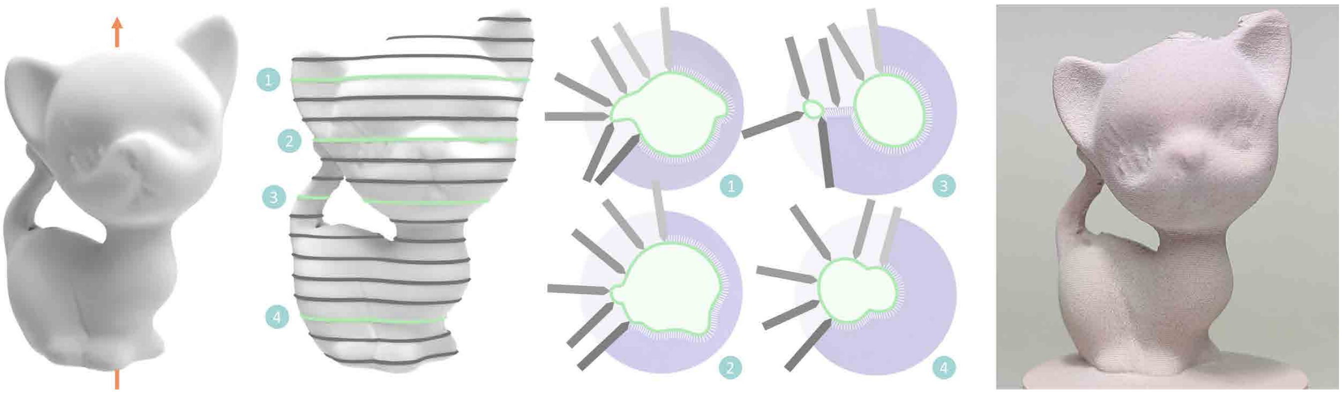
Volume 0 (1981), Number 0 pp. 1–15

Continuous Tool Path Optimization for Simultaneous Four-Axis Subtractive Manufacturing

Zhenmin Zhang[1] and Zihan Shi[1] and Fanchao Zhong[1] and Kun Zhang[1] and Wenjing Zhang[1] and Jianwei Guo[2] and Changhe Tu[1] and Haisen Zhao[1]†

[1]Shandong University, School of Computer Science and Technology, China
[2]MAIS, Institute of Automation, Chinese Academy of Sciences, China



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

Figure 1: ***Our algorithmic pipeline in brief.*** *This study proposes a general computational framework for simultaneous four-axis computerized numerical control (CNC) machining to minimize variations in the direction of the tool during continuous machining, and to ensure a collision-free process of fabrication. This figure shows the Kitten model (a). We uniformly slice it along the rotational axis after determining its orientation (b). We then optimize the tool path in each layer to generate a simultaneous four-axis path for it. The aim is to maximize geometric continuity and minimize variations along the directions of machining (c). (d) The physical outcome of fabrication of the Kitten model. It shows that the proposed framework can be used for simultaneous four-axis subtractive manufacturing.*

Abstract

Simultaneous four-axis machining involves a cutter that moves along four degrees of freedom as it carves the given object. This strategy provides higher-quality surface finishing than positional machining, but has not been adequately investigated in the relevant research. In this study, we propose the first end-to-end computational framework to optimize the tool path to fabricate complex models by using simultaneous four-axis subtractive manufacturing. Our technique involves first slicing the input 3D model into uniformly distributed 2D layers. We then analyze the accessibility of each intersected contour of each sliced layer, and apply over-segmentation and a bottom-up connecting process to generate the minimal number of fabricable segments. Finally, we propose post-processing techniques to further optimize the direction of the tool and the path of transfer between segments. The results of physical experiments on nine models verified the significant improvements brought about by our method in both the quality and efficiency of fabrication, which were superior to the results obtained when using the positional strategy and two simultaneous tool paths generated by industry-standard CAM systems.

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

† Corresponding author

(a) (b) (c)

Figure 2: ***Typical products of four-axis machining.*** *Four-axis subtractive manufacturing is widely used for the metal machining of revolving solids (a), wood crafts (b), and high-genus prototypes (c).*

Figure 3: ***Demonstration of the positional machining strategy.*** *The figure was taken from [NTM*21]. It shows the fabrication of the Kitten model by carving it based on a set of height field patches and subjecting it to manual post-processing.*

1. Introduction

CNC subtractive manufacturing (SM) is a cornerstone of modern industry that has evolved continually to meet the growing demands for precision and complexity in the fabrication of various components and products [LXG10; SKM*22]. In this context, the use of four-axis machining has emerged as a crucial and cost-effective technique by bridging the gap between the accessibility of three-axis CNC machines and the intricate capabilities of five-axis CNC machines. The importance of four-axis machining lies in its ability to strike a delicate balance between the complexity of the shape of the object and the accessibility of the machine. In contrast to three-axis machining, the additional axis in four-axis CNC machines enables the creation of intricate and multi-faceted designs, such as in case of side drilling and the drilling of the surface of a cylinder. Moreover, it improves productivity by allowing multiple operations to be performed in a single setup. Four-axis CNC machines are much more cost effective than five-axis machines, and can be used to obtain complex geometries such that they ensure greater accessibility in manufacturing†. Four-axis machining thus has a wide range of applications, especially in the aerospace, automotive, and medical industries in which 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. Its rotational capability enables the machine to perform complex and versatile machining operations to create intricate and precise designs. The interested reader can see Figure 4 for the setup of the machine. To perform subtractive manufacturing by using a four-axis CNC machine, we need to determine the direction and movement of the tool on the fine-machining surface of the target 3D shape. The movement of the cutter refers to the sequence of machining, which reflects the next machining points after having carved the current one. Four-axis machining represents a critical decision point with two primary strategies: positional and simultaneous machining strategies.

The *positional strategy* (also known as the positional fourth axis, or 3+1 machining) maintains a fixed direction of the tool during cutting by using three degrees of freedom of translation. The remaining rotational degree of freedom is used to move the cutter between the cutting materials 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 incurring any collision. A path planning process for the tool is then used to determine the movement of the cutter to carve each patch.

The key benefit of the positional machining strategy lies in its simplicity, whereby it determines the direction and movement of the tool in two independent computational stages, as has been noted in [NTM*21]. However, its performance is affected by the presence of boundary artifacts between neighboring patches that arise from discontinuous tool paths carved from different directions. As shown in Figure 3, the boundary artifacts require additional manual post-processing work to achieve the desired surface finish. By contrast, it is anticipated that the simultaneous machining strategy can address these concerns and significantly reduce the number of boundary artifacts.

The *simultaneous strategy* (also called true four-axis machining) involves the cutter simultaneously moving along all four degrees of freedom during carving. This shows that both the direction of the tool (rotational degree of freedom) and the movement of the cutter (three translational degrees of freedom) should be simultaneously determined while planning the tool path. The cutting tool gradually changes its direction throughout the machining process in the simultaneous strategy. This lends this strategy its primary advantage of a high-quality machined surface without requiring the post-processing of the boundary artifacts that arise in the positional strategy. These artifacts arise from discontinuous paths and dramatically different directions of the tool. To guarantee efficient and high-quality fabrication, the simultaneous strategy needs to ensure two key properties of the generated tool path during its planning phase: directional and geometric continuity. ***Geometric continuity*** refers to the minimization of the number of tool paths for machining, as discontinuous paths invariably generate numerous paths of transition that can hinder the overall efficiency of machining. ***Directional continuity*** refers to smooth and consistent variations in the direction of the tool, as frequent changes in it can lead to defects in surface finishing and reduce the efficiency of machining.

However, it is challenging to ensure directional and geometric continuity during the planning of a collision-free tool path for si-

† According to Statistics MRC [Gii23] and a research report [Wic24], the global four-axis and global five-axis CNC machining center markets were valued at $34,012.22 million and $4,119.9 million in 2023, respectively.

multaneous subtractive manufacturing along four axes. Moreover, we cannot simply replicate the two independent computational stages of the positional strategy to this end. This is due to the coupling between the direction and movement of the tool in the simultaneous strategy. Different directions of the tool can result in different machining sequences that affect the optimization of its direction. Tool path planning to apply the simultaneous strategy to four-axis CNC machines remains an open problem. To the best of our knowledge, few solutions to it are available in industry-standard CAM systems. However, the relevant methods can yield objects with simple geometries. Currently available solutions fall short in case of complex geometries featuring high-genus shapes or numerous branching structures, and often encounter such issues as overcuts or undercuts.

