C$	$\#P$	E	D	A	T	
Kitten	0.29	46	7.3E4	1.22	1.5	1.2	2.4	70.8	1.4	62.0	
Buddha	0.26	46	9.1E4	1.02	1.6	1.4	2.8	83.8	1.1	70.0	
David	0.25	38	7.8E4	1.15	2.5	1.8	3.6	95.2	1.2	54.2	
Bunny	0.34	36	5.4E4	1.18	1.5	1.3	2.6	75.2	1.5	60.0	
Eight	0.42	60	7.9E4	1.45	1.8	1.4	2.8	57.5	1.7	74.0	
Chair	1.37	23	4.6E4	1.34	1.8	1.5	3.0	101.9	1.3	45.5	
Fertility	0.61	43	6.9E4	2.29	5.5	3.2	6.4	79.3	2.0	46.3	
Hand	0.40	53	8.2E4	2.08	5.4	2.8	5.6	88.9	1.9	80.0	
Coral	0.90	55	5.9E4	3.57	9.4	5.5	11.0	88.7	3.9	117.0	

Table 2: Programme running time of each step (seconds). Ori is determining orientation of the model. Acc is accessibility analysis. Seg is path segments decomposition. TSP is path segments connection by solving a TSP. Con is fine-tuning path segment endpoints. Smo is smoothing of machining directions. Tot is the total time.

Model	4.1		4.2		4.3		4.4		4.5	
	Ori	Acc	Seg	TSP	Con	Smo	Tot			
Kitten	39.7	346	0.2	0.1	2.9	1.0	389.9			
Buddha	89.1	447	0.3	0.2	<0.1	2.3	538.9			
David	43.6	416	0.3	0.1	<0.1	2.1	462.1			
Bunny	49.7	246	0.1	0.1	0.6	0.9	297.5			
Eight	43.5	506	0.3	0.2	7.4	1.0	558.4			
Chair	13.2	613	0.2	<0.1	5.1	0.6	632.1			
Fertility	9.1	710	0.4	0.2	4.6	1.1	725.4			
Hand	18.0	699	0.4	0.2	11.2	1.3	730.1			
Coral	16.4	584	0.5	0.4	9.6	1.4	612.3			

experiment to validate the proposed path segment decomposition method and post-processing optimization.

5.2.1. Evaluation of path planning

As shown in Figure 12, our algorithm generates tool paths for both single and complex multi-contours within a slicing layer. All paths show excellent directional continuity and geometric continuity, where the tool directions are indicated by the smooth red lines, and transfer move paths are also reduced, as shown by the blue lines. Most contours have been decomposed into multiple path segments, which are processed by the machining tool in an interleaved order. This implies that the tool moves between different contours, and a single contour can be visited multiple times, as in the fourth tool path of the Coral model. In Table 1, it is evident that the average number of produced path segments per layer increases alongside the average number of contours (refer to $\#P_C$ and $\#C$ for *Coral* and *Hand*). This is likely due to the fact that an increase in the quantity of contours reduces the machinable direction range for sampling points. As a result, more path segments are needed to process each contour.

5.2.2. Algorithm efficiency

Our algorithm runs about 9 min on average for each model in our experiments; see Table 2. The accessibility analysis is the most time-consuming since we sample a large number of candidate directions to detect the cutter accessibility to each sampling atomic segment. Hence, the running time of accessibility analysis is determined by both the number of sampling points ($\#S$) and contours ($\#C$) of all layers, indicated in Table 1. Aside from object orientation determination and accessibility analysis, the remaining steps of our algorithm are quite fast.

5.2.3. Greedy versus graph-cut

To demonstrate the effectiveness of our path segments decomposition strategy (Section 4.3), we propose a comparison between the greedy method and the graph-cut method using the *Coral* model. As shown in Figure 14, the graph-cut method results in fewer path segments at each slicing layer of the *Coral* model. Moreover, the advantage of the graph-cut method is more noticeable when both methods produce a large number of path segments. We record the average number of path segments, the average length of the tool path per layer, and the fabrication time for both methods. The three values for the greedy method (graph-cut method) are 6.4 (5.5), 149.2 mm (88.7 mm) and 129 min (117 min), respectively.

