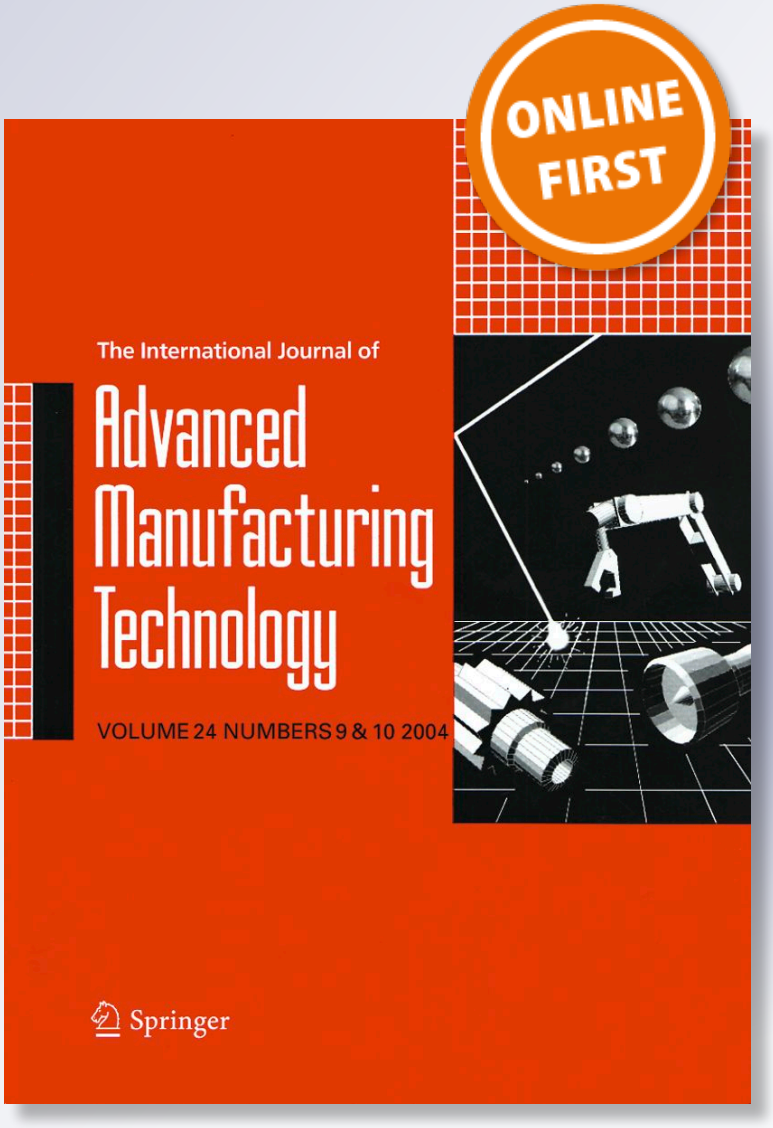
Compact radial zigzag for five-axis machining of STL surfaces

L. V. Dang, K. Vacharanukul & S. S. Makhanov

The International Journal of Advanced Manufacturing Technology

ISSN 0268-3768

Int J Adv Manuf Technol
DOI 10.1007/s00170-019-03897-7



Springer

The International Journal of Advanced Manufacturing Technology
https://doi.org/10.1007/s00170-019-03897-7

ORIGINAL ARTICLE

Compact radial zigzag for five-axis machining of STL surfaces

L. V. Dang[1] · K. Vacharanukul[2] · S. S. Makhanov[1]

Received: 13 August 2018 / Accepted: 20 May 2019

Abstract
The paper presents a new method to generate efficient milling toolpaths for five-axis sculptured surface machining in an important case when the vector field of preferred directions (VFPD) forms a star-like, radial configuration. To optimize the toolpath, a new modification of the radial toolpath aligned with the VFPD called the compact radial zigzag (CRZ) has been proposed, analyzed, and verified practically. The CRZ is combined with transfinite interpolation (TFI) to treat an irregular VFPD. The method is designed for the machining of industrial stereolithography (STL) part surfaces characterized by complex geometries and sharp extrema. A demo of the algorithm is at https://drive.google.com/open?id=1OM_z4cAOUqGu2RPAzkZOIBcEnfptdTq7.

Keywords Material removal rate · Five-axis milling machine · Radial toolpath · STL · Vector field · Redundancy · Moment invariants

1 Introduction

High-accuracy complex freeform surfaces are used by many industries, such as aerospace, automotive, shipbuilding, and medical implant companies. The parts are usually required to be smooth, but they often include valleys, saddle points, and sharp extrema. Nowadays, such parts as turbine blades or high-precision molds are almost exclusively produced on five-axis machines since they are the most accurate and reliable. Toolpath generation for the five-axis machining of complex-shaped parts is a subject of extensive research in computer numerical control (CNC)–based manufacturing.

The ultimate goals of toolpath optimization are to reduce the machining time and increase accuracy. Among the most common methods of achieving these (contradictory) goals is the selection of a suitable toolpath pattern, reducing the redundancy of the path, reducing the kinematic error, and controlling the scallops between the consecutive tracks. The toolpath algorithm must also decide whether the toolpath is a single continuous curve or that it can be subdivided into patches to optimize the quality.

This introduction considers topological aspects of toolpath generation and explains why the proposed compact radial zigzag (CRZ) toolpath is a suitable option for a certain class of sculptured surfaces. More surveys on five-axis toolpath generation can be found in [1–3].

The basic five-axis toolpath patterns are the zigzag (ZZ), spiral, contour parallel path (CP), and their modifications [2]. Their advantages are simple geometry, a small number of turns, and a straightforward way of adapting the pattern locally. A great deal of work has been done to improve and modify these patterns, such as the variety of iso-parametric, iso-scallop, and iso-planar methods.

Unfortunately, most of these methods can be applied only to parametric surfaces. Moreover, when the optimization includes the kinematics of the machine, collision avoidance, machinability of the surface, and the workpiece setup, the computational complexity of the problem becomes prohibitively high. Therefore, even advanced toolpath optimizations consider only a subset of the control parameters and the cost (utility) functions, assuming (often wrongly) that the impact of the remaining parameters is negligible or that they can be improved by a trial-and-error process.

An iso-scallop [4] and isoplanar [5] tool path intersects the surface by parallel planes generating iso-curves. The side step is evaluated based on the scallop height constraints. The method is robust and is common in commercial systems. It applies to

✉ S. S. Makhanov
makhanov@siit.tu.ac.th

[1] Sirindhorn International Institute of Technology, Thammasat University, Rangsit, Pathum Thani 12120, Thailand

[2] National Institute of Metrology (Thailand) (NIMT), Khlong Luang, Pathum Thani 12120, Thailand

Published online: 02 September 2019

compound, trimmed, NURBS, and stereolithography (STL) models. However, the algorithm often leads to conservative path intervals in an attempt to ensure the allowable scallop height.

The iso-scallop toolpaths [6–8] are based on the offsetting of an initial curve (often the boundary [9]). The idea is to minimize the overlap between tool postures of the neighboring cutter contact (CC) curves by taking a maximum possible offset and reducing the number of required tracks. Usually, ball-end cutters or flat-end cutters are used. Chiou and Lee [10] extend the method to a generalized cutter. Liu et al. [11] present an iso-scallop toolpath for a triangular mesh (STL surface). The recent iso-planar path (ISOP) [12] considers both the cutting strip width and the machine's kinematic capacities. The machining efficiency is improved by iteratively adjusting the tool orientations.

The iso-level method [13] is a combination of the iso-scallop path and the requirement of smoothness of the CC curves to maintain a high feed rate. Some energy functionals for generating such a toolpath have been derived and analyzed.

The seminal work of Kim and Sarma [14] introduces the vector field of preferred directions (VFPD). The part surface is covered by machining patches, each characterized by an optimal direction, corresponding to the maximum machining strip. The discrete VFPD is then approximated by a continuous vector field, and the corresponding streamlines are constructed. The streamlines define the topology of the *optimal* toolpaths. The algorithm partitions the streamlines, based on the so-called basins of attraction (an idea borrowed from the theory of dynamic systems).

Chiou and Lee [10] split the part surface into machining patches. For each patch, at least two preferred directions producing the maximum machining strip are established, generating the desired VFPD. Next, instead of generating the streamlines, the algorithm generates a basic *initial path*, producing the largest average machining strip. The entire toolpath is then constructed by offsetting the initial path and propagating the offsets inside the region. The basic path is re-initialized when the current offset deviates considerably from the VFPD.

Therefore, the basic VFPD approach includes (1) constructing the VFPD, (2) clustering the surface (optional), and (3) constructing the toolpath for each cluster, using a suitable pattern.

The VFPD for five-axis machining critically depends on the optimization problem, which may include several utility functions and possible constraints. Usually, the optimization reduces the total length of the toolpath or the total machining time while maintaining the prescribed accuracy. Alternatively, the user may improve the accuracy while keeping or even reducing the machining time. The number of CC points may be reduced for high-speed milling when the time to move from one cutter location (CL) point to the next becomes shorter than the servo update interval. The VFPD often includes geometric constraints, kinematic performance of the machine, capacity of the machine axis, cutting force limits, and cutting conditions. Clearly, the VFPD depends on the pattern of the toolpath. In contrast, a suitable toolpath pattern depends on the topology and geometrical structure of the VFPD. The general VFPD-aligned toolpath is, therefore, a non-trivial, high-computational complexity problem which may have a variety of mathematical formulations.

The VFPD proposed in [10, 14] is based the machining strip, the increase of which implies the reduction of the toolpath. However, minimization of the toolpath in the workpiece coordinates does not necessarily minimize it in the machine coordinates and does not necessarily minimize the machining time. "Explicitly, a toolpath with minimum total toolpath length may turn out to be inferior when the specific machine tool's capacities are considered, and in order not to exceed the limits of those capacities, the machine's controller has to keep the feed rate under an inordinately low level, thus actually prolonging the real machining time" [15].

The seminal works of Makhanov [16], Makhanov and Ivanenko [17], and later Bieterman and Sandstrom [18] introduce adaptive curvilinear coordinate systems to construct curvilinear zigzag toolpaths for complex-shaped boundaries and surfaces with islands (complex pocket milling). The corresponding energy functionals provide the smoothness of the toolpath, and adaptation to the boundaries and to the regions with large kinematic errors. They also establish constraints to reduce the redundancy of the path by penalizing too small or too large machining strips between the consecutive tool tracks. However, since the adaptive curvilinear coordinates must be topologically equivalent to a Cartesian coordinate system, the application of this method is limited to surfaces which can be easily parameterized.

In order to include the machine kinematics, Hu and Tang [15] and Chen et al. [19] introduce a machine-dependent VFPD based on the effective machining removal rate (EMRR). The EMRR is defined as the product of the machining strip and the feed rate, evaluated numerically in the workpiece coordinate system (WCS). The feed rate is subject to the velocity and acceleration limits of a particular machine. There are ten such constraints defined at each CC point for the machine's five axes. Fitting the CC curves to the corresponding EMRR streamlines is performed by offsetting an initial curve using an extended version of Chiou and Lee's approach. This approach recursively alternates between the points with the maximum EMRR and the standard iso-cusp height expansion scheme. The resulting five-axis toolpath achieves, for some complex freeform part surfaces, substantial savings in total machining time over the existing toolpath generation algorithms.

Hu et al. [12] (ISOP) extend this approach by including five additional constraints imposed on possible jerks of the machine axis (totaling 15 constraints). The ISOP expansion methodology is combined with a special index, defining one principal direction to drive parallel planes, to maximize the overall EMRR on the entire surface.

Moodleah et al. [20] use a simplified version of the EMRR, assuming that only the rotation axes exceed the machine speed limit and that the workpiece is machined with the maximum linear speed allowed for the particular material. The simplified EMRR is combined with a curvilinear zigzag that is aligned with the prescribed VFPD. The method has been verified on complex-shaped STL surfaces, e.g., a human face mask and models of human teeth (incisor, premolar, molar, and canine). The disadvantage of this approach is a possible abrupt stop of the tool by the controller to comply with the machine acceleration limits. This sudden stop may leave unexpected marks on the surface. However, the algorithm can be applied to semi-finish cuts with an additional manual polishing or an additional fine cut of the surface.

Many preceding generation methods based on the maximization of the machining strip do not consider a simple radial zigzag (SRZ) (see Fig. 1a) due to its redundancy around the pole. Some papers completely exclude this pattern from the set of the standard topologies. For instance, Kim and Sarma's design is not suitable for SRZ. Zou et al. [13] observe only three basic patterns: direction parallel, contour parallel, and spiral. A survey [2] mentions that "based on the strengths and limitations of the different path patterns investigated above, the direction parallel and contour parallel paths are considered the most widely used ones due to their simplicity and adaptability in engineering applications." The recent advanced toolpath generation methods based on the machining strip tensor flow [21, 22] are also designed in such a way that the SRZ has been practically excluded or requires complicated geometric transformations.

The methods that are based on the EMRR and the kinematic constraints show that many STL surfaces characterized by sharp extrema generate a star-like (radial) VFPD. Of course, moving along the radial directions is inevitably redundant. However, "a collection of individually-efficient cuts does not guarantee the quickest machining of the surface" [14]. The radial path is not necessarily the worst in a global sense although it could be the worst choice locally. One of the simplest examples is a cone (Fig. 1). An analysis of the machining time for the contour parallel pattern and radial pattern reveals that the time depends on the height and the radius of the cone, the redundancy of the radial lines, and the rotations required to perform the alternative circular cut. In other words, the radial path often wins on machines with slow rotation axes and fast linear axes.

Furthermore, the increasing complexity of industrial five-axis parts makes it virtually impossible to generate a single CC curve that follows the VFPD. Therefore, an efficient cutting strategy could include the decomposition of the VFPD. Such decomposition includes radial patterns formed around peaks and saddle points.