In light of the above, we propose an end-to-end framework for producing a collision-free tool path with directional and geometric continuity for simultaneous four-axis machining. Our method can be used to fabricate complex 3D shapes, including high-genus shapes and shapes with numerous branching structures (see Figure 1). We target the finishing (fine-machining) stage, which is performed by using ball-end mills or straight-groove pointed tools, and assume that only the spherical and conical parts of the tool have the capability of cutting. The specific tool shapes have been provided in the “Results" section. Our approach solves this problem of planning the path of the tool in two ways. First, we propose simplifying the scenario by transforming the 3D problem of planning the path of the tool into a 2D planning problem by using a layer-based approach to fabrication. By dividing the target object into slices, we tackle tool path planning for a simultaneous machining strategy for each layer. This allows us to break down the problem into an appropriate level of complexity for simultaneous four-axis subtractive manufacturing. 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 process followed by a process of bottom-up merging to jointly optimize the direction and movement of the tool. Specifically, we break down tool path planning for each layer into three computational stages. The first stage is the over-segmentation stage, wherein the contour of each layer is uniformly decomposed into atomic segments, each of which is then subjected to accessibility analysis. The second stage is bottom-up merging. It is designed to generate a path of machining for the tool that is as continuous as possible by merging the segments through a back-and-forth procedure of traversal, a graph cut-based procedure to resolve overlaps, and a TSP connection procedure. The third stage involves post-processing optimization to further enhance directional continuity and shorten the path of transition.

In summary, our key contribution here consists of developing the first general computational framework for simultaneous four-axis subtractive manufacturing, by focusing on generating a continuous tool path with minimal directional variation and the minimal number of paths of transition. We conducted nine fabrication and three ablation experiments to verify the effectiveness of our proposed technology. Furthermore, we performed three comparative experiments involving the industry-standard CAM systems Snapmaker [Sna23] and Autodesk [Wor23], as well as the four-axis positional machining introduced by [NTM*21], to demonstrate the significant improvements in the quality and efficiency of manufacturing brought about by our proposed method.

2. Related Work

Tool path planning is a classical subject of research on CNC machining. It has been addressed by using a wide range of approaches, including the parameterization method that maps a curved surface to a plane [RSG09], drive surface-based method that generates iso-planar tool paths by using intersecting parallel planes [CJ12; HBA13], and iso-scallop tool path method that seeks to obtain a uniform scallop distribution [CÜ10; LKLF21], and has been examined particularly extensively in the context of five-axis CNC machining [MPE17; EE18; BBR*21]. Rather than providing a comprehensive survey of research on tool path planning [YJJ*22], we focus here on studies that have investigated strategies for tool path planning in the context of four-axis CNC machining. We initially examine past work on the positional machining strategy.

Despite the availability of several commercial CAM systems in the industry, we have been unable to find any study that has addressed the problem of simultaneous tool path planning for four-axis CNC machining.

We review research that has focused on 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 adequately addressed by various methods in past work. An interaction-based method was developed in [DJ04] that involves users assigning orientations for the machining of free-form surfaces in applications of electric discharge machining. [MLS*18] proposed a method that decomposes a 3D object into height fields and then projects the decomposition toward the interior, such that this covers the entire volume of the object and ensures that each piece can be manufactured by using three-axis CNC machines. [Jos15] proposed a method to determine the orientations in CAD models based on such precise geometric primitives as lines, arcs, circles, and polygons. In case of non-complex parts, all features of which can be machined from two directions, [ZCW16] sought to find the best pair of orientations that could avoid thin web structures while preserving the life of the cutter. [FCM*18] used a polycubic representation of the original shape to decompose its surface for four-axis CNC machining.

Researchers have also addressed the positional machining strategy in the context of surface decomposition. In this process, the external surface of the target 3D model is decomposed into a minimal number of height field patches by using multi-label graph cut optimization [STC09]. Each decomposed height field patch is associated with a single direction of the tool. This graph cut-based method of surface decomposition is known to be effective for three-axis [HMA15], four-axis [NTM*21], and five-axis machining [ZZX*18] as well as volumetric decomposition for molding [AMG*19]. Our approach also uses multi-label graph cut optimization. However, instead of seeking to reduce the number of directions of the tool, we focus on minimizing the number of machining segments along the contour of each layer of the object. This approach helps achieve the desired geometric continuity necessary for simultaneous four-axis subtractive manufacturing.

Regardless of whether the positional or the simultaneous machining strategy is considered, accessibility analysis is crucial to formulate a plan of fabrication that is free of collisions. [FWJ06] proposed slicing the layers of the input 3D model for the accessibility analysis of the positional machining strategy. They started by computing 2D visibility maps of a set of the sliced contours in each layer, and then used them to determine the minimum number of directions of the tool. However, this approach relies on CAM software to generate the paths for each of the determined directions of the tool for positional machining. In the context considered here, we develop an algorithm to generate tool paths for continuous machining.

2.2. Simultaneous Machining Strategy

To the best of our knowledge, this is the first study to address the joint optimization of the direction and movement of the tool for simultaneous four-axis subtractive manufacturing. Consequently, most of the relevant literature is on CAM systems used in the industry. Due to the abundance of CAM systems and a lack of reports on their use for commercial purposes, it is impossible to fully explore and understand all such systems and their algorithms. To provide a brief understanding of the simultaneous four-axis strategy used in the industry, we consider technologies to which we currently have access. These include the Luban software attached to our four-axis Snapmaker CNC machine [Sna23], the Fusion 360 software developed by Autodesk [Wor23], and the Siemens NX [Sie16].

An examination of the source code released by Snapmaker [Sna23] reveals that Luban addresses four-axis machining by using the convex hull of the sliced contours of the object. It generates a 360-degree tool path along the boundary of the convex hull of each sliced layer and projects it onto the sliced contours. However, the results of our experiments showed that it can handle only simple geometric models that 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 simultaneous four-axis machining. Users have the option of selecting from among three rotating tool paths: spiral, linear, or circular. However, the results of our experiments revealed that Fusion 360 fails to produce a completely collision-free path for machining, which results in numerous undercuts. The Siemens NX offers a semi-automatic strategy for generating simultaneous tool paths for CAD models [Sie16], where this necessitates the manual specification of the driving geometry or guiding curves to generate tool paths at a feature-based level, such as a circular pocket or a slot feature. By contrast, our technique provides a fully automatic solution for generating simultaneous tool paths for the entire model. Therefore, there is no need to compare our technique with that of the Siemens NX.

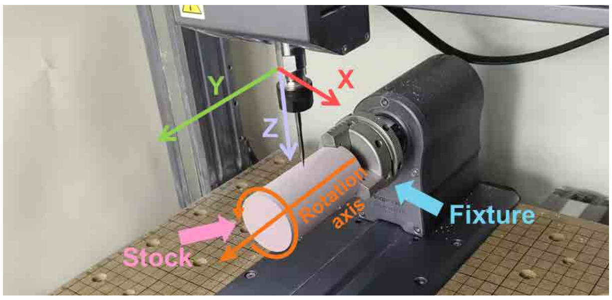


Figure 4: ***The setup of our four-axis machine.*** *The milling tool has three degrees of freedom (DOFs) and the rotational axis provides the fourth.*

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 the quality of the product. Previous studies have attempted to enhance the directional continuity or the geometric continuity in this context [MSJ*23]. This serves as the inspiration for our algorithm to optimize the continuity of the direction of the tool and the sequence of machining.

First, we use a graph-based representation in the graph cut step of our algorithm to ensure directional continuity. This is akin to the procedure in [PL14], and determines the direction of the five-axis machining tool through graph-based optimization. Second, we apply the over-segmentation and merging strategy, similar to that used in [ZGH*16] and [ZXZL23], to ensure geometric continuity. These studies involved initially decomposing a 2D domain (surface model) into several sub-domains (small-scale one-path patches), and subsequently generating a single Fermat spiral (a single one-path patch) through a process of bottom-up merging. In the case considered here, we first decompose each layer into short segments and then connect them to form a single tool path.

3. Overview

In this section, we begin by explaining the configuration of our four-axis CNC machine and provide an overview of the process of fabrication. We then reiterate the basic idea of our computational framework and provide an overview of the proposed technique.

3.1. Setup and Fabrication

We used the four-axis CNC machine Snapmaker 2.0 A350T, manufactured by Snapmaker [Sna23], in our experiments. It consisted of three linear axes of movement and an additional axis of rotation, as shown in Figure 4. We first assembled the stock on the four-axis machine through a fixture in the process of fabrication, so that its axis was aligned with the rotational axis. We then milled the stock layer by layer, as shown in Figure 5.

3.2. Overview of Proposed Technique

We propose an algorithm that forms an end-to-end framework for simultaneous four-axis machining. It takes a 3D object $\mathcal{M}$, represented by a triangular mesh, as the input and generates a

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

continuous collision-free tool path $\mathcal{TP} = \{TP_1, TP_2, ..., TP_n\}$ for its four-axis machining. TP_i represents the path of the tool for the i-th slicing layer. Each TP_i comprises a sequence of machining segments and paths of transfer. It is denoted by $TP_i = \{S_1^i, T_{1,2}^i, S_2^i, ..., T_{m-1,m}^i, S_m^i\}$, where S_k^i is a continuous segment of the tool path for the machining of the i-th slicing layer, and $T_{k-1,k}^i$ is the path of transfer between S_{k-1}^i and S_k^i. To ensure the directional and geometric continuity of TP_i, our framework reduces the length of its sequence to minimize changes in the direction of the tool within each TP_i. Our algorithm achieves the above objectives in three stages:

1) During initialization, we first determine the orientation of the object $\mathcal{M}$ (Sec. 4.1), slice it into n layers, $\mathcal{L} = \{L_1, L_2, ..., L_n\}$, and uniformly sample sub-segments along each slicing contour C_j^i of layer L_i. We then analyze the accessibility of each sub-segment (Sec. 4.2).