5.2.4. Post-processing optimization

To validate the two post-processing methods in Section 4.5, we compare surface quality and fabrication time without and with these methods. We use the *Eight* model for this comparison; see Figure 15.

Figure 15(a) shows the fabrication result when the machining directions are not smoothed by post-processing. These directions are randomly selected within the traverse MDS of each atomic

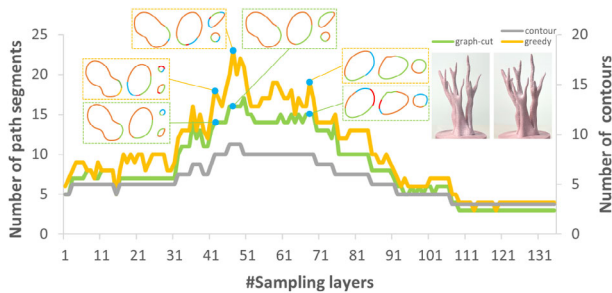


Figure 14: Comparison experiment between graph-cut and greedy method. Comparison between the graph-cut and greedy methods using the Coral model. Graph-cut method (green curve) consistently produces fewer path segments than the greedy method (yellow curve). Additionally, it is observed that the number of path segments produced increases with the number of contours (grey curve). Three layers are selected to demonstrate the path segment decomposition results.

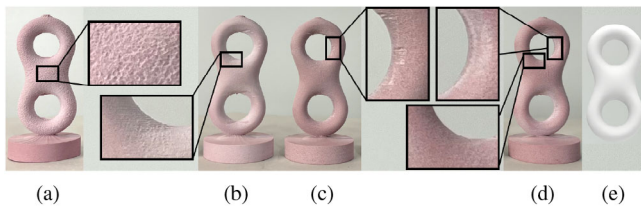


Figure 15: Comparison experiments of post-processing. (a) Fabricated Eight model using the tool path without smoothing of machining directions. The abrupt tool direction transition causes a large number of pits on the surface, which seriously reduce the surface quality. (b) Fabricated results using the heuristic method that first selects the normal direction of each atomic segment, or, if inaccessible, the closest direction within its traverse MDS. The heuristic method makes some overcut artefacts. (c) Fabricated result using the tool path without fine-tuning path segment endpoints. (d) Fabricated result using the tool path with two post-processing strategies. It demonstrates better surface quality. (e) The rendered view of Eight.

segment. We can see a large number of defects on the model surface. A heuristic method to determine the tool directions is compared; see Figure 15(b). The heuristic method first selects the normal direction of each atomic segment, or, if inaccessible, the closest direction within its traverse MDS. Figure 15(d) shows the fabrication result with post-processing smoothing of the machining directions. The average angle variation per layer for (a), (b) and (d) is 2741° , 636° and 451° , respectively. Since tuning the rotary axis during the machining takes time, Figure 15(d) takes 74 min for fabrication, which is much shorter than the 372 min of Figure 15(a) and the 95 min of Figure 15(b). Figure 15(c) shows the fabrication result of the *Eight* model without fine-tuning the endpoints of transfer moves. The surface quality of the machining result is slightly lower compared to what is shown in Figure 15(d), which includes fine-tuning. For the average length of the tool path per layer, (d) is 57.5 mm, which is slightly better than 57.7 mm of (c).



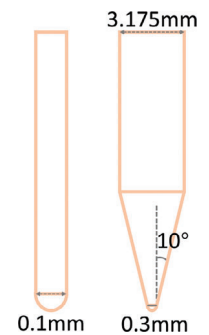
Figure 16: Fabricated results with detailed close-ups. This figure shows close-up photos of four fabrication results. They are Eight, Coral, Buddha and Fertility.