A decomposition-based strategy to cut a complex STL surface includes the following:

- Flattening (surface parametrization, see Section 3.1)
- Selection of the cost function and constraints
- Generation of the VFPD

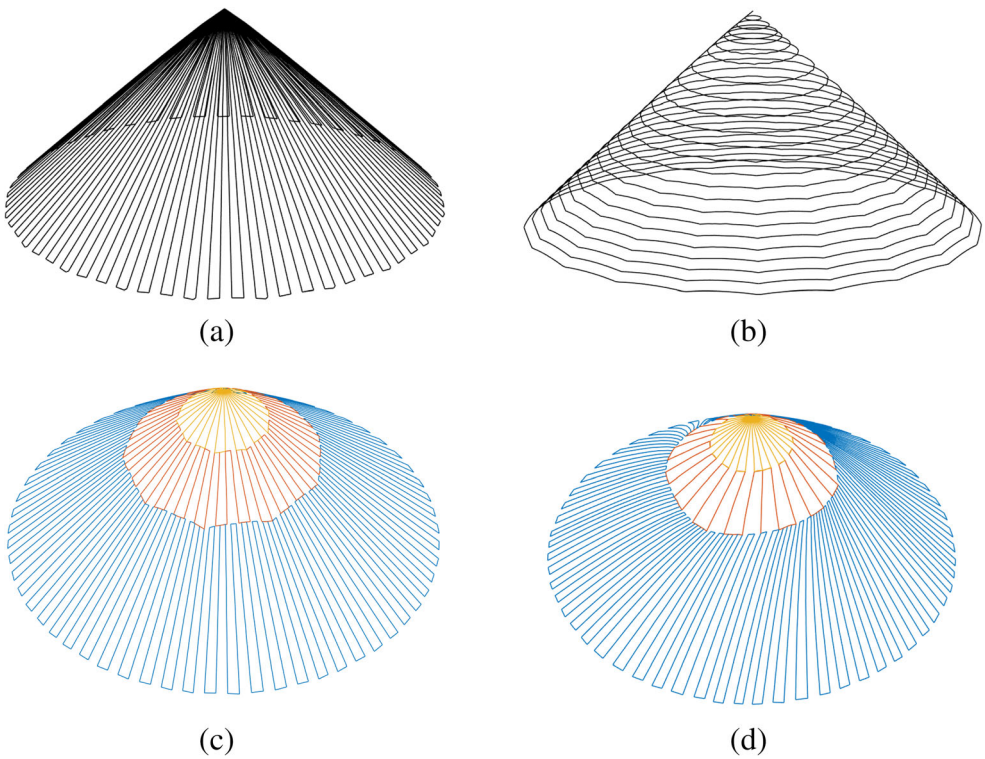


Fig. 1 Examples of toolpath patterns on a conical surface. **a** Simple radial zigzag (SRZ). **b** Contour parallel path (CP). **c** Compact radial pattern (CRZ). **d** CRZ pattern with transfinite interpolation (CRZT)

- Decomposition of the VFPD into clusters
- Classifying the clusters. On the most basic level, we have the ZZ, the contour/spiral (CP), and the SRZ
- Toolpath generation, including CRZ
- Merging the toolpaths at the boundaries of the clusters

It should be noted that there exists a variety of advanced techniques to decompose the part surfaces into subregions to cut the surfaces patch by patch. Examples are segmentations based on the complexity of the CC points [23], Gaussian curves [24, 25], clustering of the normal vectors [5], and combining the orientation of the normal vectors with the principle of a minimum tilt angle [26]. A tensor field [21, 22] and the streamlines of a VFPD [14] have been used for surface decomposition and toolpath generation.

However, to the best of our knowledge, the use of the radial pattern has been completely overlooked. The detection and the generation of such a toolpath have not been properly analyzed and verified.

Therefore, the focus of this paper is radial patterns. We detect them, isolate them, and apply a new toolpath generation algorithm called the compact radial zigzag based on the partition of the radial pattern into layers, to reduce the redundancy.

We show that in many cases, the CRZ outperforms the circular, spiral, zigzag, purely radial pattern, and recent isoplanar toolpath generation methods in terms of the machining time and accuracy.

An example of a CRZ toolpath is shown in Fig. 1c and d. Note that the new pattern is redundant and includes additional turns. However, in many cases, moving along the radial direction combined with the reduction of the redundancy outperforms conventional patterns. In this paper, we also introduce an improved version of the CRZ (called CRZT), based on fitting the radial tracks to the optimal directions using transfinite interpolation (TFI) [27]. TFI is able to treat patterns that are only approximately radial (not converging to one point, but converging to a relatively small region on the surface).

The CRZ is applied to the STL surfaces with a variety of cost functions such as the EMRR, kinematic error, and variation of the rotation angles. The machining results show that the algorithm outperforms conventional machining strategies for surfaces characterized by high curvatures around the peak.

CRZ has been tested against NX11 (formerly UG) [28] using the helical-and-spiral (HS) pattern and the follow periphery (FP) pattern. It has also been tested against a recent ISOP [12] and the conventional radial toolpaths. The numerical tests and practical machining show that the CRZ can reduce the machining time by 77%, while keeping the kinematic error and the scallops within the prescribed accuracy range. Our practical machining (Appendix) reveals that the roughness of the produced semi-finish cut is similar to that of the competing methods.

2 Background

This section introduces the basics of five-axis kinematics, kinematic error, and the VFPD based on the EMRR.

2.1 Five-axis milling machine

A five-axis milling machine is controlled by a sequence of commands (G code), composed of three Cartesian coordinates of the tooltip and two rotation angles, to establish the orientation of the tool. Denote the first rotary axis by B and the second rotary axis by A (Fig. 2). Introduce the following coordinate systems: workpiece coordinates (O_1), coordinate system of the first rotary part (O_2), coordinate system of the second rotary part (O_3), and coordinate system of the spindle

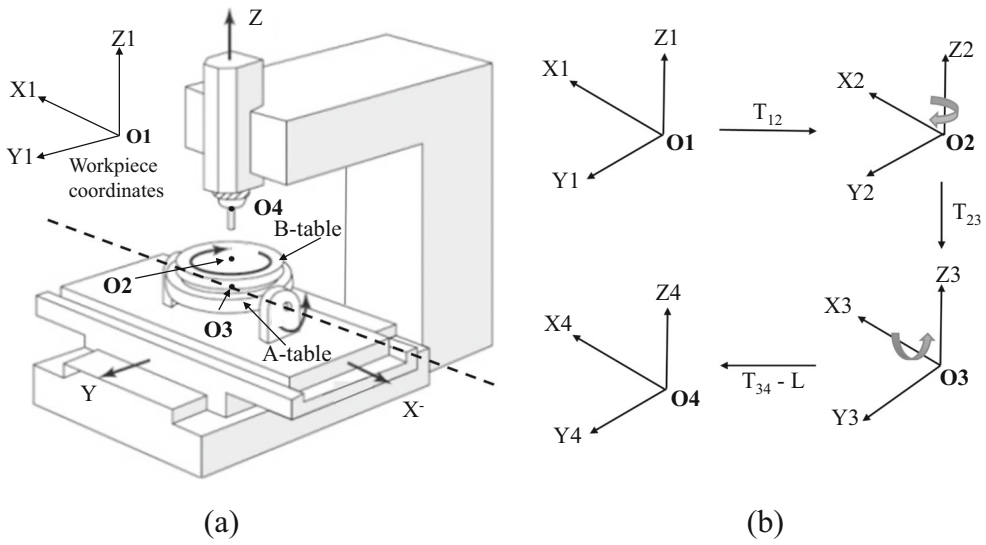


Fig. 2 Five-axis milling machine with the rotary axes on the table. **a** Haas VF-2TR. **b** Kinematic transformation

(O_4). The configuration of a five-axis milling machine is characterized by the following:

- Rotation matrices A and B, corresponding to the two rotary axes
- Translations T_{12}, T_{23}, and T_{34}, where T_{12} is the coordinate of the workpiece origin in the B-axis coordinate system, T_{23} is the coordinate of the B-axis origin in the A-axis coordinate system, and T_{34} is the coordinate of the A-axis origin in the spindle coordinate system
- The length of the tool (L) is treated as an additional translation (T_4), where T_4 is equal to $(0, 0, L)$ or $(0, 0, -L)$, depending on the orientation of the z-axis. Figure 2 displays the configuration used in this study.

2.2 Kinematic error

The coordinate transformation from the WCS to the machine coordinates is given by the kinematic transformation as follows:

$$M = \Re(a, b, W) \equiv A[a](B[b](W + T_{12}) + T_{23}) + T_{34} - T_4, \quad (1)$$

where $W = (x, y, z)$ and $M = (X, Y, Z)$ denote the workpiece and the machine coordinates, respectively, and a and b are the rotation angles required to establish a prescribed orientation of the tool. For instance, a ball nose tool is often inclined by a small angle of 5–15° [12], relative to the normal of the surface in the plane (n, f). A flat-end tool is also inclined to match the curvature of a surface [29].

The inverse transformation is given by

$$W = \Re^{-1}(a, b, M) \equiv B^{-1}[b]\left(A^{-1}[a](M + T_4 - T_{34}) - T_{23}\right) - T_{12}. \quad (2)$$

Let $W^D_{p,p+1}(t) \equiv W^S\left(t, s_p, s_{p+1}\right)$ be a space curve between tool positions W_p and W_{p+1} extracted from the machined surface $S(u, v)$, where $t \in [s_p, s_{p+1}]$ is a parametric coordinate along the curve. This curve is called the desired trajectory. The desired and the actual trajectory can differ drastically. The distance between them is machine dependent and is called the kinematic error (Fig. 3).

The tool inclination implies that the CC points are different from the CL points (Fig. 4). However, the machine understands only the CL points. Therefore, the G code needs an additional coordinate transformation between the local basis and the WCS. The local basis with the origin at the CC point is given by (n, f, τ), where n is the surface normal vector, f is the normalized projection of the direction vector connecting two consecutive CC points on the tangent plane, and $\tau = \frac{f \times n}{\|f \times n\|}$ (see Fig. 4). Coordinates of the CL points in the WCS are given by [30]

$$\mathrm{CL} = \mathrm{CC} + R_{\mathrm{local}} R_{\mathrm{T}} R_{\mathrm{I}} T_{\mathrm{tool}} \quad (3)$$

with $T_{\mathrm{tool}} = [-r, 0, 0]$ for a flat-end mill and $T_{\mathrm{tool}} = [0, 0, r]$ for a ball-end mill, where r is the radius of the tool and R_{I} and R_{T} are the rotation matrices corresponding to the inclination and tilt angle. R_{local} is the transformation matrix composed of the local basis vectors given by

$$R_{\mathrm{local}} = \begin{pmatrix} f_x & \tau_x & n_x \\ f_y & \tau_y & n_y \\ f_z & \tau_z & n_z \end{pmatrix}. \quad (4)$$

The actual trajectory $W_{p,p+1}$ is obtained using Eq. (2). First, for every $W_p \in W^D$, the transformation $M_p = \Re\left(\Re_p, W_p\right)$ is invoked. Second, the rotation angles $\Re \equiv \Re(t)$ and the machine coordinates $M = (X(t), Y(t), Z(t))$ are assumed to change linearly between the prescribed points s_p and s_{p+1} as follows:

$$\begin{aligned} M_{p,p+1}(t) &= L_{p+1}(t) M_{p+1} + L_p(t) M_p, \\ \Re(t) &= L_{p+1}(t) \Re_{p+1} + L_p(t) \Re_p, \end{aligned} \quad (5)$$

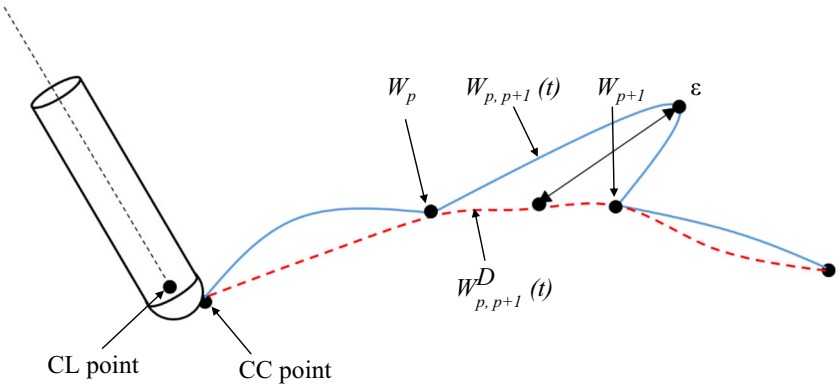


Fig. 3 Kinematic error: the desired trajectory versus the actual trajectory

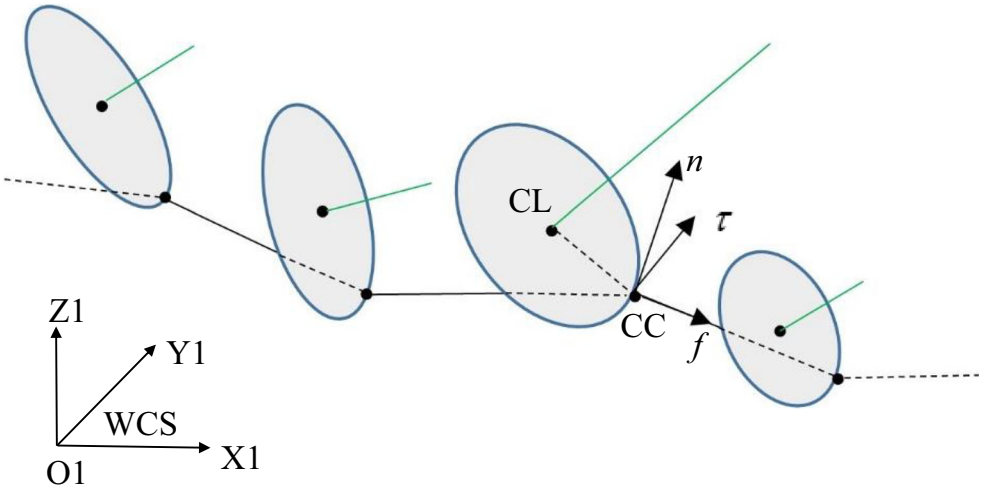


Fig. 4 Flat-end cutter. Transformation from CC to CL points

where $L_{p+1}(t)=\frac{t-s_p}{s_{p+1}-s_p}$, $L_p(t)=\frac{s_{p+1}-t}{s_{p+1}-s_p}$, $s_p\le t\le s_{p+1}$, and $\mathfrak{R}=(a,b)$ is a pair of the rotation angles. Transforming M back to W for every t yields the actual trajectory of the tool tip in the WCS

$$W_{p,p+1}(t)=\mathfrak{K}^{-1}\left(\mathfrak{R}(t),M_{p,p+1}(t)\right)= \mathfrak{K}^{-1}\left(L_{p+1}(t)\mathfrak{R}_{p+1}+L_p(t)\mathfrak{R}_p,L_{p+1}(t)M_{p+1}+L_p(t)M_p\right). \tag{6}$$

To represent the toolpath in terms of the WCS, we eliminate M_p and M_{p+1} by $M_p=\mathfrak{K}\left(\mathfrak{R}_p,W_p\right)$ and $M_{p+1}=\mathfrak{K}\left(\mathfrak{R}_{p+1},W_{p+1}\right)$.