2) During the over-segmentation and merging process, we first decompose each contour C_j^i into a set of machining segments by using a back-and-forth process of traversal. We then use the graph cut method to resolve the overlap between these segments to obtain machining segments $\{S_1, S_2, ..., S_n\}$ (Sec. 4.3). The resulting machining segments are connected to form a single tool path TP_i for layer L_i while seeking to minimize the length of path transfers (Sec. 4.4). Following this, we subject each TP_i to post-processing to further optimize its points of connection and directions of machining (Sec. 4.5).

3) We connect the tool paths $\{TP_1, TP_2, ..., TP_n\}$ of each layer to form a single tool path $\mathcal{TP}$ by identifying one connecting point for each TP_i. As each TP_i is a circuit, its connecting point is both its starting and ending point. To simplify the computation, we select the connecting point of TP_i with the maximum value along the z-axis. Finally, we generate $\mathcal{TP}$ by inserting paths of transfer as the tool retracts between the connecting points of adjacent layers.

4. Proposed Method

This section provides a detailed description of each step of our proposed algorithm. It is designed to generate a path for the tool for each layer of the object with minimal variations in its directions (directional continuity) and a minimal number of transfer moves (geometric continuity) for simultaneous four-axis CNC machining.

4.1. Object Orientation

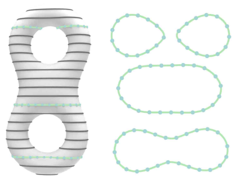
The orientation of the object refers to the alignment of the given object relative to the rotational axis of the four-axis machine. Before planning the path of the tool, we determine the orientation of $\mathcal{M}$ by using a similar method to that detailed in [NTM*21]. The only difference is that we do not divide the top area of the model to avoid the seam line caused by decomposition. Our aim here is to avoid flat areas, the normal direction to which is nearly parallel to the rotational axis, that are easily omitted by slicing. To determine the orientation of the object, we begin by generating a set of candidate orientations $\{\vec{d}_1, \ldots, \vec{d}_k\}$. This is achieved by uniformly distributing points in a hemisphere by using the Fibonacci sphere algorithm [SJ06] (with $k = 2000$ in our implementation). The best orientation is selected from among $\{\vec{d}_1, \ldots, \vec{d}_k\}$, and 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 face of the triangle f_j, $\vec{n}_j$ is the surface normal of f_j, and $\vec{d}_i$ is the candidate direction of the rotational axis.

4.2. Accessibility Analysis

The avoidance of collisions is a challenging constraint to impose when planning the path of the tool. We tackle this through a pre-computation process during the initialization stage. We precompute the range of machinable directions for each surface point of $\mathcal{M}$. This process involves a slicing-and-sampling approach, followed by a collision detection-based accessibility analysis of each sampled surface point.



Slicing and sampling. We slice $\mathcal{M}$ with flat planes vertical to the rotational axis to obtain n slicing layers, $\mathcal{L} = \{L_1, L_2, \ldots, L_n\}$. Following [Lee03], we empirically formulate the dependency between the scallop h and the thickness of slicing t between adjacent layers of the path as follows:

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

where R is the radius of the tip of the tool. To achieve a sufficiently high accuracy of machining, we set the thickness of each slice to 0.2 mm and use a tool with a radius of 0.15 mm. This allows us to obtain a scallop height of 0.033 mm (see Figure 13). There is one or more slicing contour $\mathbf{C} = \{C_1^i, C_2^i, ..., C_m^i\}$ in each layer L_i, where C_j^i denotes the j-th contour of the i-th layer. We then uniformly sample the atomic segments $\mathcal{A}^{i,j} = \{a_1^{i,j}, a_2^{i,j}, ..., a_l^{i,j}\}$ 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 of the figure shows the output of slicing and uniform sampling.

Collision detection. This step is designed to compute the range of machinable directions, abbreviated as 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 by using the layer-based strategy in our technique. The MDR of $a_k^{i,j}$ is composed of 2D sectors of the machinable direction, abbreviated as MDS. The directions within these sectors are collision free, and allow the CNC tool to carve $a_k^{i,j}$ without interference. The MDR of $a_k^{i,j}$ may include multiple MDS, $\{MDS_1, MDS_2, ..., MDS_n\}$, in which the t-th MDS

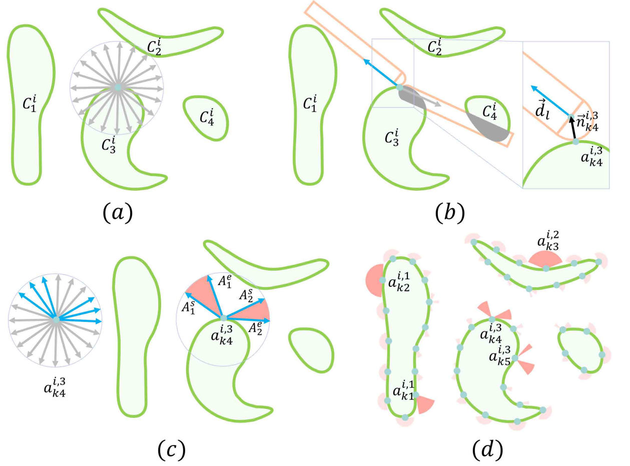


Figure 6: ***Accessibility analysis.*** *(a) Uniform sampling of directions of machining. (b) Collision detection along the direction of sampling, showing examples with and without collisions. (c) Results of collision detection, and the generated MDR from the accessible directions of machining. The red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

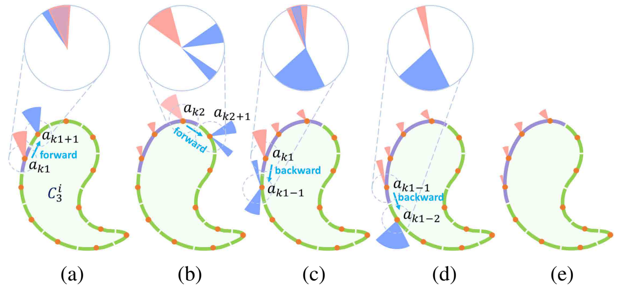


Figure 7: ***Path segment generated by back-and-forth traversal.*** *We start from atomic segment* a_{k1}*: (a) We first traverse the adjacent atomic segment* a_{k1+1} *on contour* C_3^i *in the clockwise direction. As the traversed MDS overlaps,* a_{k1+1} *is included in the path segment. (b) The clockwise traversal terminates at* a_{k2+1}*, with no MDS overlap between* a_{k2} *and* a_{k2+1}*. (c, d) We then traverse the contour in the counterclockwise direction starting from* a_{k1}*, merge* a_{k1-1}*, and terminate at* a_{k1-2}*. (e) Finally, we generate a path segment that contains four atomic segments.*

is defined by a starting boundary angle A^s and an ending boundary angle A^e, and is represented by $MDS_t = (A^s, A^e)$.

We calculate the MDR of $a_k^{i,j}$ by using a sampling-based method. We first uniformly sample the candidate directions of machining at 5° intervals to obtain $\{\vec{d}_1, \dots, \vec{d}_n\}$. Let $\vec{n}_k^{i,j}$ be the normalization vector of the surface normal of $a_k^{i,j}$. For each candidate direction $\vec{d}_l$, we designate $a_k^{i,j}$ as the cutter contact (CC) point. Following this, we position the cutter location (CL), which in this paper is set as the center 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 represents the radius of the sphere). We then align the direction of the cutter with $\vec{d}_l$. We then check for collisions between the cutter and $\mathcal{M}$ by checking if any of the sampled atomic segments on L_i is inside the cutter. If none of the sampled atomic segments is inside the cutter, then no collision has occurred and $\vec{d}_l$ is machinable. Finally, we group all the machinable directions into the machinable sectors of $a_k^{i,j}$, and assign the starting and ending boundary angles A^s and A^e, respectively, for each MDS_t as shown in Figure 6(a, b, c). Figure 6(d) shows an example of the results of accessibility analysis. Note that while the posture of the tool can be uniquely determined by the directions of machining of the two tools above, custom-shaped tools, like an ellipsoid tool, require an additional variable for this purpose. Hence, the above method does not support such tools.

4.3. Decomposition of Path Segments

This step involves decomposing each contour C_j^i of the i-th layer into the minimum number of continuous tool paths for machining, $\{S_1, S_2, \dots, S_n\}$. We define such a continuous path as a ***path segment*** S_k, and it can be machined continuously with minimal variations in the direction of the tool. A path segment can be viewed as a basic item with the desired directional and geometric continuity. For the simultaneous subtractive manufacturing of each layer, the fewer path segments a layer contains, the better its directional and geometric continuity is. We use the over-segmentation-and-merging strategy to solve the problem of decomposition of the path segments.

We use the technique described in Sec. 4.2 to over-segment each contour C_j^i 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 to be an initial path segment. We then merge these initial path segments. We use an iterative greedy method in which each iteration generates a path segment from C_j^i. However, this approach often fails to achieve the minimum number of path segments. In light of this, we propose a graph cut-based method to this end. Before detailing the above two methods, it is important to describe the back-and-forth traversal process that is used in both.