5.3. Physical evaluation

This sub-section initially introduces the setup of the fabrication experiment. Next, we evaluate the results in terms of efficiency and surface quality. We then compare to two CAM systems: the Luban system provided by Snapmaker and Fusion 360 by Autodesk, focusing on their simultaneous strategies. For a live demonstration of the manufacturing process, please refer to our supplementary video.

5.3.1. Setup of fabrication experiment

In our experiments, all the fabrication results are produced by the Snapmaker 2.0 A350T, which has a fabrication space of $350 \times 320 \times 330$ mm and a spindle speed of 15,000 r/min. We use cylindrical machinable resin boards as machining stocks, with a height of 70 mm and a radius of 17.5 mm. Except for Figure 19, which uses a 1.0-mm diameter ball-end mill, the default milling tool is a two-edge straight groove pointed tool. Specifically, the length of the carving knife is 24 mm, the diameter of the tip is 0.3 mm, the diameter of the shank is 3.175 mm and the total length of the tool is 50 mm. To run our generated tool path on Snapmaker, we export the path to a common 'gcode' file [LAYK21] with the feed rate of 800 mm/min. In the fabrication setup, our tool is longer than the machining stocks (24 mm for the carving tool vs. 17.5 mm for the stock's radius), and the machinable resin boards have low hardness. Consequently, we chose to perform finishing machining directly, without rough machining. However, if the tool were shorter or the material hardness were higher, a rough machining stage would be necessary. This issue could be addressed using existing CAM systems, such as the positional rough machining tool path in Fusion 360 [Wor23].



5.3.2. Evaluation of fabrication results

Figure 16 shows close-up, detailed views of the fabrication surfaces of four models in Figure 17. These models were machined using our simultaneous four-axis tool path, which often results in excellent surface quality. For topologically simple models such as *Buddha* and *David*, there are no visible boundary artefacts. However, for the complex model such as *Fertility* and *Coral*, tiny boundary artefacts appear on the surface due to the discontinuity of the machining



Figure 17: Fabrication results gallery. The upper portion of the figure displays a rendered view of the corresponding input 3D models. The lower portion showcases a collective image of the fabrication results.

direction at the intersections of path segments (see the inset in the figure of *Coral*).

The surface areas where the normal direction is nearly parallel to the rotation axis will be omitted by slicing, resulting in un-machinable sections. Although we have mitigated this issue by optimizing object orientation, these areas still persist unavoidably (see the head of *Buddha*). In particular, some sampling points have void MDR, *i.e.* are un-machinable points (see the inset in the figure of *Fertility*). Our algorithm simply skips these points to ensure that the model is overall successfully manufactured. When it comes to fabrication time, the length of the final tool path per layer and the height of the model are the determining factors, as shown in Table 1. In our experiment, the average time taken to fabricate all the models is 68 min.

5.3.3. Comparison with CAM systems

As noted previously, the simultaneous strategy for four-axis CNC machines remains an open research area, with only a few solutions available in industrial CAM systems. Here, we compare the Luban software provided by Snapmaker [Sna23] and Fusion 360 [Wor23], using the same parameters as in Section 5.1. We use the *Hand* model for this comparison (see Figure 18). Figure 18(a) shows the fabrication result using Luban software. The red circles highlight undercut and undercut artefacts, indicating that for multiple contours in one layer, Luban software may not accurately calculate the machinable direction for each surface point. Figure 18(b) shows the result from Fusion 360, which also clearly exhibits an undercut artefact. Moreover, we observe that the tool path from Fusion 360 includes numerous instances of idle rotational movements during machining, leading to significantly prolonged machining times. Our result shown in Figure 18(c) is significantly more efficient than both Luban and Fusion 360, taking 80 min compared to their 115 and 320 min, respectively.