Substituting M_p and M_{p+1} yields

$$W_{p,p+1}(t)=\mathfrak{K}^{-1}\left(L_{p+1}(t)\mathfrak{R}_{p+1}+L_p(t)\mathfrak{R}_p,L_{p+1}(t)\mathfrak{K}\left(\mathfrak{R}_{p+1},W_{p+1}\right)+L_p(t)\mathfrak{K}\left(\mathfrak{R}_p,W_p\right)\right). \tag{7}$$

Finally, the kinematic error is measured by the parameterization-invariant Hausdorff distance given by

$$\mathrm{dist}_H\left(W^D,W\right)=\max\left(\max_{a\in W^D(t)}\min_{b\in W(t)}\|a-b\|,\max_{a\in W(t)}\min_{b\in W^D(t)}\|a-b\|\right), \tag{8}$$

where $\|\cdot\|$ denotes the Euclidean distance. The total kinematic error is the sum of the Hausdorff distances between the desired trajectory $W^D_{p,p+1}(t)\equiv W^D\left(s_p,s_{p+1},t\right)$ and the actual trajectory $W_{p,\,p+1}(t)=W(s_p,s_{p+1},t)$.

$$\varepsilon=\sum_p \mathrm{dist}_H\left(W^D_{p,p+1}(t),W_{p,p+1}(t)\right). \tag{9}$$

The average kinematic error is then defined by

$$\varepsilon'=\frac{\varepsilon}{N_{\mathrm{CC}}-1}, \tag{10}$$

where N_{CC} is the number of the CC points.

In many cases, the machining quality is defined by the maximum error given by

$$\varepsilon_{\max}=\max_p\left(\mathrm{dist}_H\left(W^D_{p,p+1}(t),W_{p,p+1}(t)\right)\right). \tag{11}$$

The EMRR between two consecutive CC points (W_1 and W_2) is defined as the material removed per unit time when the tool moves from W_1 to W_2. An increase of the EMRR should reduce the machining time if there is no redundancy. Evaluation of the EMRR for a given type of milling machine has been developed in [12, 15, 19, 31]. We follow a simplified procedure [20, 32], which considers only the velocities of the machine axes. For a complete model, including the acceleration and jerks, we refer the reader to [12]. Let W_0 be an arbitrary CC point on the surface, a_0 and b_0 the corresponding rotation angles, and let Ω_{W_0} be a geodesic circle with the center at W_0 and radius d_0. Let us sample Ω_{W_0} so that $W_i\in\Omega_{W_0}, i=1,2,\ldots,m_0$. Note that $\Omega_{M_0}=\mathfrak{K}(\Omega_{W_0})$, where $M_0=\mathfrak{K}(W_0,a_0,b_0)$ is an irregular closed curve. Denote the feed rate by F and the machining strip at W_0 in the $\overrightarrow{W_0W_i}$ direction by $l_{0,i}$. The EMRR in the direction $\overrightarrow{W_0W_i}$ is given by

$$R_{0,i} = Fl_{0,i}. \tag{12}$$

Let $v_x^*, v_y^*, v_z^*, v_a^*, v_b^*$ be the maximum speed of the corresponding axis.

Consider $M_i \equiv (X_i, Y_i, Z_i) = \Re(W_i, a_i, b_i)$. The machining time from M_0 to M_i is given by [12]

$$t = \frac{d_0}{F} = \frac{X_i - X_0}{v_x} = \frac{Y_i - Y_0}{v_y} = \frac{Z_i - Z_0}{v_z} = \frac{a_i - a_0}{v_a} = \frac{b_i - b_0}{v_b}, \tag{13}$$

where v_x, v_y, v_z, v_a, and v_b are the actual speeds of the corresponding axis.

Hence,

$$v_x = F\frac{X_i - X_0}{d_0}, v_y = F\frac{Y_i - Y_0}{d_0}, v_z = F\frac{Z_i - Z_0}{d_0}, \tag{14}$$
$$v_a = F\frac{a_i - a_0}{d_0}, v_b = F\frac{b_i - b_0}{d_0}.$$

Consequently, F in Eq. (12) is replaced by

$$F_{0,i} = \min\left(\frac{v_x^* d_0}{X_i - X_0}, \frac{v_y^* d_0}{Y_i - Y_0}, \frac{v_z^* d_0}{Z_i - Z_0}, \frac{v_a^* d_0}{a_i - a_0}, \frac{v_b^* d_0}{b_i - b_0}\right). \tag{15}$$

Thus,

$$R_{0,i} = F_{0,i} l_{0,i}. \tag{16}$$

We call the direction $\overrightarrow{W_0 W^*}$ optimal if $W^* = \text{argmax}_i\left(R_{0,i}\right)$. In other words, W^* maximizes the EMRR in the set of sample directions. Note that for any internal CC point, there are at least two opposite optimal directions. This creates a 3D bidirectional VFPD or the orientation field (OF). The total machining time is then given by

$$T = \sum_{k=1}^{N_{CC}-1} t_k, \tag{17}$$

where $t_k = \frac{l_{CC_k}}{F_k}$ (where l_{CC_k} is the length of the machining curve between CC_k and CC_{k+1} and F_k is the actual feed rate) and N_{CC} is the number of CC points.

3 Parameterization of the STL surface and generation of the toolpath

To detect the star-like regions, we flatten the STL surface onto a parametric domain. The EMRR vector field on the surface is transferred into the parametric domain, generating a 2D OF. The star configurations are detected by rotationally invariant moments [33]. To reduce the redundancy, the new CRZ toolpath consists of layers with a varying angular step between the tracks. The CRZ can be combined with TFI to treat irregular configurations.

3.1 Parametrization of the STL surface

To create a 2D OF and generate the required toolpath, we flatten (parameterize) the STL surface. The parameterization is a bijective (invertible) map between the surface and a triangulated planar domain (D), where each point in D corresponds to exactly one point on the surface. The OF is accordingly mapped onto D.

Maps that minimize the angular distortion, or shear, are called conformal (harmonic) maps. Riemann's theorem guarantees that a conformal map always exists for a smooth surface on a simply connected domain [34]. Therefore, one can argue that with a sufficient number of vertices, the STL surface can be flattened, with a little angular distortion [35].

One of the most popular flattening algorithms is the graph embedding method [36]. The boundary vertices are mapped onto the boundary of the (convex) planar region. The positions of the remaining vertices are obtained by solving a linear system

$$\begin{cases} \Phi u = 0 \\ \Phi v = 0 \end{cases}, \tag{18}$$

where

$$\Phi_{i,j} = \begin{cases} -\sum_{k \neq i} \Phi_{i,k}, \text{if } i = j, \\ \omega_{i,j}, \text{if } (i,j) \in E, \\ 0, \quad \text{otherwise.} \end{cases} \tag{19}$$

If the weights $\omega_{i,j}$ defined for each edge (E) are positive and the matrix is symmetric, the mapping is guaranteed to be bijective [35].

Currently, several free-boundary parameterization mappings have been proposed, such as the least-square conformal map [37], ABF++ [38], and the harmonic map [39]. We use a shape-preserving parameterization [40] derived from solutions of linear systems similar to Eq. (18), based on convex combinations. The algorithm generates a well-behaved smooth mapping onto a rectangle or a circle.

The parameterization translates the problem of toolpath generation from 3D to 2D. The CC points are arranged in a planar region (D) in the (u, v)-plane, and the final toolpath is obtained through inverse mapping [41–44].

3.2 Detection of the radial patterns

The SRZ is a particular case of the ZZ path in which one boundary is represented by a single point, while the other boundary is an open or a closed curve (Fig. 5).

The EMRR approach often generates star-like configurations which require a radial path (see Fig. 6). Such patterns can be detected and localized.

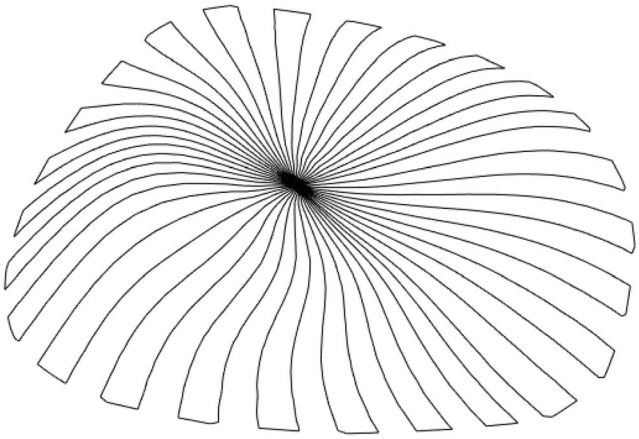


Fig. 5 Radial zigzag toolpath

There exist a number of algorithms to detect radial patterns. The most popular are the phase portrait analysis [45], moment invariants [46], Rankine model of flows [47], linear affine model [48], and topological analysis [49]. Unfortunately, a majority of the methods are designed for a classical VFPD where each vector has a single direction. In our case, we deal with an OF. Therefore, the detector must be *flip invariant*. In other words, if the direction of a vector changes to the opposite direction, the moment must retain the original value. Also, the detector must be scale, translation, and rotation invariant.

Let us represent the EMRR vector by the complex number $R_u + iR_v$, defined in the local coordinate system with the origin at the CC point. Define

$$\theta(u,v) = \begin{cases} \arctan\left(\dfrac{R_v}{R_u}\right), \text{ if } R_u \neq 0, \\ -\dfrac{\pi}{2}, \quad \text{otherwise.} \end{cases} \tag{20}$$

Clearly, psi is independent on the direction of the vector $R_u + iR_v$. Therefore, $f(u,v) = e^{i\theta(u,v)}$ is flip invariant. The OF is defined on a 2D triangular mesh in the parametric domain (u,v). Complex moment c_{pq} is given by [33, 50]

$$c_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (u+iv)^p (u-iv)^q f(u,v)\,du\,dv. \tag{21}$$

We replace the integral $\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}$ by an integral over a sampling circle with radius (R_s). The corresponding orientations inside

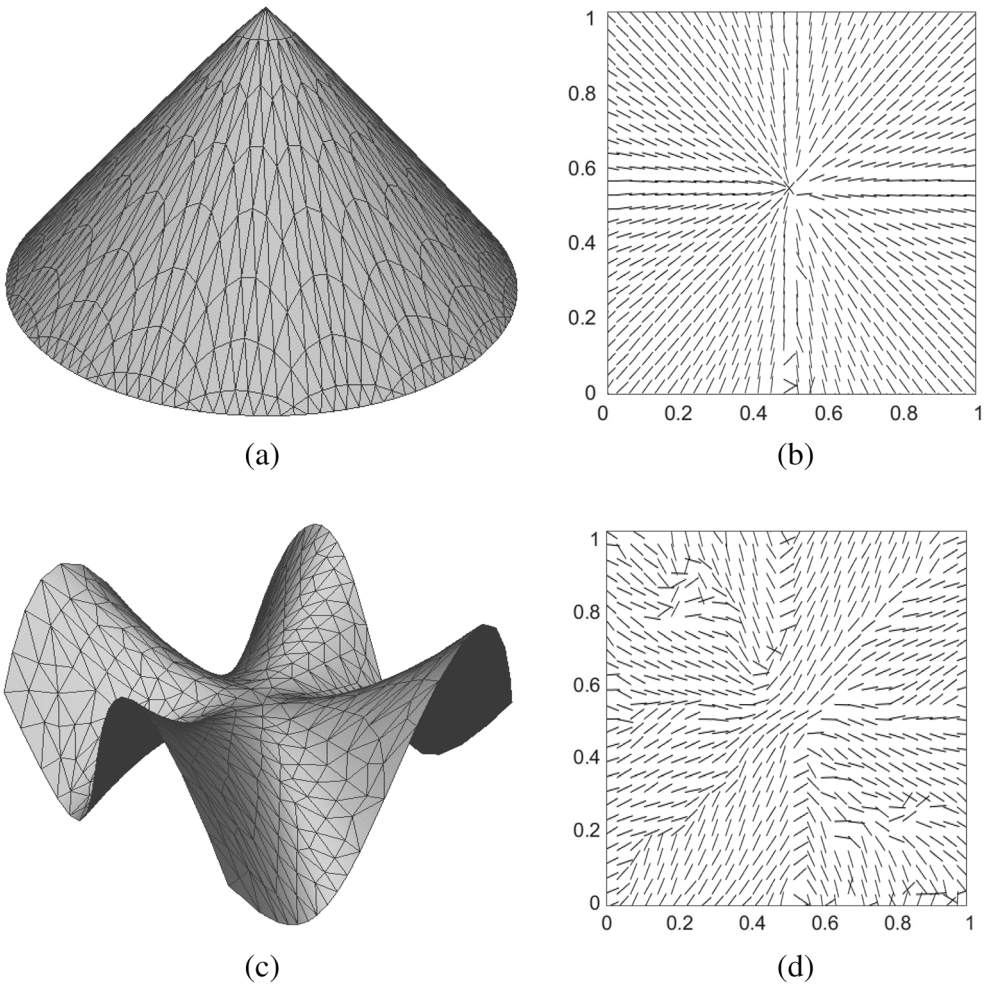


Fig. 6 Examples of radial OFs. **a** Cone. **b** OF of the EMRR, cone. **c** Saddle. **d** OF of the EMRR, saddle surface

R_s are evaluated by barycentric interpolation [51]. In polar coordinates

$$c_{pq} = \int_0^{R_s} \int_0^{2\pi} r^{p+q+1} e^{i(p-q)\varphi} f(r,\varphi) dr d\varphi, \tag{22}$$

Substituting $f(u, v) = e^{i\theta(u,v)}$ yields

$$c_{pq} = \int_0^{R_s} \int_0^{2\pi} r^{p+q+1} e^{i(p\varphi - q\varphi + \theta(r,\varphi))} dr d\varphi. \tag{23}$$

Note that $0 \le \varphi \le 2\pi$, whereas $-\frac{\pi}{2} \le \theta < \frac{\pi}{2}$.

If $(r, \varphi) = e^{i\theta(r,\varphi)}$ is rotated by angle α, $f_{\text{new}}(r, \varphi) = e^{i(\theta(r,\varphi) - \alpha)}$. Clearly, $c_{pq,\ \text{new}} = e^{i(p-q+1)\alpha} c_{pq}$. Therefore, any moment constructed as

$$\prod_{j=1}^{n} c_{p_j q_j} \text{ with } \sum_{j=1}^{n} \left(p_j - q_j + 1\right) = 0 \tag{24}$$

is rotationally invariant [46].

A change of variables $\hat{u} = u - \overline{u}$ and $\hat{v} = v - \overline{v}$ and normalizing c_{pq} imply the translation and scale invariance as follows:

$$c_{pq} = \frac{1}{\nu^{\gamma}} \int_0^{\infty} \int_0^{\infty} (\hat{u} + i\hat{v})^p (\hat{u} - i\hat{v})^q f(\hat{u}, \hat{v}) d\hat{u} d\hat{v}, \tag{25}$$

where

$$\gamma = \frac{p+q+2}{2}, \nu = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \chi(\hat{u}, \hat{v}) d\hat{u} d\hat{v}$$

and

$$\chi(u,v) = \begin{cases} 1, \text{if } (u,v) \in W_s \\ 0, \text{otherwise} \end{cases},$$

$$\overline{u} = \frac{\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} u\chi(u,v) du dv}{\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \chi(u,v) dx dy},$$

$$\overline{v} = \frac{\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} v\chi(u,v) du dv}{\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \chi(u,v) du dv}.$$

Finally, Flusser et al. [33] derive a set of independent moments of the order $p + q \le 2$ as follows:

$$M_2 = \left\{ c_{01}, c_{00}c_{02}, c_{11}c_{02}, c_{10}c_{02}^2, c_{20}c_{02}^3 \right\}. \tag{26}$$

The remaining second-order moments can be obtained from M_2 by multiplication, involution with an integer exponent, and complex conjugation. To classify the patterns, we apply a straightforward test to compare vector M_2 with a template vector ($M_{2,\text{pattern}}$), using a classification threshold (ζ) [52, 53]. Note that a decision tree or neural network can be used for this classification [54, 55]. However, in our case, the above simple test works well. As an example, consider the detection of *star*, *curl*, *spiral*, and *shear* patterns, shown in Fig. 7. The tests have been constructed by adding Gaussian noise with a zero mean and a standard deviation, $\sigma = 0.2$, 0.3, and 0.4 [56]. The classification has been verified using 50 random samples for each configuration. The recognition rate (shown in Table 1) indicates excellent accuracy.