Back-and-forth traversal. This process aims to generate the longest path segment (S_k) from C_j^i, starting from one of its segments $a_k^{i,j}$. When calling the back-and-forth traversal process, a specific MDS (MDS_t) of $a_k^{i,j}$ must be input to the algorithm, referred to as the **traversal 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. We initially 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 traversed MDS of the current atomic segment $a_{cur}^{i,j}$, we include it in the path segment and designate the corresponding MDS as the traversed MDS of $a_{enc}^{i,j}$; see Figure 7. If $a_{enc}^{i,j}$ is successfully included, this indicates that the machining tool can continue machining between $a_{cur}^{i,j}$ and

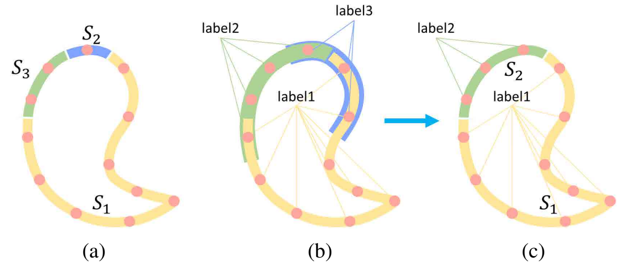


Figure 8: ***Two strategies for generating path segments.*** *(a) Greedy method. The contour is decomposed into three non-overlapping path segments. (b) Graph cut method: It first generates candidate path segments on the contour, and then uses the graph cut algorithm to resolve the overlap between path segments. (c) Finally, we obtain two path segments, fewer than the number of segments obtained when using the greedy method.*

$a_{enc}^{i,j}$ by using any direction within the overlapping sectors of their traversed MDS.

Greedy decomposition of path segments. This method uses a heuristic greedy strategy. The key heuristic rule is to call back-and-forth traversal to generate as few path segments as possible while determining a traversed MDS for each atomic segment. We first randomly select an atomic segment $a_k^{i,j}$ from among all the atomic segments of C_j^i, and randomly determine its traversed MDS. We then call back-and-forth traversal to generate the longest path segment from $a_k^{i,j}$. Following this, we repeat the previous operation to generate the longest path segment from the remaining atomic segments of C_j^i. Here, the remaining atomic segments refer to those that are not among the already generated path segments. The iterations continue until the generated path segments include all the atomic segments, thereby also determining the traversed MDS for each atomic segment as shown in Figure 8(a). A complex scenario involving the application of the greedy method can be found in Figure 14.

This method does not yield overlapping path segments. However, such a greedy method can easily overlook the optimal solution 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 Decomposition of Path Segments. This method initially generates a set of potential path segments C_j^i for the tool. It does so by running back-and-forth traversal by starting from each MDS of every atomic segment of C_j^i. Because a unique path segment can be obtained if back-and-forth traversal starts from any atomic segment within it and its traversed MDS, we do not need to re-run back-and-forth traversal if the MDS of one atomic segment has been included in the generated path segment. However, the resulting path segments may overlap (see Figure 8(b)), which leads to multiple machining passes when we directly use them as the path of the tool. Therefore, we need to resolve these overlaps while minimizing the number of path segments generated. To this end, we apply a multi-label graph cut algorithm [STC09].

We first associate a label with each path segment and then assign it to all points within the segment. In case of overlaps between path segments, an atomic segment may have multiple labels. We seek a label assignment l that minimizes the following energy function:

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

where D is the data term, S is the pairwise smoothness term, $\mathcal{A}$ represents the atomic segments of a contour, and α is a trade-off parameter between D and S ($\alpha = 2000$ in our implementation). The data term D is used to estimate 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 traversed MDS of atomic segment a_i in path segment $l(a_i)$. We can then define D as follows:

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

The above formula tends to choose a larger traversed MDS when a_i is within a segment. A larger MDS implies a safer direction of machining, and provides a broader range of options for the direction of the tool, where this is conducive to finding a smoother direction in subsequent post-processing, described in Sec. 4.5. The smoothness term S measures the cost of assigning path segments (labels) to two adjacent atomic segments a_i and a_j. We define S as follows:

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

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

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

4.4. Connection between Path Segments

This step aims to connect all non-overlapping path segments along L_i into a single tool path TP_i by generating a machining sequence and paths of transfer between adjacent path segments. We seek to minimize the length of the paths of transfer to reduce the machining time. This can be regarded as the classic traveling salesman problem (TSP), which is NP hard [HPR*13]. We construct a weighted complete graph G, in which 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 zero, and otherwise set its weight to the length of the transfer path. If the tool can move along a straight line between endpoints without incurring a collision, the line segment connecting them can be considered to be the transfer path. If there is a collision, we generate a transfer path by using the additional paths, known as retraction paths (see Figure 9(b)). Generating retraction paths involves retracting the tool to a safe intermediate point, known as the retraction point, and requires retracting it by a certain distance that is known as the retraction distance. Suppose e_k and e_{k+1} are endpoints of two path segments, with the middle directions of their traversed MDS being $\vec{d}_k$ and $\vec{d}_{k+1}$, respectively. To generate a retraction path, we calculate the retraction points $wd_k = e_k + W * \vec{d}_k$

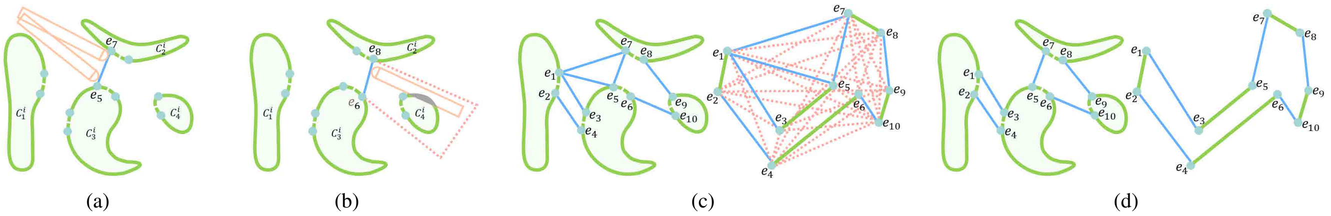


Figure 9: ***Connection between path segments.*** *(a) Two path segments can be connected by a straight transfer move (blue line). (b) Collision occurs if two segments are connected by straight transfer moves (blue line). We avoid collisions by retracting the tool (red lines), which inevitably prolongs the path. (c) The complete graph built by our method. The blue lines means that the machining tool can move from one endpoint to another in a straight line. The red lines means that the connection requires a retraction operation. The green lines connect nodes that are endpoints of the same path segment. (d) The resulting path is calculated by an exhaustive method to solve the TSP.*

and $wd_{k+1} = e_{k+1} + W * \vec{d}_{k+1}$, where W is the retraction distance ($W = 35$ mm in our implementation). In this case, the retraction path between e_k and $ek+1$ consists of three straight segments connecting the four points e_k, wd_k, wd_{k+1}, and e_{k+1}.

To balance the performance and efficiency of the algorithm, we propose two TSP solvers for G depending on the number of nodes. With 60 nodes of fewer, we run an exhaustive method that starts from a randomly selected node, and then traverses G by using the depth-first search (DFS) strategy. The path from the head node to the leaf node in the search tree is taken as the path for the TSP. For graphs with more than 60 nodes, we propose an iteratively greedy TSP solver. Starting from a randomly selected path segment, it seeks the nearest path segments to the two endpoints of the generated TSP path in each iteration. See Figure 9 for an example of the proposed methods to connect path segments. Moreover, we propose a heuristic rule to expedite the TSP solvers. Essentially, when the endpoint of an untraversed path is included in the search, the node of the other endpoint is set as the next traversed node.

4.5. Post-processing optimization

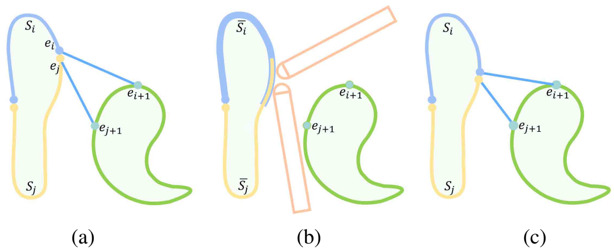


Figure 10: ***Fine-tuning the endpoints of the path segment.*** *(a) is the result of the TSP connection. (b) shows the calculated overlapping path segment, from which we select the endpoints to fine-tune. (c) shows the endpoints of the path segment obtained after post-processing. A comparison between (a) and (c) shows that the distance of connection in the latter is clearly shorter.*

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

Fine-tuning the endpoints of path segment. This step aims to slightly shorten the paths of transfer by adjusting the locations of their endpoints. These endpoints are indeed the endpoints of the path segments as well. We apply the fine-tuning process to each endpoint of the path segments 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 $\overline{S}_i$ and $\overline{S}_j$, starting from e_i and e_j, and their traversal MDS, respectively. We then identify the overlapping path segment between $\overline{S}_i$ and $\overline{S}_j$. Following this, we 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)). e_{i+1} and e_{j+1} are the other endpoints of the transfer paths connecting S_i and S_j.