5.3.4. Comparison with positional strategy

We compare our method to the positional four-axis machining introduced by Nuvoli *et al.* [NTM*21]. We use the *Kitten* model for this comparison (see Figure 19). To keep a single variable in the compar-

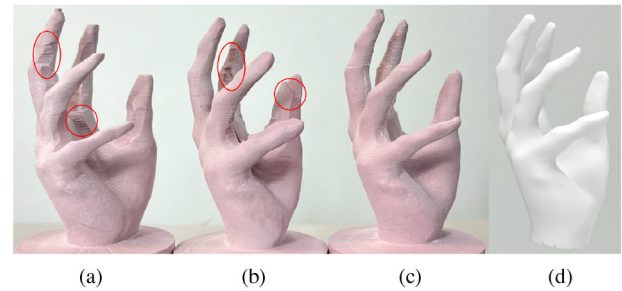


Figure 18: Fabrication results comparison with CAM systems. (a) Fabrication result of the *Hand* model using the tool paths generated by the Luban CAM software, which takes 115 min for manufacturing. The results exhibit both undercut and over-cut (red circles). The little finger is much thinner than it should be, due to being raised by the overcut. (b) Fabrication result of the Fusion 360 CAM software, which takes 320 min for manufacturing. The results show a lot of undercut, where the palm part is much thicker than it should be. (c) Fabrication result of our method, which takes 80 min for manufacturing. (d) The rendered view of *Hand*.

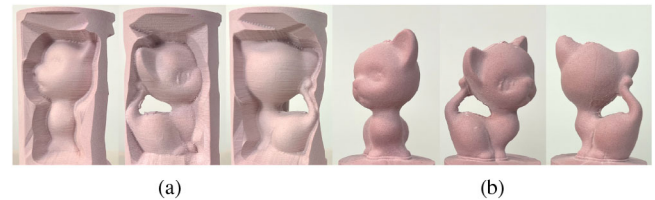


Figure 19: Comparison experiment with positional strategy. (a) is the fabrication result of Nuvoli *et al.* [NTM*21]. It was machined from three different directions, each producing a height-field patch. (b) is the manufacturing result of our method, as depicted from three different views.

ison as consistent as possible, both fabrication results use the same parameters as in Section 5.1, and we apply a Zigzag pattern to generate the fine machining tool path for the positional strategy. For both fabrication results, rough machining utilizes a ball-end mill with a diameter of 3.175 mm, while finish machining employs a ball-end mill with a diameter of 1.0 mm. Figure 19(a) shows the fabrication results from Nuvoli *et al.* [NTM*21], which exhibit many undercut parts and require additional manual intervention to remove. Moreover, its surface quality is inferior to that of our method shown in Figure 19(b), which is characterized by a less pronounced staircase effect. Note that Nuvoli *et al.* [NTM*21] treat the top as an independent machining patch, which requires separate machining beyond the capabilities of four-axis CNC machining. Therefore, we can only report the machining time for its side surfaces: 41 min for Figure 19(a) and 62 min for Figure 19(b). Taking into account the top patch machining and the manual removal of undercuts, our method would be comparable to that of Nuvoli *et al.* [NTM*21].

5.4. Limitation and discussion

Our pipeline enables the manufacturing of complex 3D free-form shapes from a single solid stock using the simultaneous machining

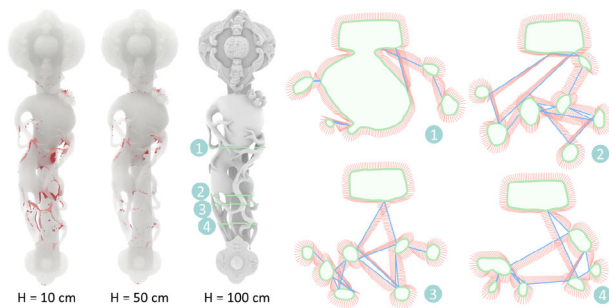


Figure 20: Tool path results of Ruyi model. Left: Three Ruyi models in different heights, where the red areas indicate the non-visible areas. Right: Four layers are selected to show the tool paths generated by our method.