We are interested in the star patterns. A template star is evaluated numerically on the unit square. The second-order moments of the pattern are given by

$$M_{2,\text{star}} = \{0, 0.0948 + 0.0032i, 0.0177,\ 0, -0.0002\}. \tag{27}$$

The tested OFs are interpolated onto a unit square using the same grid size. We define a moving square window. For each position, we evaluated a sequence of moments with an increasing size of the window, creating a pyramid structure [46]. The largest window satisfying $\|M_{2,\ \text{star}} - M_2\| \le \zeta$, where $\zeta = 0.01$ is a threshold, is selected.

3.3 Scallop height and toolpath interval

The scallop height (h) indicates how close the machined surface is to the design surface between the consecutive tool tracks (Fig. 8).

The toolpath interval (l) is the maximum distance between two subsequent passes of a cutter such that the scallop height is under a prescribed limit (h). For a ball-end mill with the radius (r)

$$l_{\max} = \sqrt{\frac{8hrR'}{R' \pm r}}, \tag{28}$$

where R' is the radius of the surface curvature, $R' + r$ is for a convex surface, and $R' - r$ is for a concave surface. The formula has been derived with the assumption that $h << R'$ [57].

Equation (28) requires an evaluation of the curvature at a CC point on the STL surface across the feed direction. A straightforward approach is to fit a parametric surface to the local neighborhood of the CC point and use the parametric derivatives to evaluate the curvature. A quadric surface is the lowest-order interpolation suitable for computation of the second-order derivatives [58]. Under the implicit assumptions of low surface variation, high sampling frequency, and sampling regularity, the quadric approach can produce accurate results. A cubic bivariate polynomial surface patch with constraints is proposed in [59]. A recent technique employs the first- and the second-order Taylor series [60]. Second-order rational polynomial fitting is proposed in

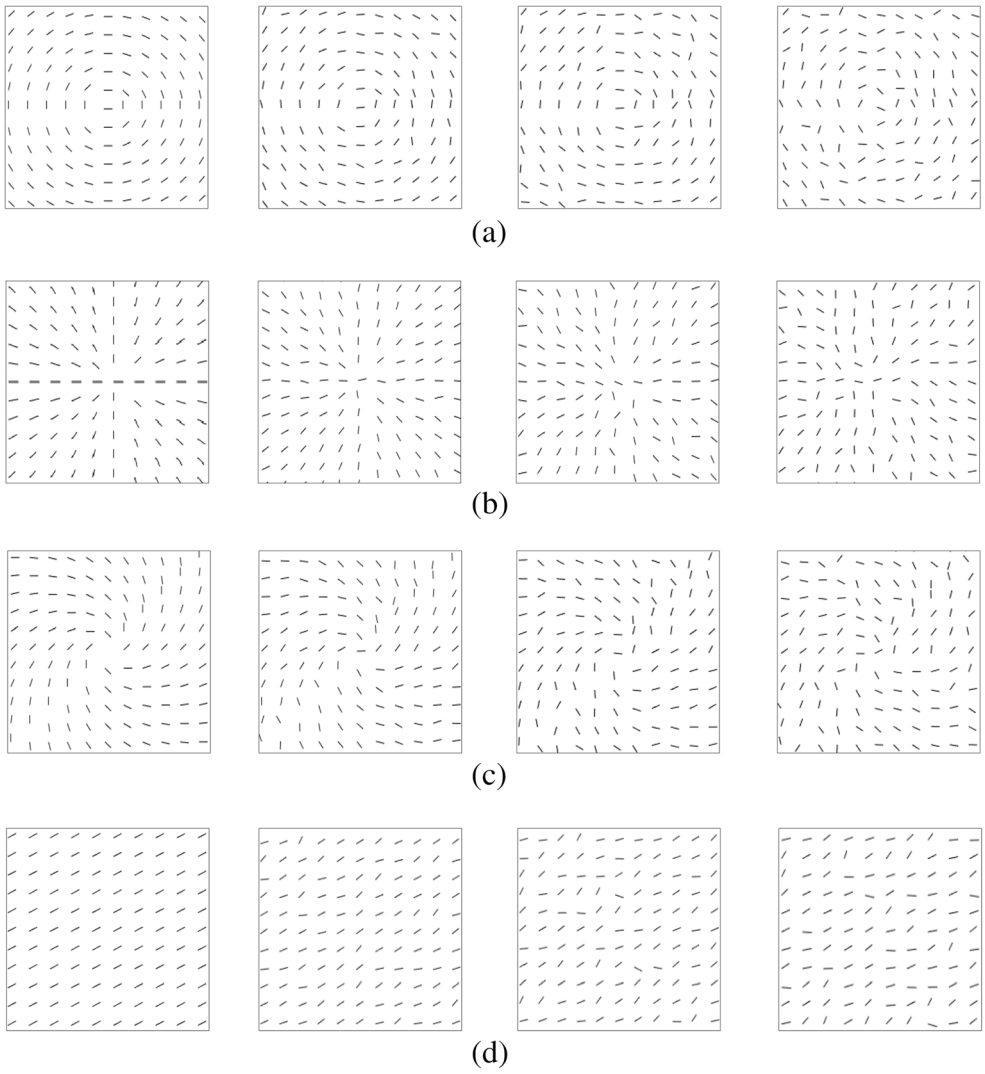


Fig. 7 Sample patterns subjected to Gaussian noise. From left to right: original pattern, noise level 0.2, 0.3, and 0.4. **a** Curl. **b** Star. **c** Spiral. **d** Shear

[19]. However, the above methods show a significant deterioration in the accuracy when irregular sampling (such as STL) is involved.

An early technique which avoids surface fitting was proposed in [61]. The approach is based on Euler's formula for the curvature in the direction θ.

Table 1 Recognition rate

Pattern	Standard deviation, σ		
	$\sigma = 0.2$	$\sigma = 0.3$	$\sigma = 0.4$
Star	100	98	86
Curl	100	94	88
Spiral	98	96	84
Shear	98	92	90
Average recognition rate	99	95	87

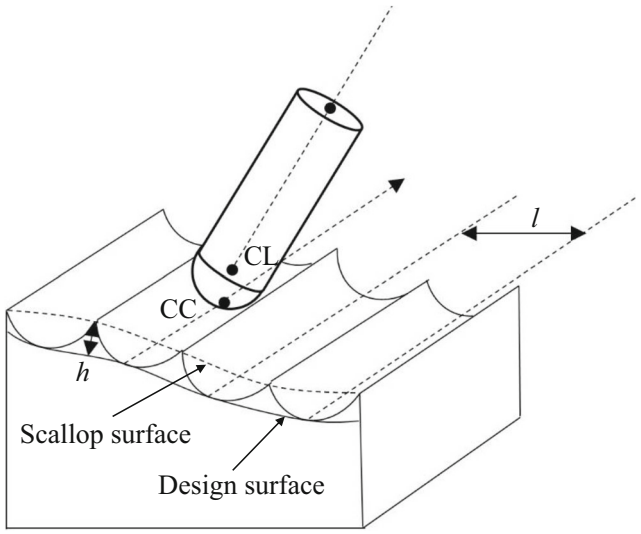


Fig. 8 Scallop height and the toolpath interval

$$k = k_{\min}\sin^2(\psi) + k_{\max}\cos^2(\psi), \tag{29}$$

where ψ is the angle between the principal direction ($T_{\min}$) and the tangent ($T\psi$). The circle passing through the points is computed, and the curvature of this normal section is estimated as the inverse of the radius of the circle. Combining this with the Euler formula yields the principal curvatures $k_{\min}$ and $k_{\max}$. A similar approach is introduced in [62]. The main problem with the above algorithms is that the circular arc may not be an accurate approximation of the normal section.

Tensor averaging methods [63, 64] evaluate the average of the curvature tensor over a small patch on the surface. The curvature of a polyhedron is zero within a face and infinite along the edges, but its average over the patch is finite. Unfortunately, tensor averaging methods may produce large errors for certain vertex arrangements.

In this paper, we use an efficient and robust method [65] (see also [32]) that is free of degenerate configurations. The surface properties are evaluated per face. Then, each value is estimated per vertex as a weighted average over the adjacent faces. The per-face computations are based on the finite difference derivatives. The second-order fundamental tensor of each facet (Π_f) is computed. The corresponding tensor (Π_{v_i}) at vertex v_i is a weighted sum of $\Pi_{f_{i,j}}$, where $f_{i,\ j}$ is the adjacent facet of vertex v_i. The weights are evaluated using the Voronoi area. The curvature is approximated by $\kappa(s) = s\,\Pi\,s^T$, where s is a unit direction vector [66]. When the CC point lies on an edge, a re-meshing is used so that the CC point becomes a vertex.

3.4 Transfinite interpolation for the radial vector field

Grid generation using TFI is characterized by fast execution, minimal input, automatic node connectivity, and good correlation between the boundary nodes and the interior mesh [27].

First, consider the case when the preferred OF is close to the radial pattern. Introduce a computational region $\Delta = \{0 \le \xi \le 1, 0 \le \eta \le 1\}$. The grid is a mapping from Δ onto a region D in the parametric coordinates (u, v). Let $R_k(\xi)$, $k = \overline{1, N_\eta}$ and $T_p(\eta)$, $p = \overline{1, N_\xi}$ be the boundary and inner coordinate curves in the parametric space (u, v) to be transformed into the computational space (ξ, η) (Fig. 9). Define the coordinate transformation by

$$\begin{aligned} F(\xi,\eta) &\equiv (u(\xi,\eta), v(\xi,\eta)) \\ &= \sum_p T_p(\eta)\psi_p(\xi) + \sum_k R_k(\xi)\varphi_k(\eta) - \sum_k \sum_p T_p(\eta_k)\psi_p(\xi)\varphi_k(\eta), \end{aligned} \tag{30}$$

where $\psi_p(\xi)$ and $\varphi_k(\eta)$ are the blending functions such that

$$\psi_p(\xi_i) = \begin{cases} 1, & p = i \\ 0, & \text{otherwise} \end{cases}, \ i = \overline{0, N_\xi}$$

$$\varphi_k(\eta_j) = \begin{cases} 1, & k = j \\ 0, & \text{otherwise} \end{cases}, \ j = \overline{0, N_\eta}.$$

Clearly,

$$\begin{aligned} F(\xi, \eta_k) &= R_k(\xi), k = \overline{0, N_\eta}, \\ F(\xi_p, \eta) &= T_p(\eta), p = \overline{0, N_\xi}. \end{aligned} \tag{31}$$

For simplicity, assume that the OF converges to a single point O_R (Fig. 9) and $N_\xi = N_\eta = 1$. The boundaries of the computational region and the blending functions are selected by $\xi_0 = \eta_0 = 0$, $\xi_1 = \eta_1 = 1$, $\psi_0(\xi) = 1 - \xi$, $\psi_1(\xi) = \xi$, $\varphi_0(\eta) = 1 - \eta$, and $\varphi_1(\eta) = \eta$. TFI generates the so-called O-grid with the boundary curves defined by $T_0(\eta)$, $T_1(\eta)$, and $R_0(\xi) = R_1(\xi)$ (Fig. 9). Note that $O_R = T_0(\eta)$ is a curve degenerated into a single point while $R_0(\xi) = R_1(\xi)$ is a linear cut connecting O_R and $T_1(\eta)$.

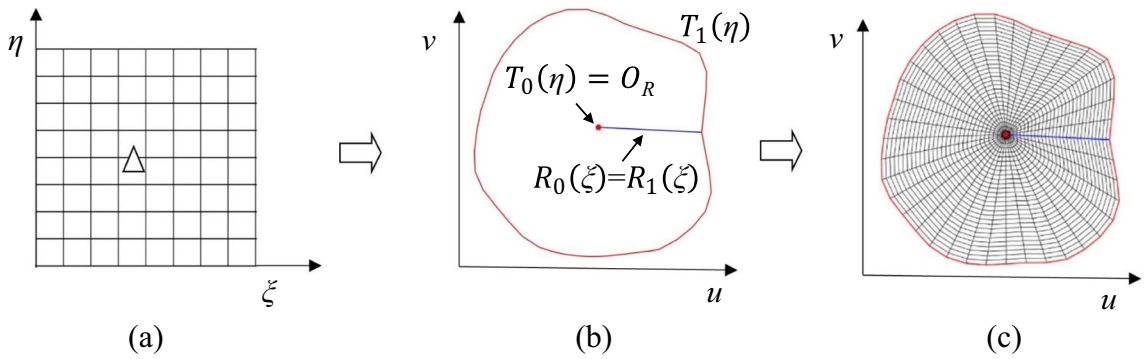


Fig. 9 O-grid generated by TFI. **a** Computational domain. **b** Boundary curves. **c** TFI grid

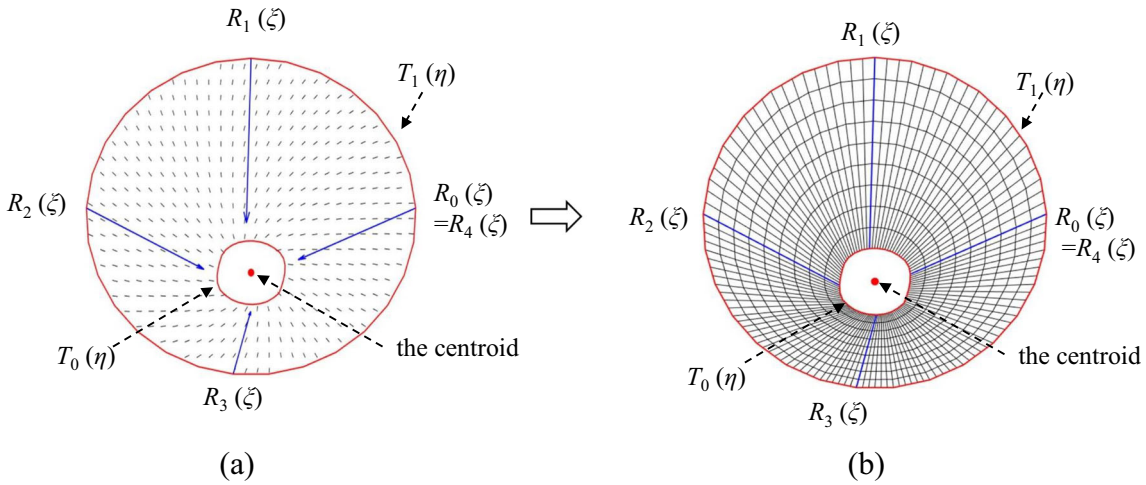


Fig. 10 Irregular OF and a corresponding TFI grid. **a** Important vectors. **b** TFI grid