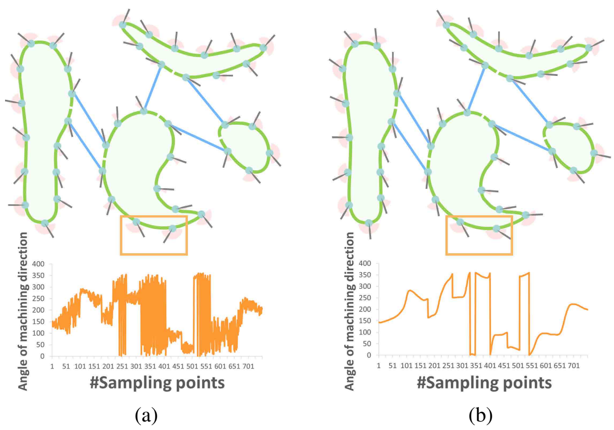


Figure 11: ***Smoothing of machining directions.*** *The horizontal axis of the line chart represents the index of the atomic segments, and the vertical axis represents the angle of the machining directions. (a) The initial direction of machining of each atomic segment is randomly selected in its MDR,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

Smoothing of machining directions. In case of simultaneous four-axis machining, we need to determine the direction of machining of each atomic segment of the final TSP path. We initialize the direction by randomly selecting one within the traversed MDS of

Table 1: *Statistics of the results. R_s is the surface area-to-volume ratio. H is the height (mm) of the model along the orientation of the object. The number of slices is determined by dividing the height of the model by the thickness of the layers. #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 numbers of input and output atomic segments for graph cut, respectively. E is the average number of endpoints per layer in the TSP. D is the average length of the final tool path after post-processing. A is the average transition in the directions of machining of adjacent atomic segments after post-processing. T is the total fabrication time (minutes).*

		4.1	4.2		4.3		4.4	4.5		
Model	R_s	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

each atomic segment (see Figure 11(a)). We then iteratively apply the Laplacian smoothing method to all adjacent points [SCL*04]. In each Laplacian iteration, we update the direction of machining of an atomic segment by first taking the average direction of its pre-order, post-order, and the atomic segment itself, and then setting the direction of machining to the updated one, so long as it belongs to the traversed MDS of the atomic segment. The smoothing continues until the sum of changes in angles along all directions is smaller than 1° (Figure 11(b)). Finally, we sample the atomic segments uniformly along all transfer paths by using the same spacing of 0.2 mm, as mentioned in Sec. 4.2. The direction of machining of the new atomic segments is determined by a linear interpolation between the directions of machining of their two endpoints. The line chart in Figure 11(b) shows the direction of machining of each atomic segment, the transition of which is significantly smoother than that before optimization in Figure 11(a).

5. Results

This section details the planning of the path of the tool and the generation of 3D models with varying degrees of topological complexity. We conduct a thorough evaluation of the efficiency, generality, and effectiveness of our algorithm. We also compare it with prevalent approaches in the field, and discuss its limitations.

5.1. Implementation and Parameters

Our algorithm was implemented in C++ by using CGAL [FP09] and Libhgp [Zha24] for geometric processing, Eigen [GJ*10] to solve the linear equations, and gco-v3.0 [VD15] for graph cut optimization. We ran the program on a PC equipped with an Intel Core i7-13700 CPU running at 2.1 GHz, and with 32 GB of RAM. To determine the orientation of the object, we sampled 2,000 candidate orientations in the Gaussian sphere. We set the thickness of slicing to 0.2 *mm* and the sampling interval to 0.2 *mm* to uniformly resample the atomic segments on each contour. For collision detection, we uniformly sampled 72 directions of machining after every 5 degrees. We set the retraction distance to 35 mm for the connection between path segments. Of the above hyper-parameters, the thickness of slicing was the most crucial as it directly determined the number of layers. A smaller value of thickness implies more layers, increases the fabrication time, but also improves the surface quality of the machined object.

5.2. Simultaneous Four-axis Tool Path

We assessed the efficiency and capability of path planning of our algorithm. Figure 12 depicts the results of its path planning for eight models, each with two or four visualized tool paths. Figure 13 shows the results of analysis of the surface scallop heights of three models subjected to simulated machining, by using Siemens NX, based on the paths generated by our algorithm. Table 1 provides the relevant statistical data, while Table 2 details the run time of the algorithm for each step. We also conducted a physical experiment to validate the proposed method to decompose path segments and the post-processing optimization.

5.2.1. Evaluation of Path Planning

As shown in Figure 12, our algorithm generated tool paths for both single and multiple contours within a slicing layer. All paths exhibited excellent directional continuity and geometric continuity. The directions of the tool are represented by the smooth red lines. The lengths of the transfer paths, represented by the blue lines, were also reduced. Most contours were decomposed into multiple path segments that were processed by the machining tool in an interleaved order. This implies that the tool moved between contours, and a single contour could be visited multiple times, as in the fourth tool path in the Coral model. It is evident from Table 1 that the average number of path segments generated per layer increased with the average number of contours (refer to $\#P_C$ and $\#C$ for *Coral* and *Hand*). This is likely because an increase in the number of contours reduced the range of machinable directions of the sampling points. As a result, more path segments were needed to process each contour.

5.2.2. Algorithmic Efficiency

Our algorithm took about 9 minutes on average for each model in our experiments (see Table 2). The accessibility analysis took the most time because we sampled a large number of candidate directions to assess the accessibility of the cutter to each sampled atomic segment. The run time for accessibility analysis was determined by both the number of sampling points (#*S*) and the number of contours (#*C*) of all layers, as shown in Table 1. The steps of the proposed algorithm other than the determination of the orientation of the object and accessibility analysis were quickly executed.

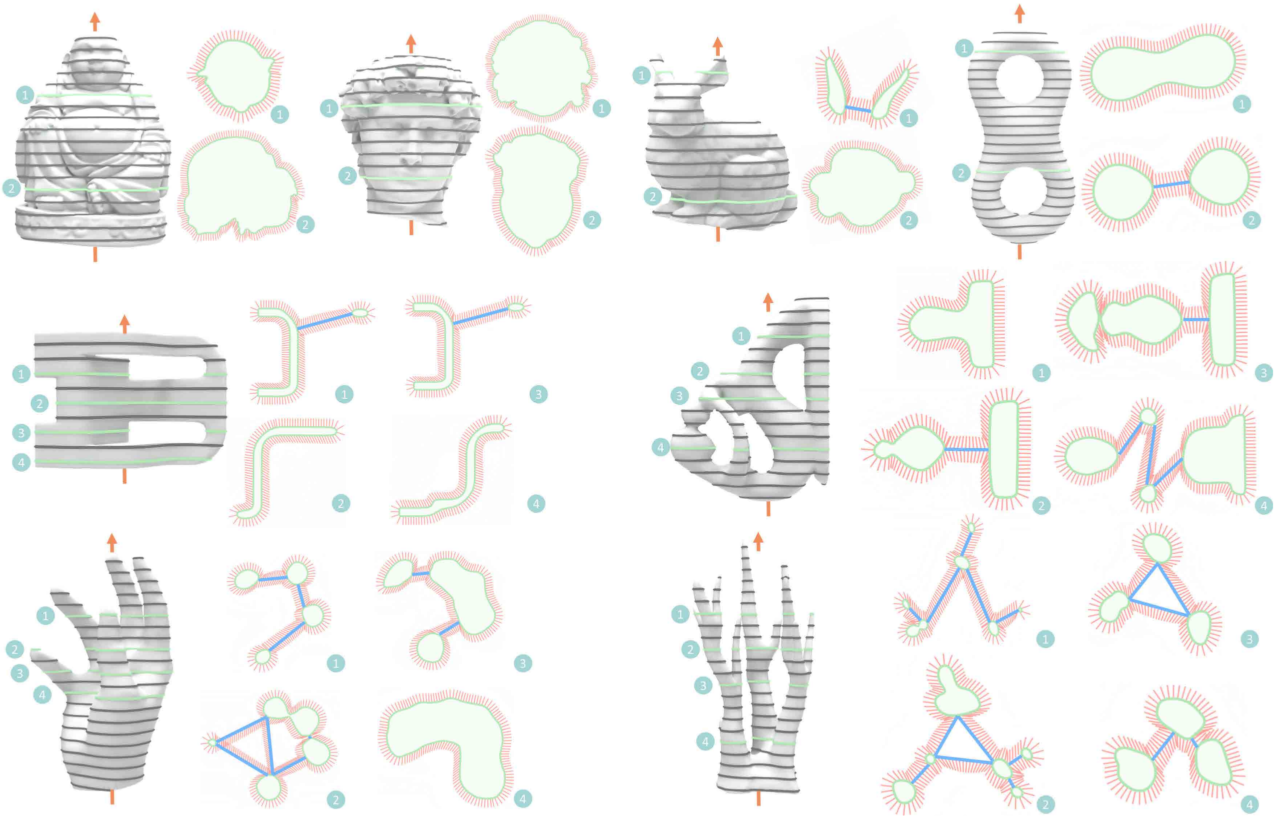


Figure 12: ***Gallery of tool paths generated by our method.*** *The models were arranged in order of Buddha, David, Bunny, Eight, Chair, Fertility, Hand, and Coral. We show the determined orientation of each model and its layer-by-layer slicing, where two or four layers have been chosen and presented. The red lines represent the directions of the tool for the atomic segments, while the blue lines represent the transfer moves between path segments. Note that we used a thicker slice for visualization, and the intersection between the directions of the tool (red lines) does not imply collisions as the tool moved linearly.*