strategy of four-axis CNC machines. To our knowledge, we are the first to propose an end-to-end pipeline that fully explores the potential of the simultaneous machining strategy for four-axis CNC machines. The key limitations of our current technique are three-fold: the intrinsic constraints imposed by four-axis CNC machines, the limited search space imposed by layered SM and the lack of guarantee for global optimality.

5.4.1. Intrinsic fabrication limitation

The first limitation, related to intrinsic constraints, was well introduced in Nuvoli *et al.* [NTM*21]. As clarified in our introduction, four-axis CNC machining is a cost-effective fabrication technique that bridges the gap between three-axis CNC machines and the advanced capabilities of five-axis CNC machines. This technique cannot fabricate arbitrary complex shapes. According to Nuvoli *et al.* [NTM*21], there is no formal definition of shapes that can be manufactured in a single block using four-axis CNC machines. This paper also does not address this problem, as it falls outside the scope of our research. Therefore, our current method cannot handle non-visible features of target shapes, such as the Ruyi model with a height of 10 cm shown in Figure 20, where the red areas cannot be reached by our default fabrication settings. However, as the model's size increases, the non-visible areas will decrease until there are no non-visible areas at a height of 100 cm. Our algorithm can effectively handle this and generate a simultaneous four-axis machining tool path, as shown in Figure 20.

5.4.2. Limited search space

Our technique simplifies the problem of simultaneous machining by reducing the 3D tool path issue to a 2D planning problem. We achieve this with a layer-based fabrication approach, which effectively simplifies the problem. However, layer-based milling limits the possibility of achieving an optimal solution for tool path planning in simultaneous machining strategies. A more effective tool path planning approach may involve combining region-decomposition with layer-based milling methods to produce a tool path that is as continuous as possible across the entire surface. Additionally, while the current solution is effective in optimizing the

tool path within each slicing layer, it does not guarantee global optimality. Our algorithm decomposes each sliced contour into minimal fabricable segments using a multi-label graph-cut optimization method [STC09], but it does not guarantee global optimality. However, the graph-cut optimizer consistently produces reasonable solutions in our experiments. Additionally, our approach only considers ball-end mills and straight groove pointed tools, without accounting for other types of tools such as toroidal cutters; although our tools have a conical part, four-axis CNC machines lack the degrees of freedom needed to position a conical tool effectively for flank milling.

6. Conclusion and Future Work

In this paper, we propose the first end-to-end computational framework for four-axis simultaneous machining strategies to fabricate complex shapes featuring high-genus or numerous branch structures. Our framework includes a tool path generation process that optimizes the continuity of tool direction and the machining sequence. The main advantage of our simultaneous machining strategy is its ability to significantly reduce seam artefacts, which are difficult to avoid when using positional machining strategies.

As discussed in the results section, the main efficiency bottleneck in our algorithm is accessibility analysis. We plan to accelerate this step using CUDA parallelization, adaptive spatial partitions of the bounding volume hierarchy (BVH) [LAM06] or the FFT-based collision metric [CRCM23]. There are several areas for future research. First, exploring the effectiveness of adaptive thickness slicing methods [XGD*18], curved slicing layers [ZFH*22] and spiral-like slicing layers [ZXZL23] in additive manufacturing could enhance the efficiency and surface quality of the simultaneous machining strategy for four-axis CNC machines. Second, investigating a method that integrates fabricable segment decomposition and TSP linking stages into a single graph-cut-based process would be valuable. Third, exploring a hybrid machining strategy that leverages the advantages of both positional and simultaneous machining strategies is a promising research direction. Fourth, considering additional physical factors, such as machining stability during SM, would be beneficial. Fifth, a more in-depth study of the object orientation optimization problem is needed. Lastly, applying our method to existing CAM systems for four-axis CNC machines is another avenue for future work.