It is often the case that the EMRR vectors form an irregular star; i.e., the OF does not converge to one point but to a relatively small region in the parametric domain. In this case, we select *important vectors*. For instance, the important vectors can be defined at CC points with a high curvature or large rotation angles. An important vector can also be a representative of a cluster characterized by similar radial directions. In this case, we interpolate the boundary of the parametric region and include the important vectors or the flow lines, representing constant generalized coordinate lines [27]. Figure 10a and b exemplify the case characterized by four generalized coordinate lines, $R_0(\xi)$, $R_1(\xi)$, $R_2(\xi)$, and $R_3(\xi)$; a cut, $R_4(\xi) = R_0(\xi)$; and two circular boundaries, $T_0(\eta)$ and $T_1(\eta)$ (curvilinear disks). The blending functions are:

$$N_\xi = 4, N_\eta = 1, \xi_i = i\Delta\xi, \Delta\xi = \frac{1}{4}, i = \overline{0, N_\xi}, \eta_j = j, j = \overline{0, N_\eta},$$
$$\varphi_0(\eta) = 1-\eta, \varphi_1(\eta) = \eta,$$
$$\psi_i(\xi) = \begin{cases} 0, & \xi < \xi_i, \\ (\xi-\xi_{i-1})/\Delta\xi, & \xi_{i-1} \le \xi < \xi_i, \\ 1-(\xi-\xi_i)/\Delta\xi, & \xi_i \le \xi < \xi_{i+1}, \\ 0, & \xi > \xi_{i+1}. \end{cases} \tag{32}$$

The important EMRR vectors represent clusters proposed in [67] for simplifying and smoothing the VFPD (Fig. 10). The clustering is based on a hierarchical tree and the angular distance between the vectors. The important vector is a weighted average of the members of the cluster.

Finally, TFI is adapted to the acceptable toolpath interval by bisection. The Hausdorff distance between the neighboring tracks, $P_i(\xi) = S(F(\xi, \eta_i))$ and $P_{i+1}(\xi) = S(F(\xi, \eta_{i+1}))$, is evaluated. If $\text{dist}_H(P_i, P_{i+1}) < l_i$, where l_i is defined by Eq. (28), a new grid line is inserted.

4 CRZ

As noted, the most important drawback of the radial toolpath is the redundancy. In this section, we propose a simple but efficient algorithm to reduce the redundancy by decomposing the radial path into layers and using TFI.

4.1 Redundancy

Redundancy is characterized by an area covered by the tool several times when the actual interval (l) between two adjacent paths is smaller than the maximum allowed interval ($l_{\max}$). Consider two adjacent paths, $P_i(s)$ and $P_{i+1}(s)$ (Fig. 11), and let $l_{\max,i}(s)$ be the maximum allowed toolpath interval along $P_i(s)$ and $l_i(s)$ the actual distance. Omitting the subscript i, the redundancy is defined by [68]

$$w(s) = \frac{l_{\max}(s) - l(s)}{l_{\max}(s)}. \tag{33}$$

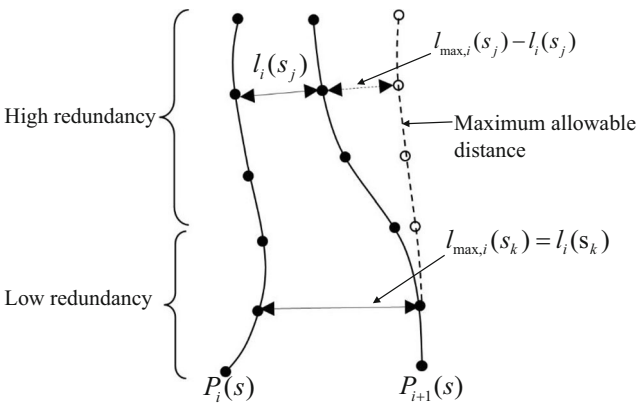


Fig. 11 Redundancy of a toolpath

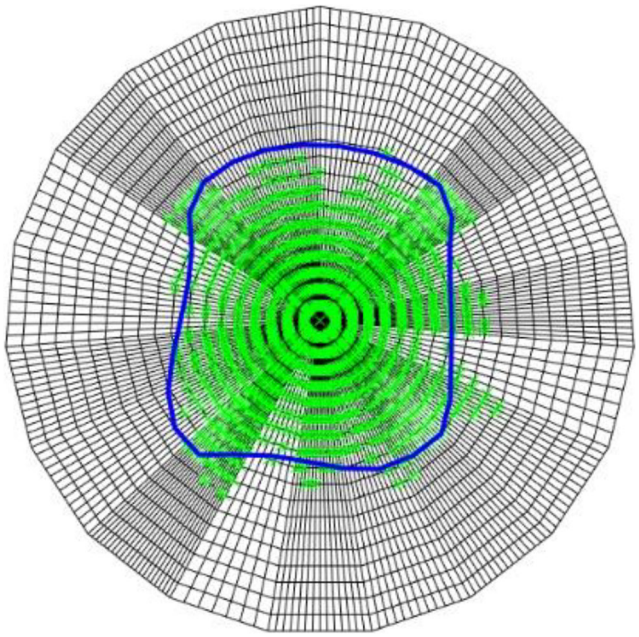


Fig. 12 Cluster of redundant points, saddle surface (Fig. 6c)

We call a CC point (i, j) redundant if $w_{i,j} > w_0$, where w_0 is a prescribed threshold.

For a radial toolpath, the redundant points form a cluster around the pole (Fig. 12). The isolated redundant points can be removed using the outlier removal technique [69]. The boundary of the cluster is approximated by a convex hull and smoothed by cubic spline interpolation (Fig. 12).

4.2 Minimization of the machining time

The CRZ minimizes the machining time using a partition of the region dividing a star-like toolpath into several subregions. The subregions are characterized by the increasing the stepover between the radial tracks.

Consider the CRZ in Fig. 13. The original radial grid is displayed in Fig. 13a. The machining is performed along a radial zigzag in each layer. The machining time $(t = t(R_1, R_2, ..., R_{N_P}))$ is a function of the partition of the radius (R) into $R_1, R_2, ..., R_{N_P}$. Consider a single partition point R_1 and $t_1 = t_1(R_1)$. Clearly, if $R_1 = R_{1,1}$ is small, the machining time increases due to additional zigzag turns ($ABCD$ in Fig. 13b). However, if $R_1 = R_{1,2}$ is large, the machining time is close to $t_0 = t_0(R)$ (Fig. 13c). Numerical experiments show

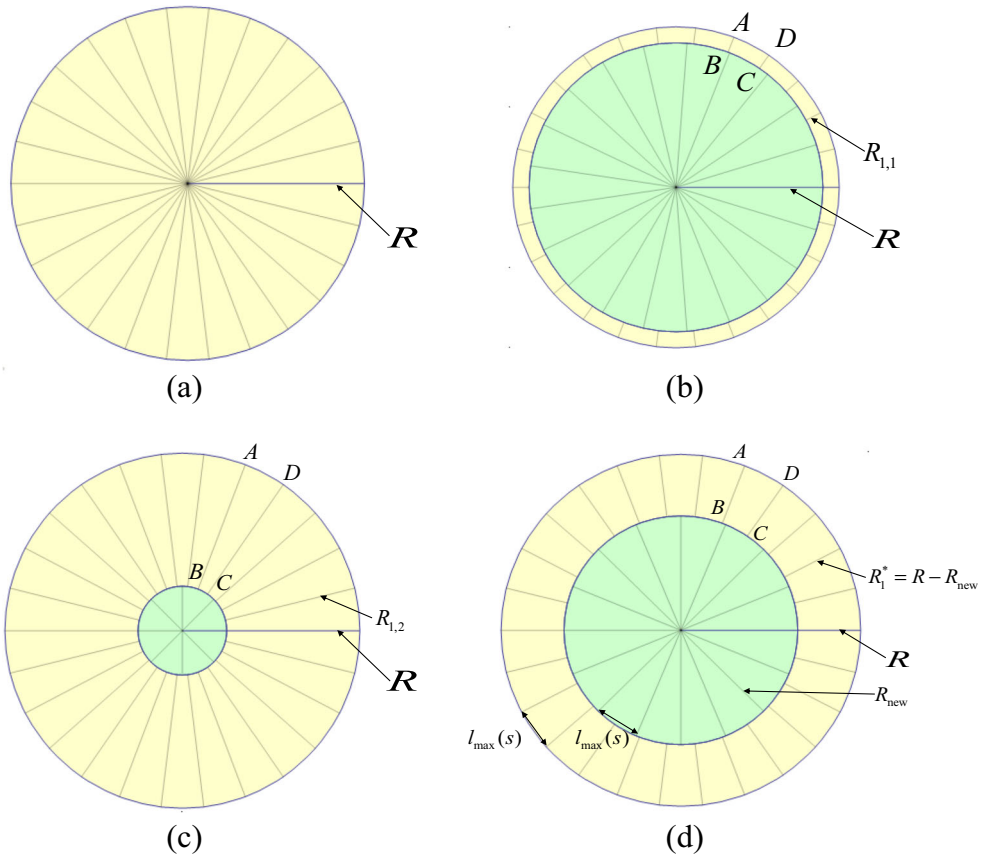


Fig. 13 Minimization of the machining time. **a** Radial pattern. **b** Small (insufficient) split. **c** Excessive split. **d** Optimal split

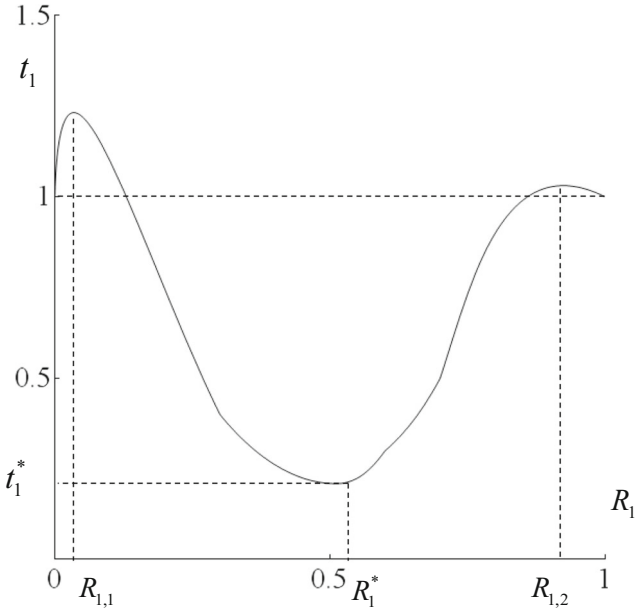


Fig. 14 Typical graph of the machining time with one partition point

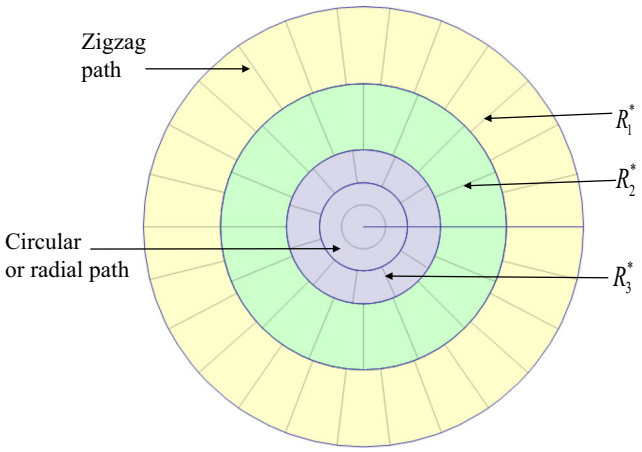


Fig. 15 Final toolpath with three partition points and a circular path around the pole

that for surfaces with a sharp peak or a saddle, the function $t_1 = t_1(R_1)$ often has a global minimum on $[0, R]$ (Fig. 13d).

The function $t_1 = t_1(R_1)$ is an excellent test of the applicability of the method. If $t_1(R_1) \geq t_0(R)$ for $\forall R_1$, the method is not applicable. A typical graph of the machining time $t_1 = t_1(R_1)$ is shown in Fig. 14 (R_1 and t_1 have been normalized). Let $R_1^* = \text{argmin}_{R_1}(t_1(R_1))$, and let $t_1^* = t_1(R_1^*)$ be the corresponding machining time using the SRZ on the disk (R_1^*, R). Consider a reduced region $R_{\text{new}} = R - R_1^*$. Repeat the procedure for $t_2 = t_2(R_2)$, $R_2 \in [0, R_{\text{new}}]$, obtaining R_2^* and t_2^*. The partition makes it possible to set up the maximum machining strip $l_{\max}$ at every circular boundary, reducing the redundancy and therefore reducing the machining time. The process is repeated until the total machining time cannot be improved. The small remaining area around the pole can be machined by a circular or a radial path (Fig. 15).

Note that for simplicity, Figs. 13, 14, and 15 present a toolpath for an ideal circular configuration. However, the CRZ works on irregular configurations, such as in Fig. 16a and b. The calculation of the total machining time for each iteration of the CRZ is computationally expensive. A simplified version of the algorithm is based on an evaluation of the redundancy. If $w_i(s)$ is sufficiently large, the next partition point R_i is inserted. Although this approach is heuristic and requires a threshold, our numerical experiments show that $w(s) \geq w_0 = 0.5$ produces good results.

5 Toolpath generation

The CRZ toolpath generation algorithm can be summarized as follows:

Input includes the STL surface (S), tool radius (r), maximum allowed scallop height (h), parameters of the kinematic

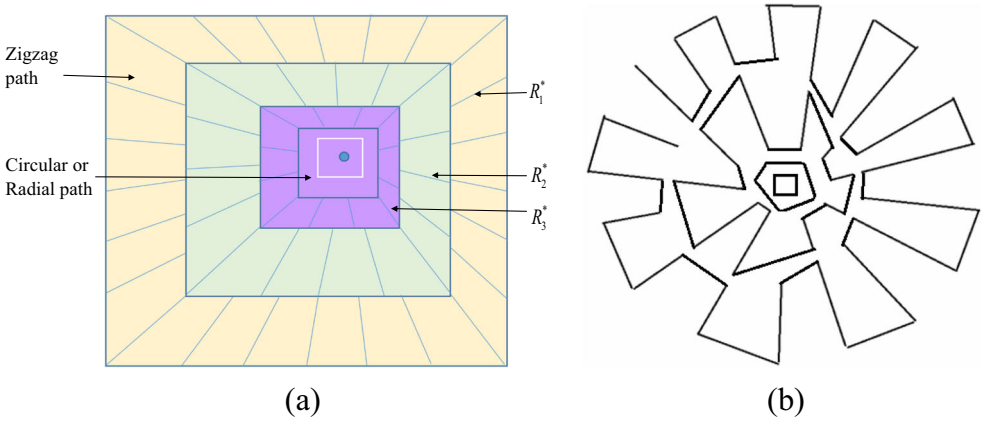


Fig. 16 CRZ path on a rectangular configuration. **a** CRZ layers. **b** CRZ toolpath

transformation (1) (Section 1), and threshold for the redundancy coefficient (optional, if the threshold method is used).