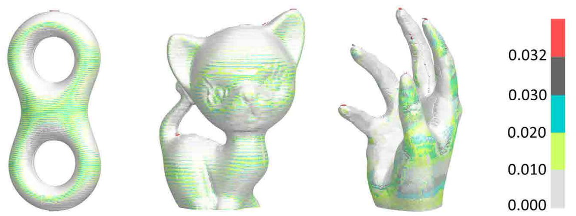


Figure 13: ***Results of analysis of the surface scallop height.*** *This figure shows the results of analysis of the surface scallop heights of three models: Eight, Kitten, and Hand. It is clear from it that the surface scallop height was smaller than the maximum scallop height of 0.033 mm.*

5.2.3. Greedy vs. Graph cut

To demonstrate the effectiveness of our strategy for the decomposition of path segments (Sec. 4.3), we compared the greedy method and the graph cut method on the *Coral* model. As shown in Figure 14, the graph cut method yielded fewer path segments in each slicing layer of the *Coral* model. Moreover, the advantage of the graph cut method was more noticeable when both methods generated a large number of path segments. We recorded the average number of path segments, average length of the tool path per layer, and the fabrication time of both methods. Their values for the greedy method (graph cut method) were 6.4 (5.5), 149.2 mm (88.7 mm), and 129 min (117 min).

5.2.4. Post-processing Optimization

To verify the two post-processing methods detailed in Sec. 4.5, we compared the surface quality of the object and the fabrication time with and without these methods. We used the *Eight* model to this end (see Figure 15).

Figure 15(a) shows the results of fabrication when the directions of machining were not smoothed by post-processing. These directions were randomly selected within the traversed MDS of each atomic segment. A large number of defects were evident on the surface of the model. We also compared the heuristic methods used to determine the directions of the tool (see Figure 15(b)). The heuristic method first selected the normal direction of each atomic segment or, if this was inaccessible, the closest direction within its traversed MDS. Figure 15(d) shows the results of fabrication with

Table 2: *Program run time for each step (s). Ori represents the orientation of the model. Acc represents accessibility analysis. Seg represents the decomposition of the path segments. TSP represents the connection between path segments obtained by solving the TSP. Con represents the endpoints of the path segment that were fine-tuned. Smo fine-tuning smoothing of the machining directions. Tot is the total time.*

	4.1	4.2	4.3	4.4	4.5		
Model	*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

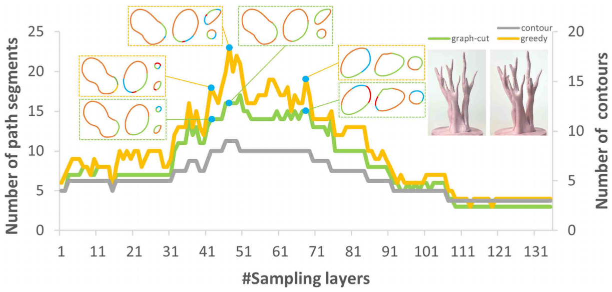


Figure 14: ***Comparison between the graph cut and greedy methods.*** *Comparison between the graph cut and greedy methods on the Coral model. The graph cut method (green curve) consistently generated fewer path segments than the greedy method (yellow curve). Furthermore, the number of path segments generated increased with the number of contours (gray curve). Three layers were selected to demonstrate the results of decomposition of the path segments.*

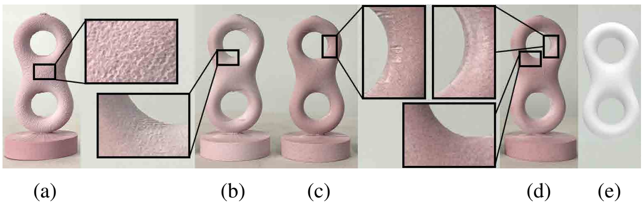


Figure 15: ***Comparative post-processing experiments.*** *(a) The Eight model fabricated by using the tool path without smoothing the directions of machining. The abrupt transition in tool direction led to the formation of a large number of pits on the surface, which seriously reduced the surface quality. (b) Results of fabrication obtained by using the heuristic method. It selected the normal direction of each atomic segment. If this was inaccessible, it chose the closest direction within its traversed MDS. The heuristic method yielded some overcut artifacts. (c) Results of fabrication obtained by using the tool path without fine-tuning the endpoints of the path segments. (d) Results of fabrication obtained by using the tool path with two post-processing strategies. This yielded a better surface quality. (e) Rendered view of Eight.*

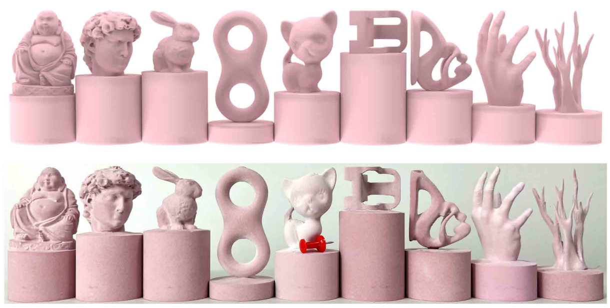


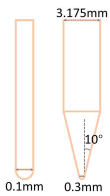
Figure 16: ***Gallery of the results of fabrication.*** *The upper portion of the figure shows a rendered view of the corresponding input 3D models, while the lower portion shows images of the results of fabrication.*

post-processing to smoothen the directions of machining. The average variations in the angle per layer for (a), (b), and (d) were 2741°, 636°, and 451°, respectively. Because tuning the rotary axis during machining takes time, Figure 15(d) took 74 min for fabrication, which is much shorter than the 372 min taken in Figure 15(a) and the 95 min in Figure 15(b). Figure 15(c) shows the results of fabrication of the *Eight* model without fine-tuning the endpoints of path transfer. The surface quality of the machined object was slightly poorer than that shown in Figure 15(d), which was fine-tuned. The average length of the tool path per layer in (d) was 57.5 mm, which is slightly better than the value of 57.7 mm shown in (c).

5.3. Physical Evaluation

For the physical evaluation of the proposed method, we first introduce the setup of the fabrication experiment and then evaluate the results in terms of its efficiency and the surface quality of the machined object. We compared our method with two CAM systems: the Luban system developed by Snapmaker and Fusion 360 by Autodesk. For a live demonstration of the manufacturing process, the interested reader can refer to our supplementary video.

5.3.1. Setup of fabrication experiment.



The results of all fabrication experiments were generated in Snapmaker 2.0 A350T, which had a fabrication space of 350 × 320 × 330 *mm* and a spindle speed of 15,000 r/min. We used machinable cylindrical resin boards as machining stock, each with a height of 70 mm and a radius of 17.5 mm. Except for the result shown in Figure 19, which was obtained by using a ball-end mill with a diameter of 1.0 mm, the default milling tool was a two-edge straight-grooved pointed tool. The length of the carving knife was 24 mm, the diameter of the tip was 0.3 mm, the diameter of the shank was 3.175 mm, and the total length of the tool

Figure 17: ***Results of fabrication with detailed close-ups.*** *This figure shows close-up photographs of the results of fabrication of four models: Eight, Coral, Buddha, and Fertility.*

was 50 mm. To run the generated path of our tool on Snapmaker, we exported it to a common gcode file [LAYK21] at a feed rate of 800 mm/min. In the setup for fabrication, our tool was longer than the machining stocks (24 mm for the carving tool vs. 17.5 mm for the radius of the stock), and the machinable resin boards had a low hardness. Consequently, we performed finishing directly without requiring rough machining. However, if the tool had been shorter or the hardness of the material had been higher, a rough machining stage would have been necessary. This issue can be addressed in available CAM systems, such as through the positional rough machining tool path in Fusion 360 [Wor23].

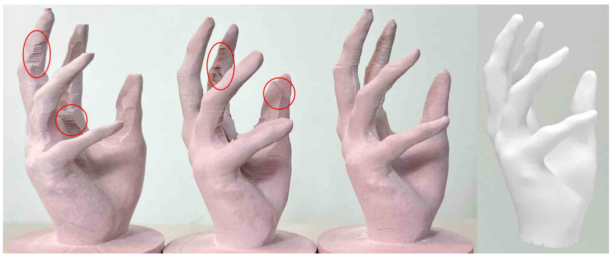
5.3.2. Evaluation of Results of Fabrication

Figure 17 shows close-up, detailed views of the fabricated surfaces of the four models shown in Figure 16. These models were machined by using our simultaneous four-axis tool path, which often yielded objects with excellent surface quality. No boundary artifacts were visible in topologically simple models, such as *Buddha* and *David*. However, in complex model such as *Fertility* and *Coral*, tiny boundary artifacts appeared on the surface due to discontinuities in the directions of machining at the intersections of the path segments (see the inset for *Coral*).