It is crucial to ensure manufacturability when evaluating the fabrication capacity of a four-axis CNC machine. This leads to two research directions. First, there is a need to explore topology optimization techniques that take into account specific constraints related to manufacturability during the modelling process. Second, addressing the problem of transforming un-fabricable shapes into fabricable ones for four-axis CNC machines, while minimizing shape variations, presents an intriguing challenge.

Acknowledgements

We thank all reviewers for their valuable comments and constructive suggestions. Thanks for the models provided by Thingiverse and GrabCAD. This work is supported in part by grants from

the National Key R&D Programme of China (2022YFB3303200), the National Natural Science Foundation of China (U23A20312) and the Guangdong Basic and Applied Basic Research Foundation (2023B1515120026). The authors thank Zhihao Zhang and Qibing Wu for their practical help with proofreading.

References

- [AMG*19] ALDERIGHI T., MALOMO L., GIORGI D., BICKEL B., CIGNONI P., PIETRONI N.: Volume-aware design of composite molds. *ACM Transactions on Graphics* 38 (2019), 1–12.
- [BBR*21] BARTON M., BIZZARRI M., RIST F., SLIUSARENKO O., POTTMANN H.: Geometry and tool motion planning for curvature adapted CNC machining. *ACM Transactions on Graphics* 40, 4 (2021), 1–16.
- [CJ12] CHOI B. K., JERARD R. B.: *Sculptured Surface Machining: Theory and Applications*. Springer Science & Business Media, 2012. Kluw Academic Publishers, 1998.
- [CRCM23] CUI Q., RONG V., CHEN D., MATUSIK W.: Dense, interlocking-free and scalable spectral packing of generic 3D objects. *ACM Transactions on Graphics (TOG)* 42, 4 (2023), 1–14.
- [CÜ10] CAN A., ÜNÜVAR A.: A novel iso-scallop tool-path generation for efficient five-axis machining of free-form surfaces. *The International Journal of Advanced Manufacturing Technology* 51 (2010), 1083–1098.
- [DJ04] DING S., JIANG R.: Tool path generation for 4-axis contour EDM rough machining. *International Journal of Machine Tools and Manufacture* 44, 14 (2004), 1493–1502.
- [EE18] EZAIR B., ELBER G.: Automatic generation of globally assured collision free orientations for 5-axis ball-end tool-paths. *Computer-Aided Design* 102 (2018), 171–181.
- [FCM*18] FANNI F. A., CHERCHI G., MUNTONI A., TOLA A., SCATENI R.: Fabrication oriented shape decomposition using polycube mapping. *Computers & Graphics* 77 (2018), 183–193.
- [FP09] FABRI A., PION S.: CGAL: The computational geometry algorithms library. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (2009), pp. 538–539.
- [FWJ06] FRANK M. C., WYSK R. A., JOSHI S. B.: Determining setup orientations from the visibility of slice geometry for rapid computer numerically controlled machining. *Journal of Manufacturing Science and Engineering, Transactions of the ASME*, 128, 1 228–238.