1. Generate the EMRR-OF on S.
2. Flatten S onto the parametric domain (D) by conformal mapping (18) (Section 3.1) and transfer the OF into D, accordingly.
3. Construct the OF on a uniform grid in D by barycentric interpolation.
4. Calculate the moment invariants on the OF; detect the radial patterns and their poles.
5. For each radial region, evaluate the important vectors.
6. For each radial region, generate the CRZ using either direct calculation of the machining time or the redundancy threshold (w_0) (less accurate but faster).
7. Generate the toolpath for the remaining parts of the parametric domain.
8. Generate CC points along the toolpath so that the kinematic error ($\varepsilon_{\max}$) does not exceed the prescribed tolerance h.
9. Map the CC points by inverse transformation onto S.

Output includes a CNC program (G code) that is ready for milling operations.

6 Numerical and cutting experiments

We test the CRZ against conventional methods, i.e. ZZ, CP, SRZ, as well as the ISOP [12] - the method that find optimal feeding angle of the iso-plane that maximizes the total EMRR. We also compare the CRZ with the HS and the FP paths obtained by NX11.

Our virtual and the actual milling machine is Haas VF-2TR. The maximum speed of the linear axes v_x, v_y, and v_z is 275 mm/s, and that of the rotary axes v_a and v_b is 33.33°/s. The kinematic transformation in Eqs. (1) and (2) is applied to post-process the G code and to evaluate the kinematic error. The experiments use a ball-end mill with a radius (r) equal to 2 mm and the maximum allowable scallop height (h) between 0.1 and 1.0 mm. The toolpath length (L), machining time (T), number of partitions (N_P), and the number of CC points (N_{CC}) are used to measure the performance of CRZ relative to the preceding methods. The maximum allowable kinematic error ($\varepsilon_{\max}$) is equal to h.

Real cutting was performed, and the quality of the surfaces was compared by their roughness following the conventional standard ISO 4287-1997 [70], using a Taylor Hobson Talysurf 120 stylus machine (Fig. 17) operated by National Institute of Metrology (Thailand) (NIMT). Additionally, the accuracy of the surfaces was compared by the kinematic error per forward step (see the Appendix).

6.1 Example 1: cone

A cone and the corresponding OF are given in Fig. 6a and b, respectively. Figure 18 shows sample toolpaths obtained by the preceding methods (Fig. 18b–e) and the proposed algorithm (Fig. 18f–h). In Fig. 18g, the CRZ layers are visualized by using a large allowable scallop height. It is our experience that undesirable tool marks and scallops at the boundary between the layers can be eliminated if the layers slightly overlay (Fig. 18h). Table 2 shows that the SRZ has the longest path due to the large redundant area around the peak. L is the length of the toolpath, N_{CC} is the number of CC points, T is the machining time with the maximum feed rate, and A is the advantage of the proposed CRZ toolpath.

The CRZ toolpath is shorter than the SRZ; however, it is still the second largest path relative to the competing methods. Nevertheless, the CRZ is the fastest. The advantage of the CRZ is calculated by

$$A = \frac{T_{\text{conventional}} - T_{\text{CRZ}}}{T_{\text{conventional}}} 100\%. \tag{34}$$

The machining time decrease is about 77%, relative to the HS and FP paths of NX11. The reduction of the machining time is due to a reduction of the angular variations when the tool follows the VFPD (Tables 2, 3, and 4).

Figure 19 shows the corresponding CRZ partitions of the circular region into layers, providing the optimal

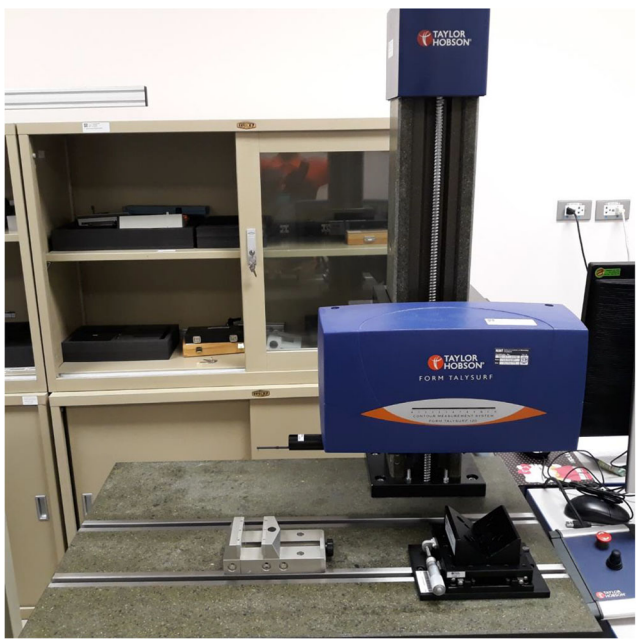


Fig. 17 Taylor Hobson Talysurf (120) stylus machine, NIMT

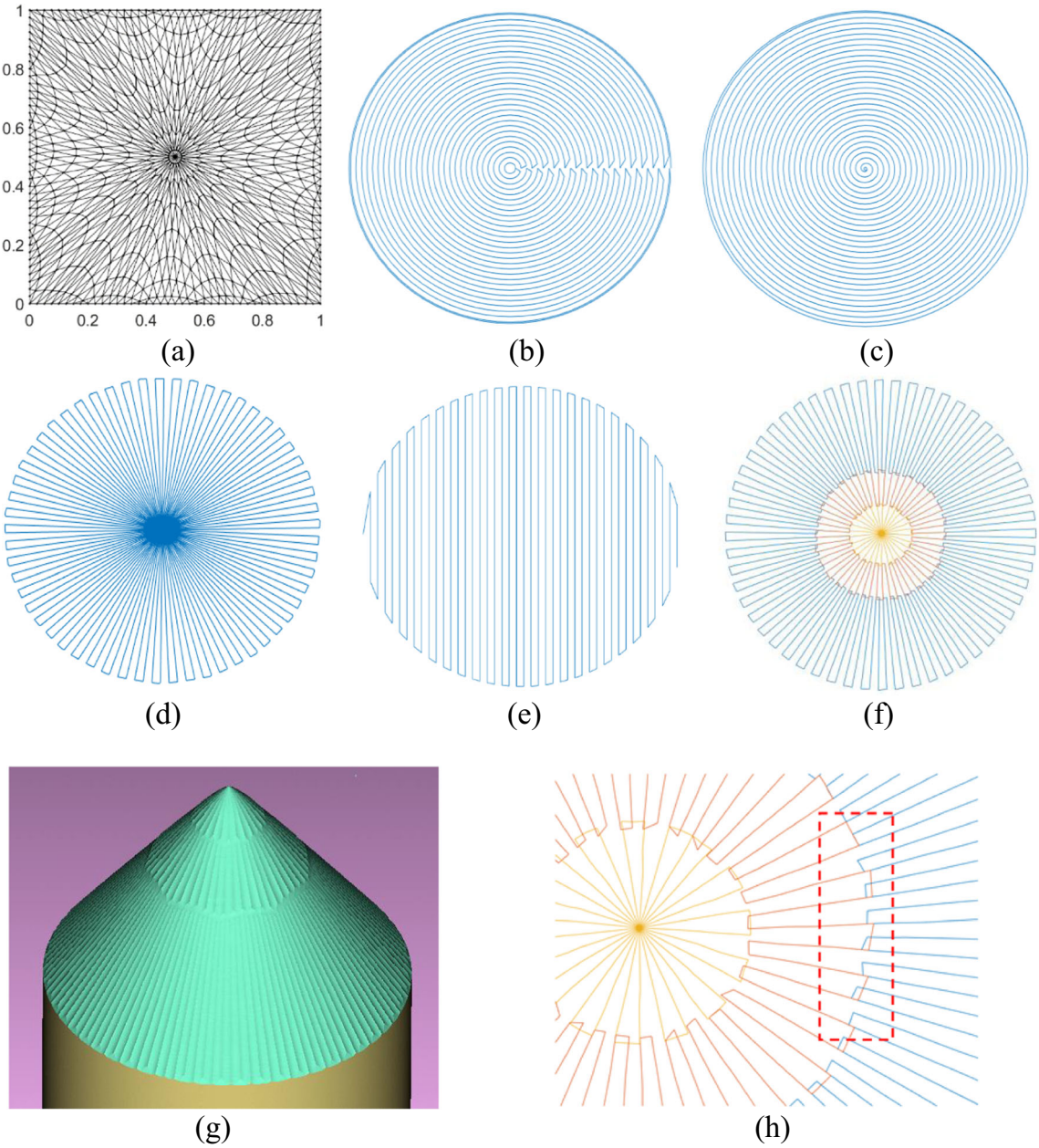


Fig. 18 Toolpath on a cone. CRZ versus the preceding methods. **a** Flattened cone. **b** FP/NX11. **c** HS/NX11. **d** SRZ path. **e** ISOP path. **f** CRZ path. **g** CRZ layers, virtual cutting. **h** Overlaid CRZ layers

machining time for h = 0.1. The first partition ($R_1^* = 0.92$) reduces the machining time significantly from 214 to 78s. The second partition decreases the time by 2.5s. The third partition does not reduce the time and, therefore, is discarded. The remaining part of the surface is cut by a conventional radial path. A two-point partition ($R_1^* = 0.92$ and $R_2^* = 0.039$) reduces the machining time by 2.8 times (the radius has been normalized to [0,1]). Similar partitions for h = 0.5 and h = 1.0 presented in Table 2 show that the CRZ toolpath outperforms the preceding methods in terms of the machining time for every h. Although the number of required CC points for the CRZ is usually larger than for the CP, this does not affect the machining time. The best result is a 77.6% time reduction with regard to the FP for h = 0.1. The worst is a 20% advantage over the zigzag toolpath for h = 1.0.

6.2 Example 2: saddle surface

The test surface is given by $z = xy\frac{x^2-y^2}{x^2+y^2}, -30 \le x, y \le 30$ (Fig. 6c). The surface has been converted to the STL format by Matlab libraries [71]. A five-axis toolpath

Table 2 Testing the proposed method on a conical surface (example 1)

Example 1: conical surface	h = 0.1 mm				h = 0.5 mm				h = 1.0 mm			
	L (mm)	N_{CC}	T (s)	A (%)	L (mm)	N_{CC}	T (s)	A (%)	L (mm)	N_{CC}	T (s)	A (%)
NX11												
HS	2505.5	1924	317.6	76.2	1243.0	952	172.2	77.0	1224	926	172.0	77.7
FP	2457.5	1921	336.9	77.6	1284.5	931	162.0	75.6	1128.7	804	140.4	72.7
Iso-parametric												
ZZ	2368.1	1120	173.6	56.5	1005.9	435	73.0	45.8	669.1	402	48.3	20.9
CP	2073.4	795	259.6	70.9	1107.1	406	122.5	67.7	752.4	406	90.2	57.6
SRZ	4151.9	3787	226.29	66.6	1696.4	1040	95.7	58.7	1386.0	1040	80.2	52.3
ISOP	2798.7	2286	179.9	58.0	968.8	1045	69.1	42.8	760	801	53.7	28.8
CRZ	*3217.3*	*2274*	*75.4*	N_P = 2	*1517.7*	*1132*	*39.5*	N_P = 1	*1238.7*	*790*	*38.2*	N_P = 1

L is the length of the toolpath, N_{CC} is the number of *CC* points, T is the machining time with the maximum feed rate, N_P is the number of partitions, and A is the advantage of the proposed *CRZ* toolpath. The italic font indicates the best value.

Table 3 Testing the proposed method on a saddle surface (example 2)

Example 2: saddle surface	h = 0.1 mm				h = 0.5 mm				h = 1.0 mm			
	L (mm)	N_{CC}	T (s)	A (%)	L (mm)	N_{CC}	T (s)	A (%)	L (mm)	N_{CC}	T (s)	A (%)
NX11												
HS	6099.4	10128	1002.5	58.7	3949.4	5476	640.0	48.0	3838.0	4969	590.5	50.0
FP	7939.8	8197	1131.2	63.4	3446.0	4671	575.0	42.1	2915.4	3965	482.7	38.8
Iso-parametric												
ZZ	5347.9	4774	770.2	46.2	2889.8	916	388.5	14.3	2185.8	495	281.7	− 4.7
CP	6980.2	6123	964.6	57.1	3236.6	1634	602.9	44.8	2810.5	1082	572.1	48.41
SRZ	10383	10441	1367.0	66.4	4488.8	2687	876.5	62.0	3102.8	1678	598.4	50.68
ISOP	4350.6	3794	1023.4	59.5	3351.2	1127	409.7	18.7	2164	775	256.7	− 14.9
CRZ	*5481.3*	*5066*	*458.02*	N_P = 3	*2587.7*	*1389*	*332.8*	N_P = 2	*1994.6*	*988*	*295.1*	N_P = 2

The notations are given in Table 2 and in the text

Table 4 Testing the proposed method on a crown surface (example 3)

Example 3: crown surface	h = 0.1 mm				h = 0.5 mm				h = 1.0 mm			
	L (mm)	N_{CC}	T (s)	A (%)	L (mm)	N_{CC}	T (s)	A (%)	L (mm)	N_{CC}	T (s)	A (%)
NX11												
HS	4683.1	3576	501.4	74.7	2279	1908	270	56.2	2209	1758	259	58.6
FP	4658.7	4149	541.1	76.6	2253.0	1955	260.3	54.6	2196	1952	260	58.8
Iso-parametric												
ZZ	5213.9	2421	316.1	59.9	1959	686	129	8.5	1458	505	92	− 16.3
CP	3971.2	1933	428.9	70.4	2277	541	240	50.8	2188	452	219	51.1
SRZ	6373.2	5708	152.3	16.9	2819	1568	127	7.0	1874	908	107	0
ISOP	3685.5	2470	213.6	40.7	2700	1062	165	28.4	1907	805	118	9.3
CRZ	*5517.3*	*2275*	*126.5*	N_P = 1	*2683*	*1010*	*118*	N_P = 1	*1874*	*908*	*107*	N_P = 0