The surface areas where the normal direction was nearly parallel to the rotational axis were removed by slicing, resulting in unmachinable sections. Although we mitigated this issue by optimizing the orientation of the object, these areas inevitably persisted (see the head of *Buddha*). In particular, some sampling points contained void MDR, i.e., there were unmachinable points (see the inset for *Fertility*). Our algorithm simply skipped these points to ensure that the model was successfully manufactured overall. The length of the final tool path per layer and the height of the model were the determinants of the fabrication time, as shown in Table 1. In our experiment, the average time taken to fabricate all models was 68 min.

5.3.3. Comparison with CAM Systems

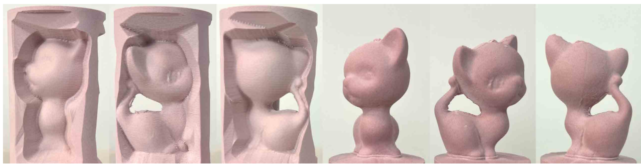
As noted previously, the simultaneous strategy for four-axis CNC machines remains an open research area, with only a few solutions for it available in industrial CAM systems. We compared the Luban software, manufactured by Snapmaker [Sna23], with Fusion 360 [Wor23] by using the same parameters as in Sec. 5.1. We used the *Hand* model for this comparison (see Figure 18). Figure 18(a) shows the results of fabrication obtained by using Luban. The red circles highlight overcut and undercut artifacts, indicating



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

Figure 18: ***Comparison of the results of fabrication of the proposed method with CAM systems.*** *(a) Results of fabrication of the Hand model by using tool paths generated by the Luban CAM software, which took* 115 *min. The results show both undercuts and overcuts (red circles). The little finger is much thinner than it should be because it was raised by the overcut. (b) Results of fabrication of the Fusion 360 CAM software, which took* 320 *min. The results show a large number of undercuts, such that the palm is much thicker than it should be. (c) Results of fabrication of our method, which took* 80 *min. (d) Rendered view of Hand.*

that Luban did not accurately calculate the machinable direction for each surface point in case there were multiple contours in one layer. Figure 18(b) shows the results of Fusion 360, which also clearly exhibited undercut artifacts. Moreover, the tool path of Fusion 360 contained numerous instances of idle rotational movements during machining, leading to significantly prolonged machining times. The results of our method, shown in Figure 18(c), were significantly superior to those of both Luban and Fusion 360, and it took only 80 min, while Luban and Fusion 360 took 115 min and 320 min, respectively.



(a) (b)

Figure 19: ***Comparison with the positional strategy.*** *(a) shows the results of fabrication of [NTM*21]. It was machined from three directions, each of which generated a height field patch. (b) shows the results of fabrication of our method from three views.*

5.3.4. Comparison with Positional Strategy

We compared our method with positional four-axis machining introduced by [NTM*21] on the *Kitten* model (see Figure 19). To keep each variable in the comparison as consistent as possible, the parameters considered in Sec. 5.1 were used for both methods, and we applied a zigzag pattern to generate the tool path for fine machining for the positional strategy. The results for both methods were obtained by using rough machining with a ball-end mill with a diameter of 3.175 mm, while finish machining was applied by using a ball-end mill with a diameter of 1.0 mm. Figure 19(a) shows the results of fabrication of the method proposed in [NTM*21], which yielded many undercut parts that required additional manual intervention to remove. Moreover, the surface quality obtained by it was inferior to that obtained by our method, as shown in Figure 19(b), which was characterized by a less pronounced staircase effect. Note that [NTM*21] treated the top as an independent machining patch, which required separate machining that is beyond the capabilities of four-axis CNC machining. Therefore, we can report only the machining times for its side surfaces: 41 min for the object shown in Figure 19(a) and 62 min for that shown in Figure 19(b). Taking into account the machining of the top patch and the manual removal of undercuts, our method is comparable in terms of performance to that proposed in [NTM*21].

5.4. Limitations and Discussion

Our pipeline enables the manufacturing of complex 3D free-form shapes from a single solid stock by using the simultaneous machining strategy of four-axis CNC machines. To the best of our knowledge, this the first study to propose an end-to-end pipeline that fully exploits the potential of the simultaneous machining strategy for four-axis CNC machines. The key limitations of our technique are threefold: the intrinsic constraints imposed by four-axis CNC machines, the limited search space imposed by layered subtractive manufacturing, and a lack of guarantee of global optimality.

5.4.1. Intrinsic Fabrication-related Limitation

The first limitation, related to intrinsic constraints on fabrication, was introduced in [NTM*21]. As we clarified in the Introduction, four-axis CNC machining is a cost-effective technique of fabrication 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, however. According to [NTM*21], there is no formal definition of shapes that can be manufactured from a single block by using four-axis CNC machines. We have not addressed this problem in this study as it is beyond the scope of our research. Therefore, our method cannot handle the invisible features of the target shapes, such as the *Ruyi* model, with a height of 10 cm, shown in Figure 20, in which the red areas cannot be reached by our default fabrication settings. However, as the size of the model increases, the number of invisible areas decreases until none remains at a height of 100 cm. Our algorithm can handle this scenario and generate a simultaneous four-axis machining tool path for it, 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 planning problem to a 2D planning problem. We achieved this with a layer-based approach to fabrication that simplifies the problem. However, layer-based milling limits the likelihood of achieving an optimal solution for tool path planning in simultaneous machining strategies. A more effective approach to planning the path of the tool may involve combining region decomposition with layer-based milling methods to produce a path that is as continuous as possible across the surface of the object. Further, while our current solution can optimize the path of the

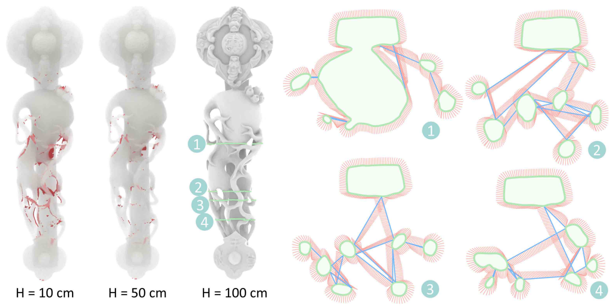


Figure 20: ***Tool paths for the Ruyi model.*** *Left: Three Ruyi models of different heights, where the red areas indicate invisible areas. Right: Four layers are selected to show the tool paths generated by our method.*

tool within each slicing layer, it does not guarantee global optimality. Our algorithm decomposes each sliced contour into minimal fabricable segments by using a multi-label graph cut-based method of optimization [STC09], but it does not guarantee global optimality. However, the graph cut optimizer consistently generated reasonable solutions in our experiments. Moreover, our approach considers only ball-end mills and straight groove-pointed tools, and does not account for other types of tools, such as toroidal cutters. Although our tools have a conical part, four-axis CNC machines lack the number of degrees of freedom needed to effectively position a conical tool for flank milling.

6. Conclusion and Future Work

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

As discussed in the Results section, the main bottleneck in our algorithm is its accessibility analysis. We plan to expedite this step by using CUDA parallelization, adaptive spatial partitions of the bounding volume hierarchy (BVH) [LA06], and the FFT-based collision metric [CRCM23]. Our method offers several avenues for future research in the area. First, research in the field should explore the effectiveness of slicing methods with adaptive thickness [XGD*18], curved slicing layers [ZFH*22], and spiral slicing layers [ZXZL23] in additive manufacturing to enhance the efficiency and surface quality of the simultaneous machining strategy for four-axis CNC machines. Second, it would be useful to investigate a method that can integrate the decomposition of fabricable segments with stages of TSP linking into a single graph cut-based process. Third, it is important to explore a hybrid machining strategy that leverages the advantages of both positional and simultaneous machining strategies. Fourth, it would be useful to consider such additional physical factors as the stability of machining during subtractive manufacturing. Fifth, a more detailed examination of the problem of optimizing the orientation of the object is needed. Finally, future research in the area should seek to apply our method to prevalent CAM systems for four-axis CNC machines.

It is important to ensure manufacturability when evaluating the capacity of a four-axis CNC machine for fabrication. This leads to two further directions of research. First, there is a need to explore techniques of topological optimization that consider specific constraints related to manufacturability during the modeling process. Second, the problem of transforming shapes that cannot otherwise be fabricated into ones that can by using four-axis CNC machines, while minimizing variations in shape, poses a daunting challenge.

7. Acknowledgments

We thank the reviewers for their valuable comments and constructive suggestions. We thank Thingiverse and GrabCAD for providing the models used in this study. This work was supported by the National Key R&D Program 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 help in proofreading the manuscript.

8. Data Availability Statement

The data that support the findings of this study are available in the Supplementary Materials.

References

[AMG*19] ALDERIGHI, THOMAS, MALOMO, LUIGI, GIORGI, DANIELA, et al. "Volume-aware design of composite molds". *ACM Transactions on Graphics* (2019) 3.