- [Gii23] Giiresearch: 4-axis CNC machining center market forecasts to 2030. <https://www.giiresearch.com/report/smr1371881-axis-cnc-machining-center-market-forecasts-global.html> (2023). Accessed 29 July 2024.
- [GJ*10] GUENNEBAUD G., JACOB B.: Eigen v3. <http://eigen.tuxfamily.org> (2010). Accessed 29 June 2023.
- [HBA13] HILDEBRAND K., BICKEL B., ALEXA M.: Orthogonal slicing for additive manufacturing. *Computers & Graphics* 37, 6 (2013), 669–675.
- [HMA15] HERHOLZ P., MATUSIK W., ALEXA M.: Approximating free-form geometry with height fields for manufacturing. *Computer Graphics Forum* 34 (2015), 239–251.
- [HPR*13] HOFFMAN K. L., PADBERG M., RINALDI G.: Traveling salesman problem. In *Encyclopedia of Operations Research and Management Science*. Springer, Boston, MA (2013), vol. 1, pp. 1573–1578.
- [JLZ*21] JI S., LEI L., ZHAO J., LU X., GAO H.: An adaptive real-time nurbs curve interpolation for 4-axis polishing machine tool. *Robotics and Computer-Integrated Manufacturing* 67 (2021), 102025.
- [Jos15] JOSHI A.: Computer aided process planning for multi-axis CNC machining using feature free polygonal cad models. Digital Repository, 2015.
- [LAM06] LARSSON T., AKENINE-MÖLLER T.: A dynamic bounding volume hierarchy for generalized collision detection. *Computers & Graphics* 30, 3 (2006), 450–459.
- [LAYK21] LATIF K., ADAM A., YUSOF Y., KADIR A. Z. A.: A review of G code, STEP, STEP-NC, and open architecture control technologies based embedded CNC systems. *The International Journal of Advanced Manufacturing Technology* 114 (2021), 2549–2566.
- [Lee03] LEE E.: Contour offset approach to spiral toolpath generation with constant scallop height. *Computer-Aided Design* 35, 6 (2003), 511–518.
- [LKL21] LIANG F., KANG C., LU Z., FANG F.: Iso-scallop tool path planning for triangular mesh surfaces in multi-axis machining. *Robotics and Computer-Integrated Manufacturing* 72 (2021), 102206.
- [LXG10] LASEMI A., XUE D., GU P.: Recent development in CNC machining of freeform surfaces: A state-of-the-art review. *Computer-Aided Design* 42, 7 (2010), 641–654.
- [MLS*18] MUNTONI A., LIVESU M., SCATENI R., SHEFFER A., PANOZZO D.: Axis-aligned height-field block decomposition of 3D shapes. *ACM Transactions on Graphics (TOG)* 37, 5 (2018), 1–15.
- [MPE17] MACHCHHAR J., PLAKHOTNIK D., ELBER G.: Precise algebraic-based swept volumes for arbitrary free-form shaped tools towards multi-axis CNC machining verification. *Computer-Aided Design* 90 (2017), 48–58.
- [MSJ*22] MA H.-y., SHEN L.-y., JIANG X., ZOU Q., YUAN C.-m.: A survey of path planning and feedrate interpolation in computer numerical control. *Journal of Graphics* 43, 6 (2022), 967–986.
- [NTM*21] NUOLI S., TOLA A., MUNTONI A., PIETRONI N., GOBBETTI E., SCATENI R.: Automatic surface segmentation for