The notations are given in Table 2 and in the text

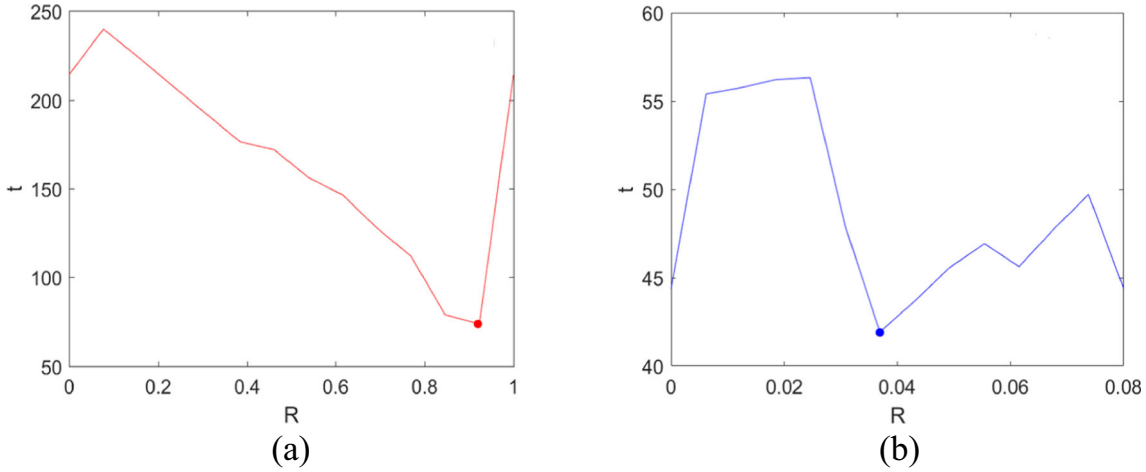


Fig. 19 CRZ partitions. **a** 1st partition, $R_1^* = 0.92$. **b** 2nd partition, $R_2^* = 0.039$

cannot be generated by NX11 from the STL format. Therefore, we convert the STL to a solid model, using the fit surface functions of NX11. The average Euclidean error is 0.05 mm, which ensures the accuracy of the approximated surface. The HS and FP toolpaths are generated from the solid model. The STL surface and its OF are shown in Fig. 6c and d, respectively. The flattened STL is shown in Fig. 20a. Figure 20b–e show the CRZ partition graphs of the saddle surface. The first partition (usually the most significant) reduces the machining time by 60% (1367 s to 552 s), whereas the second partition reduces the machining time by 20% (149 s to 119 s). The third partition only slightly improves the machining time (61.5 s to 59.7 s). The iterations stop at the fourth partition which actually increases the time (Fig. 20e). Therefore, the optimal CRZ decomposition is given by $R_1^* = 0.77$, $R_2^* = 0.08$, and $R_3^* = 0.13$. Since the third partition does not substantially improve the machining time, the first two partitions are selected. The CRZ toolpath is shown in Fig. 20j. The toolpaths produced by the conventional methods are shown in Fig. 20f–i. The toolpath generated by the optimal ISOP method is shown in Fig. 20i. The results are summarized in Table 3. The best result for h = 0.1 is a 66% reduction with regard to the SRZ path, whereas the smallest reduction is 46% with regard to the ZZ path. Note that for h = 1.0, the ZZ and ISOP methods outperform CRZ. In this case, the additional turns combined with the redundancy of the CRZ slow down the cut, whereas the ZZ and ISOP required a small number of turns and were characterized by a small redundancy index. Additionally, ISOP follows the average direction of the VFPD which reduces the redundancy. Hence, for h = 1.0, the ISOP method is approximately 15% faster than the proposed CRZ. However, note that the CRZ is still 50% faster than the conventional SRZ even for h = 1.0.

Conversely, for h = 0.1 and 0.5, ISOP and ZZ paths still are the shortest. However, the CRZ is the best strategy for reducing the machining time. The sample virtual cuttings by the ISOP path and the proposed CRZ are shown in Fig. 21a and b, respectively.

6.3 Example 3: dental crown, semi-finish cut

Consider a model of a dental crown for a canine tooth 1:10 (46 mm × 46 mm × 46 mm), represented by the STL format. The solution is of particular interest since it offers a new approach to the practical problem of producing crowns (implants) by five-axis machining. The STL model, flattened STL, and OF are shown in Fig. 22a–c, respectively. Figure 22d and e show the partition graphs. The first partition reduces the machining time by 17% (from 152 to 126 s) while the second partition fails (Fig. 22e). Figure 22j presents the resulting CRZ with a one-point partition. The CP, the ZZ, the SRZ, and the ISOP paths are shown in Fig. 22f–i, respectively. From Table 4, the ISOP is the shortest with h=0.1; however, the CRZ outperforms it in terms of the machining time by 40%. For small h, CRZ outperforms other methods as well. Note that the advantage of CRZ increases as the maximum allowable scallop height decreases.

Figure 23 shows virtual and real SRZ and CRZ cuts performed with F480 (slow, safe machining). The machining time of the SRZ is 52 min, whereas the machining time for CRZ is 40 min.

The roughness of SRZ and CRZ was compared by the stylus Taylor Hobson Talysurf (120) at NIMT (Fig. 17). The profiles (amplitudes) were measured across the tool track directions. Twenty samples were randomly scanned on the SRZ surface. The CRZ surface was sampled ten times randomly in the area of the first and the second layers. The average roughness for the SRZ of about 19.1 μm is better than that of the CRZ (25.0 μm). The maximum roughness is approximately

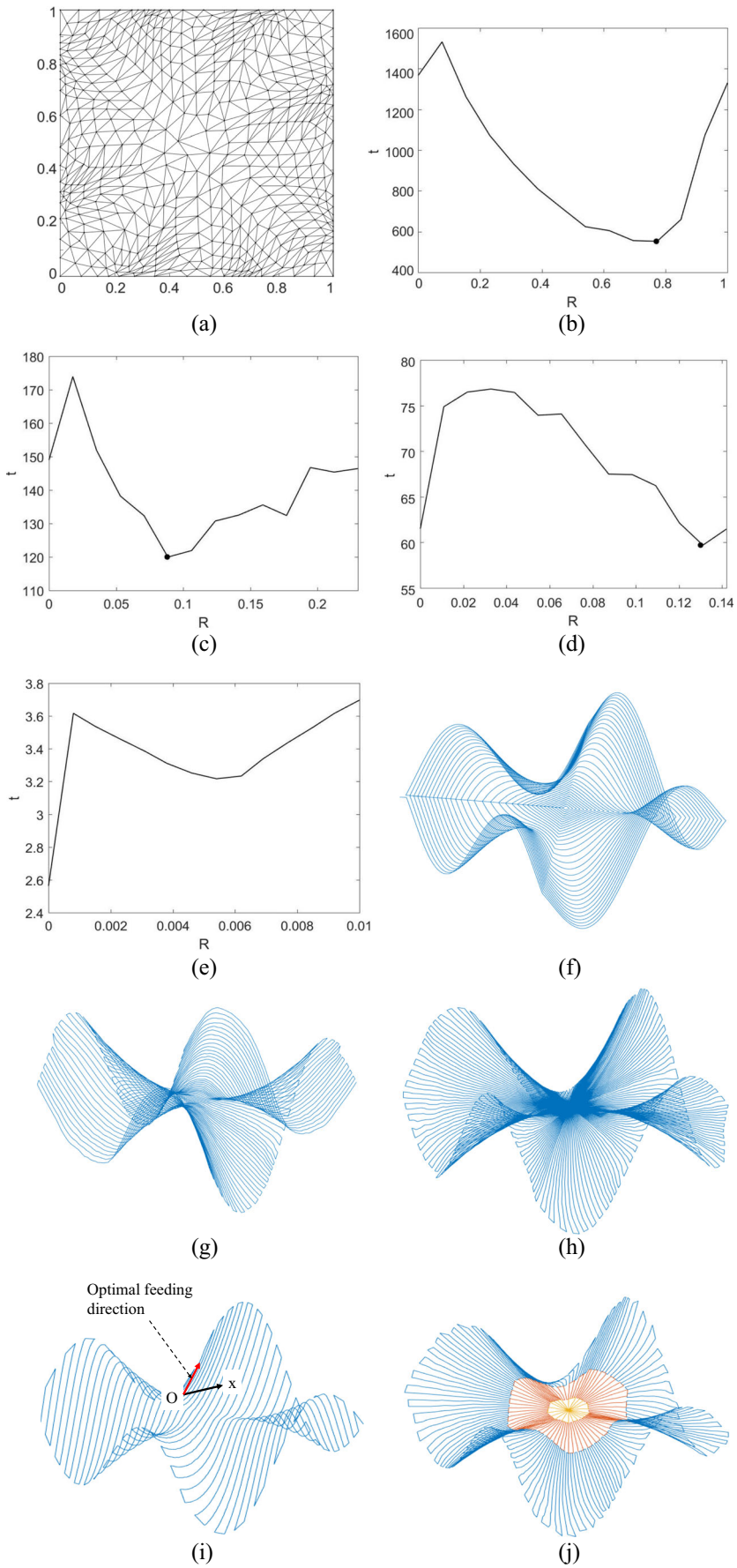


Fig. 20 CRZ versus conventional toolpath strategies. **a** Flattened STL surface of Fig. 6c. **b** 1st partition, $R_1^* = 0.77$. **c** 2nd partition, $R_2^* = 0.08$. **d** 3rd partition, $R_3^* = 0.13$. **e** 4nd partition, no improvement. **f** FP/NX11 path. **g** ZZ path. **h** SRZ path. **i** ISOP path. **j** CRZ path

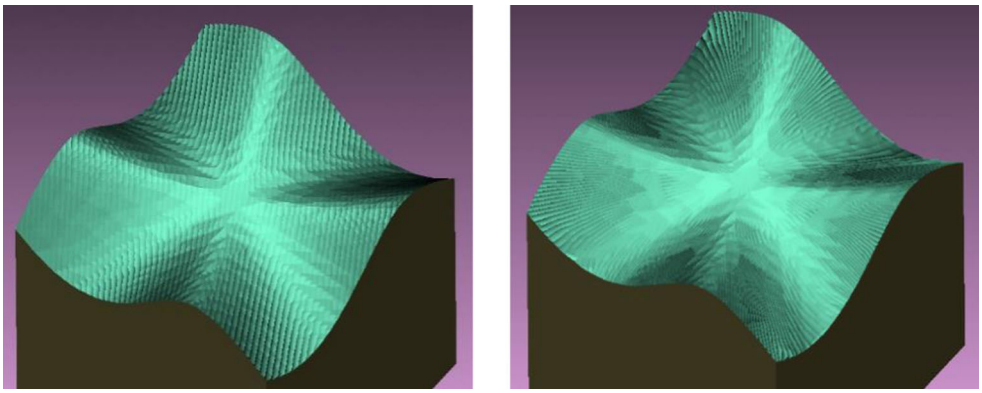


Fig. 21 Virtual cutting of a saddle surface. **a** ISOP path. **b** CRZ path

the same, i.e., 85.0 μm and 91.1 μm, respectively. However, the above difference is acceptable, given that the CRZ is faster by 22%. Note that this machining time differs from the theoretical estimate in Table 4, where the feed rate was assumed to be maximal. Finally, the accuracy of the machined surfaces (examples 3 and 4) was evaluated by means of the kinematic error induced by non-linear trajectories of the machine (see the Appendix).

6.4 Example 4: twisted surface, transfinite interpolation

Consider an STL surface, a flattened mesh, and the OF EMRR in Fig. 24a–c, respectively. The CRZ and the competing toolpaths are generated for $h = 0.1$, 0.5, and 1.0. The comparison results are given in Table 5. The CRZ outperforms the conventional toolpath for every h.

To verify the roughness of the part surface, the real cut was performed using the CRZ and the CP for $h = 0.01$. Figure 24d shows the first partition graph of the CRZ toolpath for $h = 0.01$. CP and CRZ toolpaths are given in Fig. 24e and f, respectively. The initial rough cut is shown in Fig. 25a. The semi-finish CP and the CRZ are shown in Fig. 25b and c, respectively. The machining time was 40 min and 28 min, respectively, with a maximum feed rate of 275 mm/s.

The roughness of the surfaces was measured using 20 random samples. The average roughness of the CRZ surface is 7.10 μm, and the CP surface is 6.29 μm. The maximum roughness is 36.8 μm and 32.2 μm, respectively. An example of the roughness profile for the CRZ surface is shown in Fig. 26.

It should be noted that examples 3 and 4 represent a semi-finish cut. There are still some tool marks on the surfaces due to discontinuities of the feed rate at the CC points and rear gouging. In order to remove them, the entire optimization scheme [12] has to be employed including 15 kinematic constraints, rear gouging avoidance, and possibly, a smaller tool. However, we consider the experiment a success since the average and the maximum roughness of the surfaces obtained for the reference methods is in the same range, whereas the *machining time of the CRZ is substantially shorter*. As noted above, the analysis of the kinematic error for examples 3 and 4 is given in the Appendix.

Consider the OF of the STL surface in Fig. 24c. This OF does not converge to one point or a small region around a pole. Consequently, the above CRZ can be improved. To better align the toolpath with the VFPD and enhance the efficiency of the toolpath, we combine the CRZ with TFI, as presented in Section 3. The OF has been arranged in clusters, characterized by similar directions using a clustering algorithm [67]. The weighted averages of the clusters are called important vectors. Since the important vectors do not meet at a pole, the *convergence region* is detected as shown in Fig. 27a. It is approximated by a convex hull of the intersection points and then circumscribed by the closest circle or an ellipse using a least-square approximation. Recall that a toolpath generated by the CRZ on a TFI grid is denoted by CRZT. The CRZ and CRZT one-point partitions $R_1^* = 0.84$ and $R_1^* = 0.72$, respectively, are presented in Fig. 27c. Note that the convergence region must be included in the redundancy region. Otherwise, the vectors may not form a star-like configuration. The CRZT for the twisted surface has been generated with a pole selected as the centroid of the redundancy region. The tool trajectories of the CRZT are shown in Fig. 27b. Clearly, the CRZT tracks are better aligned with the OF.

Figure 28a and b display the CRZ and CRZT paths, respectively. Since the CRZT more closely follows the EMRR direction, it provides an extra improvement of about 8.3% relative to the original CRZ (see Table 5). Figure 29a and b show a virtual cutting by the CRZT. The resulting surfaces are almost identical; however, the CRZT is smoother and faster than the CRZ. Moreover, for $h = 1$ mm, the CRZT machining time is 22% better than the CRZ. This is a promising result, which makes it possible to conjecture that CRZT can solve the problem of the slow machining time of the CRZ for a large allowable scallop height (rough and semi-rough cut).