[BBR*21] BARTON, MICHAEL, BIZZARRI, MICHAL, RIST, FLORIAN, et al. "Geometry and tool motion planning for curvature adapted CNC machining". *ACM Transactions on Graphics* 40.4 (2021), 1–16 3.

[CJ12] CHOI, BYOUNG K and JERARD, ROBERT B. *Sculptured surface machining: theory and applications*. Springer Science & Business Media, 2012 3.

[CRCM23] CUI, QIAODONG, RONG, VICTOR, CHEN, DESAI, and MATUSIK, WOJCIECH. "Dense, Interlocking-Free and Scalable Spectral Packing of Generic 3D Objects". *ACM Transactions on Graphics (TOG)* 42.4 (2023), 1–14 13.

[CÜ10] CAN, AHMET and ÜNÜVAR, ALI. "A novel iso-scallop toolpath generation for efficient five-axis machining of free-form surfaces". *The International Journal of Advanced Manufacturing Technology* 51 (2010), 1083–1098 3.

[DJ04] DING, SONGLIN and JIANG, RIDONG. "Tool path generation for 4-axis contour EDM rough machining". *International Journal of Machine Tools and Manufacture* 44.14 (2004), 1493–1502 3.

[EE18] EZAIR, BEN and ELBER, GERSHON. "Automatic generation of globally assured collision free orientations for 5-axis ball-end tool-paths". *Computer-Aided Design* 102 (2018), 171–181 3.

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

[FP09] FABRI, ANDREAS and PION, SYLVAIN. "CGAL: The computational geometry algorithms library". *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*. 2009, 538–539 9.

[FWJ06] FRANK, MATTHEW C, WYSK, RICHARD A, and JOSHI, SANJAY B. "Determining setup orientations from the visibility of slice geometry for rapid computer numerically controlled machining". (2006) 4.

[Gii23] GIIRESEARCH. *4-axis CNC Machining Center Market Forecasts to 2030*. https://www.giiresearch.com/report/smrc1371881-axis-cnc-machining-center-market-forecasts-global.html. 2023 2.

[GJ*10] GUENNEBAUD, GAËL, JACOB, BENOÎT, et al. *Eigen v3*. http://eigen.tuxfamily.org. 2010 9.

[HBA13] HILDEBRAND, KRISTIAN, BICKEL, BERND, and ALEXA, MARC. "Orthogonal slicing for additive manufacturing". *Computers & Graphics* 37.6 (2013), 669–675 3.

[HMA15] HERHOLZ, PHILIPP, MATUSIK, WOJCIECH, and ALEXA, MARC. "Approximating free-form geometry with height fields for manufacturing". *Computer Graphics Forum*. Vol. 34. 2. Wiley Online Library. 2015, 239–251 3.

[HPR*13] HOFFMAN, KARLA L, PADBERG, MANFRED, RINALDI, GIOVANNI, et al. "Traveling salesman problem". *Encyclopedia of operations research and management science* 1 (2013), 1573–1578 7.

[JLZ*21] JI, SHIJUN, LEI, LIANGGEN, ZHAO, JI, et al. "An adaptive real-time NURBS curve interpolation for 4-axis polishing machine tool". *Robotics and Computer-Integrated Manufacturing* 67 (2021), 102025 2.

[Jos15] JOSHI, ASHISH. "Computer aided process planning for multi-axis CNC machining using feature free polygonal CAD models". (2015) 3.

[LA06] LARSSON, THOMAS and AKENINE-MÖLLER, TOMAS. "A dynamic bounding volume hierarchy for generalized collision detection". *Computers & Graphics* 30.3 (2006), 450–459 13.

[LAYK21] LATIF, KAMRAN, ADAM, ANBIA, YUSOF, YUSRI, and KADIR, AINI ZUHRA ABDUL. "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 12.

[Lee03] LEE, EUNGKI. "Contour offset approach to spiral toolpath generation with constant scallop height". *Computer-Aided Design* 35.6 (2003), 511–518 5.

[LKLF21] LIANG, FUSHENG, KANG, CHENGWEI, LU, ZHONGYANG, and FANG, FENGZHOU. "Iso-scallop tool path planning for triangular mesh surfaces in multi-axis machining". *Robotics and Computer-Integrated Manufacturing* 72 (2021), 102206 3.

[LXG10] LASEMI, ALI, XUE, DEYI, and GU, PEIHUA. "Recent development in CNC machining of freeform surfaces: A state-of-the-art review". *Computer-Aided Design* 42.7 (2010), 641–654 2.

[MLS*18] MUNTONI, ALESSANDRO, LIVESU, MARCO, SCATENI, RICCARDO, et al. "Axis-aligned height-field block decomposition of 3D shapes". *ACM Transactions on Graphics (TOG)* 37.5 (2018), 1–15 3.

[MPE17] MACHCHHAR, JINESH, PLAKHOTNIK, DENYS, and ELBER, GERSHON. "Precise algebraic-based swept volumes for arbitrary free-form shaped tools towards multi-axis CNC machining verification". *Computer-Aided Design* 90 (2017), 48–58 3.

[MSJ*23] MA, HONG-YU, SHEN, LI-YONG, JIANG, XIN, et al. "A survey of path planning and feedrate interpolation in computer numerical control". *arXiv preprint arXiv:2303.01368* (2023) 4.

[NTM*21] NUVOLI, STEFANO, TOLA, ALESSANDRO, MUNTONI, ALESSANDRO, et al. "Automatic Surface Segmentation for Seamless Fabrication Using 4-axis Milling Machines". *Computer Graphics Forum*. Vol. 40. 2. Wiley Online Library. 2021, 191–203 2, 3, 5, 12, 13.

[PL14] PLAKHOTNIK, DENYS and LAUWERS, BERT. "Graph-based optimization of five-axis machine tool movements by varying tool orientation". *The International Journal of Advanced Manufacturing Technology* 74 (2014), 307–318 4.

[RSG09] REN, FEI, SUN, YUWEN, and GUO, DONGMING. "Combined reparameterization-based spiral toolpath generation for five-axis sculptured surface machining". *The international journal of advanced manufacturing technology* 40 (2009), 760–768 3.

[SCL*04] SORKINE, OLGA, COHEN-OR, DANIEL, LIPMAN, YARON, et al. "Laplacian surface editing". *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*. 2004, 175–184 9.

[Sie16] SIEMENS. *NX Software*. https://plm.sw.siemens.com/en-US/nx/. 2016 4.

[SJ06] SWINBANK, RICHARD and 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 5.

[SKM*22] SATHISH, K, KUMAR, S SENTHIL, MAGAL, R THAMIL, et al. "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 3, 4, 12.

[STC09] SCHMIDT, FRANK R., TOPPE, ENO, and CREMERS, DANIEL. "Efficient planar graph cuts with applications in Computer Vision". *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, 351–356. DOI: 10.1109/CVPR.2009.5206863 3, 7, 13.

[VD15] VEKSLER, OLGA and DELONG, ANDREW. *gco-v3.0*. https://github.com/nsubtil/gco-v3.0. 2015 9.

[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 2.

[Wor23] WORKS, SAUNDERS MACHINE. *Fusion 360*. https://www.nyccnc.com/simultaneous-4th-axis/. 2023 3, 4, 12.

[XGD*18] XU, JING, GU, XIZHI, DING, DONGHONG, et al. "A review of slicing methods for directed energy deposition based additive manufacturing". *Rapid Prototyping Journal* 24.6 (2018), 1012–1025 13.

[YJJ*22] YUWEN, SUN, JINJIE, JIA, JINTING, XU, et al. "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 3.

[ZCW16] ZAHID, MN OSMAN, CASE, KEITH, and WATTS, DM. "Cutting orientations for non-complex parts in 4th axis machining". *IOP Conference Series: Materials Science and Engineering*. Vol. 114. 1. IOP Publishing. 2016, 012013 3.

[ZFH*22] ZHANG, TIANYU, FANG, GUOXIN, HUANG, YUMING, et al. "S3-slicer: A general slicing framework for multi-axis 3D printing". *ACM Transactions on Graphics (TOG)* 41.6 (2022), 1–15 13.

[ZGH*16] ZHAO, HAISEN, GU, FANGLIN, HUANG, QI-XING, et al. "Connected fermat spirals for layered fabrication". *ACM Transactions on Graphics (TOG)* 35.4 (2016), 1–10 4.

[Zha24] ZHAO, HAISEN. *Libhgp*. https://github.com/haisenzhao/libhgp. 2024 9.

[ZRZ23] ZHANG, WENBO, REN, JUNXUE, and ZHOU, JINHUA. "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* (2023), 1–29 2.

[ZXZL23] ZHONG, FANCHAO, XU, YONGLAI, ZHAO, HAISEN, and LU, LIN. "As-continuous-as-possible Extrusion-based Fabrication of Surface Models". *ACM Transactions on Graphics* 42.3 (2023), 1–16 4, 13.

[ZZX*18] ZHAO, HAISEN, ZHANG, HAO, XIN, SHIQING, et al. "DSCarver: decompose-and-spiral-carve for subtractive manufacturing". *ACM Transactions on Graphics (TOG)* 37.4 (2018), 1–14 3.