- seamless fabrication using 4-axis milling machines. *Computer Graphics Forum* 40 (2021), 191–203.
- [PL14] PLAKHOTNIK D., LAUWERS B.: Graph-based optimization of five-axis machine tool movements by varying tool orientation. *The International Journal of Advanced Manufacturing Technology* 74 (2014), 307–318.
- [RSG09] REN F., SUN Y., GUO D.: Combined reparameterization-based spiral toolpath generation for five-axis sculptured surface machining. *The International Journal of Advanced Manufacturing Technology* 40 (2009), 760–768.
- [SCOL*04] SORKINE O., COHEN-OR D., LIPMAN Y., ALEXA M., RÖSSL C., SEIDEL H.-P.: Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing* (2004), pp. 175–184.
- [Sie16] Siemens: NX software. <https://plm.sw.siemens.com/en-US/nx/> (2016). Accessed 1 May 2023.
- [SJP06] SWINBANK R., JAMES PURSER R.: Fibonacci grids: A novel approach to global modelling. *Quarterly Journal of the Royal Meteorological Society: A Journal of the Atmospheric Sciences, Applied Meteorology and Physical Oceanography* 132, 619 (2006), 1769–1793.
- [SKM*22] SATHISH K., KUMAR S. S., MAGAL R. T., SELVARAJ V., NARASIMHARAJ V., KARTHIKEYAN R., SABARINATHAN G., TIWARI M., KASSA A. E.: A comparative study on subtractive manufacturing and additive manufacturing. *Advances in Materials Science and Engineering* 2022.1 (2022), 6892641 2.
- [Sna23] Snapmaker: Luban v4.9.1. <https://snapmaker.cn/snapmaker-luban> (2023). Accessed 1 July 2023.
- [STC09] SCHMIDT F. R., TOPPE E., CREMERS D.: Efficient planar graph cuts with applications in computer vision. In *2009 IEEE Conference on Computer Vision and Pattern Recognition* (2009), pp. 351–356. <https://doi.org/10.1109/CVPR.2009.5206863>.
- [VD15] VEKSLER O., DELONG A.: gco-v3.0. <https://github.com/nsubtil/gco-v3.0> (2015). Accessed 1 September 2022.
- [Wic24] Wicz: 5-axis CNC machining centers market size and share analysis 2024. <https://www.wicz.com/story/50969181/5-axis-cnc-machining-centers-market-size-and-share-analysis-2024-business-growth-emerging-trends-and-regional-forecast-to-2032> (2024). Accessed 29 July 2024.
- [Wor23] WORKS S. M.: Fusion 360. <https://www.nyccnc.com/simultaneous-4th-axis/> (2023). Accessed 2 July 2023.
- [XGD*18] XU J., GU X., DING D., PAN Z., CHEN K.: A review of slicing methods for directed energy deposition based additive manufacturing. *Rapid Prototyping Journal* 24, 6 (2018), 1012–1025.
- [YJJ*22] YUWEN S., JINJIE J., JINTING X., MANSEN C., JINBO N.: Path, feedrate and trajectory planning for free-form surface machining: A state-of-the-art review. *Chinese Journal of Aeronautics* 35, 8 (2022), 12–29.
- [ZCW16] ZAHID M. O., CASE K., WATTS D.: Cutting orientations for non-complex parts in 4th axis machining. In *IOP Conference Series: Materials Science and Engineering* (2016), vol. 114, IOP Publishing, pp. 012013.
- [ZFH*22] ZHANG T., FANG G., HUANG Y., DUTTA N., LEFEBVRE S., KILIC Z. M., WANG C. C.: S3-Slicer: A general slicing framework for multi-axis 3D printing. *ACM Transactions on Graphics (TOG)* 41, 6 (2022), 1–15.
- [ZGH*16] ZHAO H., GU F., HUANG Q.-X., GARCIA J., CHEN Y., TU C., BENES B., ZHANG H., COHEN-OR D., CHEN B.: Connected fermat spirals for layered fabrication. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 1–10.
- [Zha24] ZHAO H.: Libhgp. <https://github.com/haisenzhao/libhgp> (2024). Accessed 1 May 2023.
- [ZRZ23] ZHANG W., REN J., ZHOU J.: Noninterference tool orientations and maximum taper angles of conical cutters in 4-axis milling of complex channel parts. *The International Journal of Advanced Manufacturing Technology* 127 (2023), 1–29.
- [ZXZL23] ZHONG F., XU Y., ZHAO H., LU L.: As-continuous-as-possible extrusion-based fabrication of surface models. *ACM Transactions on Graphics* 42, 3 (2023), 1–16.
- [ZZX*18] ZHAO H., ZHANG H., XIN S., DENG Y., TU C., WANG W., COHEN-OR D., CHEN B.: DSCarver: Decompose-and-spiral-carve for subtractive manufacturing. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–14.

Supporting Information

Additional supporting information may be found online in the Supporting Information section at the end of the article.

Supplemental Movie 1

Supporting Information