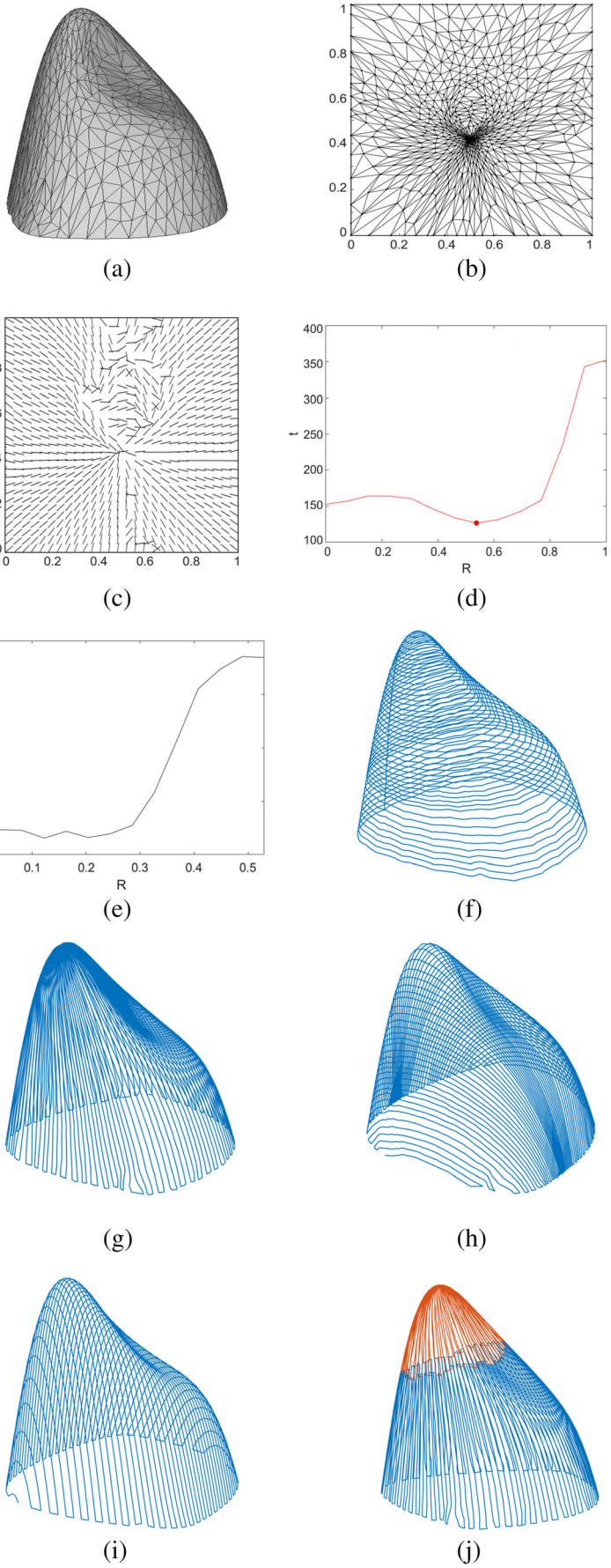


Fig. 22 Toolpath results for the canine tooth model. **a** Dental crown (canine tooth). **b** Flattened surface (**a**). **c** EMRR orientation field. **d** 1st partition, $R_1^* = 0.54$. **e** 2nd partition (no improvement). **f** CP path. **g** ZZ path. **h** SRZ path. **i** ISOP path. **j** CRZ path

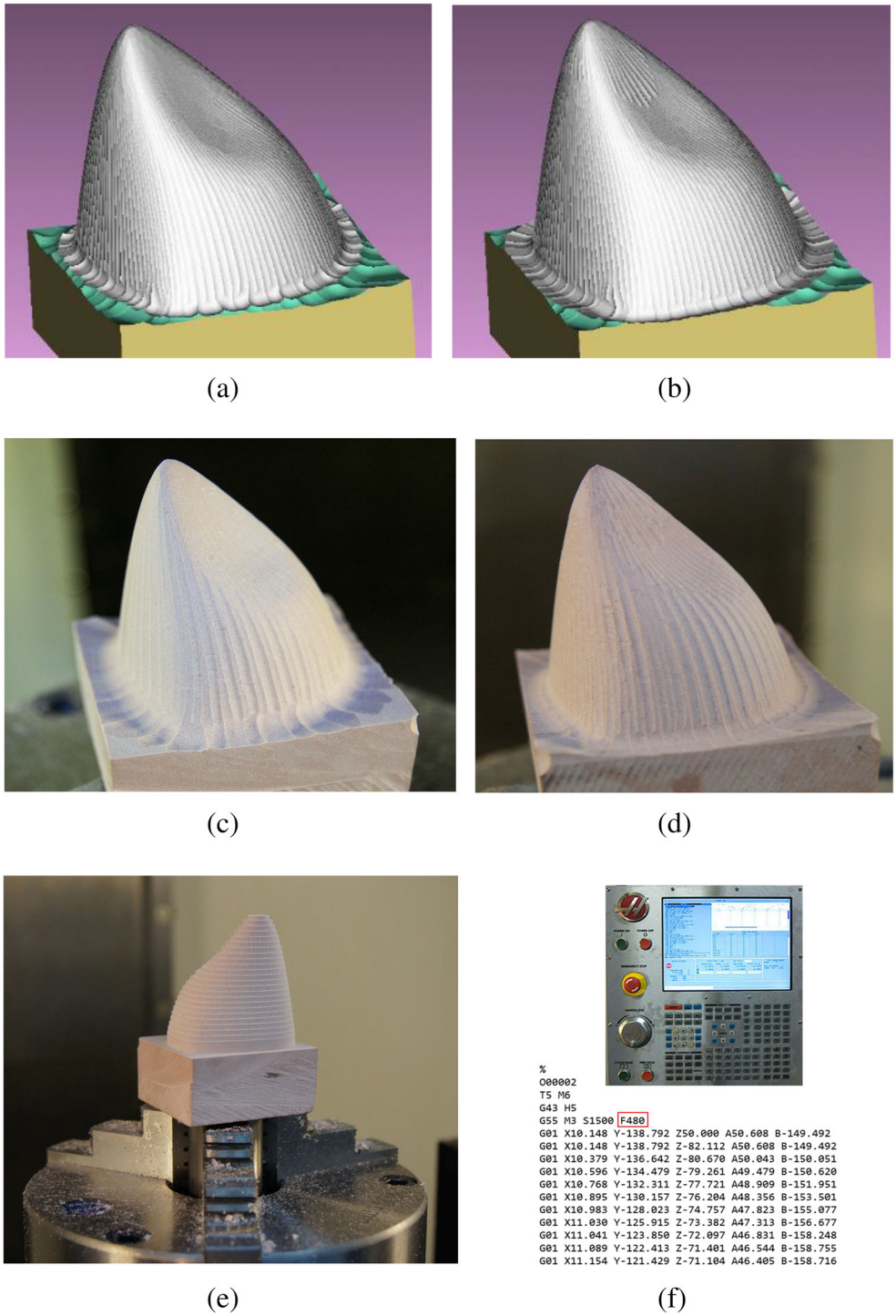


Fig. 23 Virtual and real cutting for the canine crown, $h = 0.1$. **a** SRZ, virtual cut. **b** CRZ, virtual cut. **c** Semi-finish SRZ (52 min). **d** CRZ (40 min). **e** Workpiece clamped on the rotary table (rough cut). **f** A sample G code on the Haas VF-2TR controller

7 Region around the pole

In many cases, the region around the pole can be milled by a *mini-CP* to reduce the redundancy. Specifically, the CP is an excellent option when the pole area is nearly flat. For instance, in example 2, we replaced the radial pattern in the last CRZ layer by a small CP pattern, shown in Fig. 30. The total machining time for the original CRZ is 458 s, while the CRZ combined with a CP pattern requires only 414 s. However, this option is not recommended when the last layer is characterized by a high curvature. For instance, for the surface in example 3, we also replaced the radial pattern near the pole by the mini-CP. The machining time of the new toolpath is 259 s, which is much longer than the 126 s of the original CRZ. Note that this combination creates a new modification of the CRZ or CRZT called the CRZT/CP path.

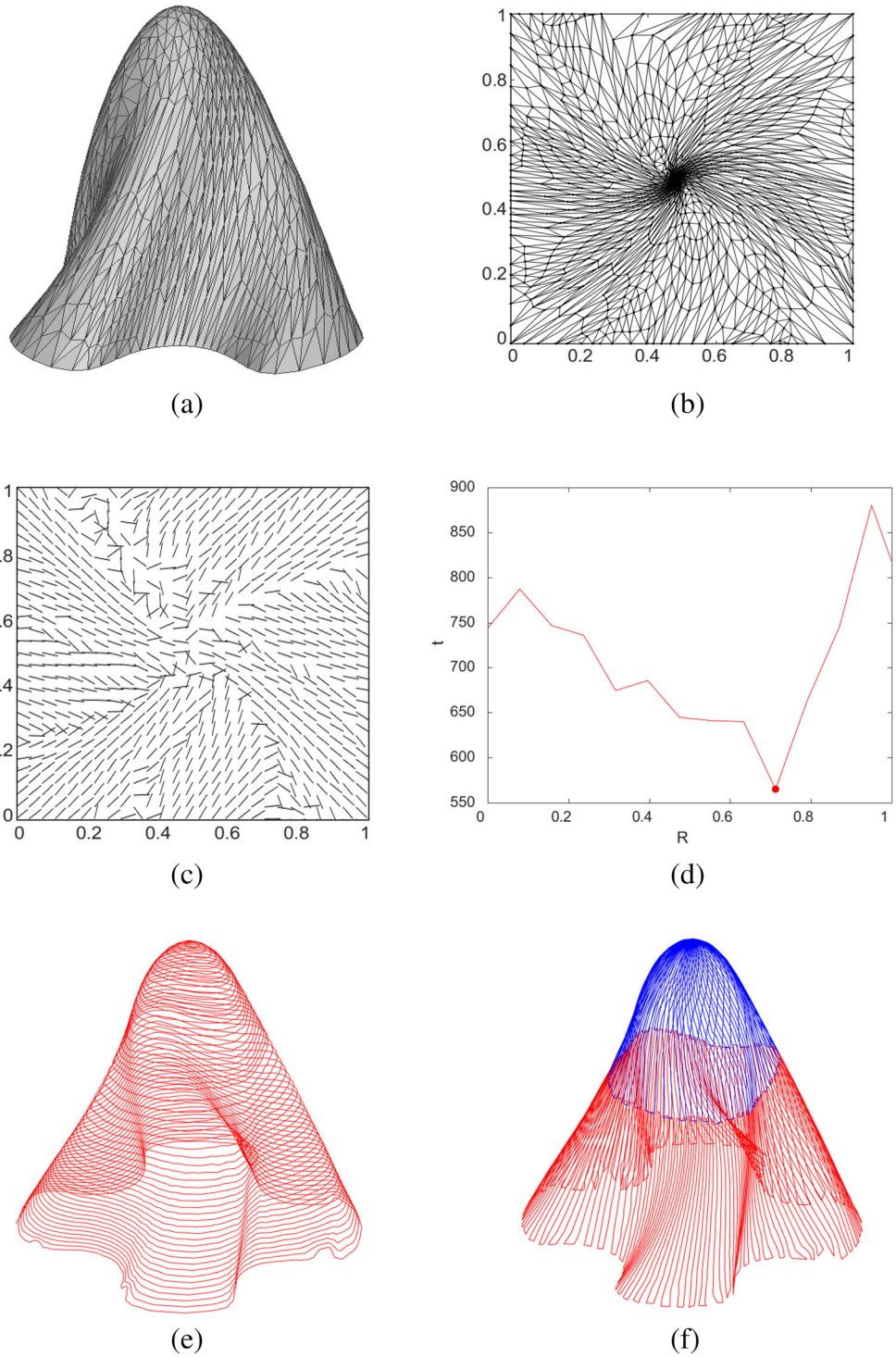


Fig. 24 Twisted STL surface, CP, and CRZ toolpath. **a** Twisted surface. **b** flattened twisted surface. **c** OF of the EMRR. **d** 1st partition, $R_1^* = 0.71$. **e** CP. **f** CRZ

8 Conclusions

A new compact radial zigzag or CRZ toolpath for five-axis machining has been introduced and validated on STL surfaces. The internal radial layers eliminate the redundancy and improve the machining time. The toolpath in each layer follows the VFPD while the step over is close to the largest.

For certain surfaces, the CRZ strategy reduces the machining time by 77%. However, the algorithm is not suitable for every surface. A large allowable scallop height jeopardizes the

Table 5 Testing the proposed method on a twisted surface (example 4)

Example 4: twisted surface	h = 0.1 mm				h = 0.5 mm				h = 1.0 mm			
	L (mm)	N_{CC}	T (s)	A (%)	L (mm)	N_{CC}	T (s)	A (%)	L (mm)	N_{CC}	T (s)	A (%)
Iso-parametric												
ZZ	5612	3793	487	52.7	2405	851	184	54.3	1785	550	124	60.4
CP	3997	3221	649	64.5	2647	1130	215.9	61.0	2509	798	93	47.3
SRZ	7181	6336	653	64.7	3052	1642	204	58.8	2064	987	115	57.3
ISOP	4903	4099	357	35.5	2428	1450	104.4	19.5	1871	1039	83.7	41.4
CRZ	5681	2245	251	8.3	2565	1003	86	2.3	1895	805	63	22.2
CRZT	*5153*	*2059*	*230*	$N_P = 1$	*2374*	*865*	*84*	$N_P = 1$	*1636*	*400*	*49*	$N_P = 1$

The notations are given in Table 2 and in the text

Fig. 25 **a** Rough cut, CP. **b** Semi-finish CP (40 min). **c** Semi-finish CRZ (28 min)

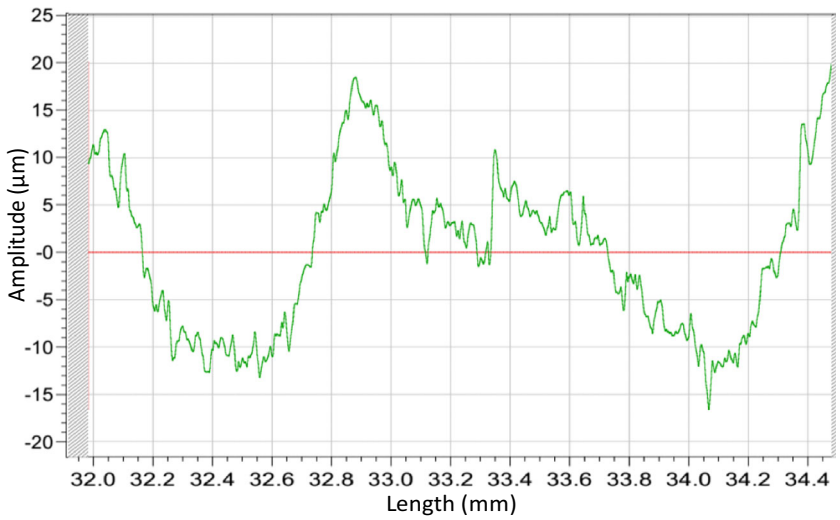


Fig. 26 Random sample of the amplitude of the CRZ surface

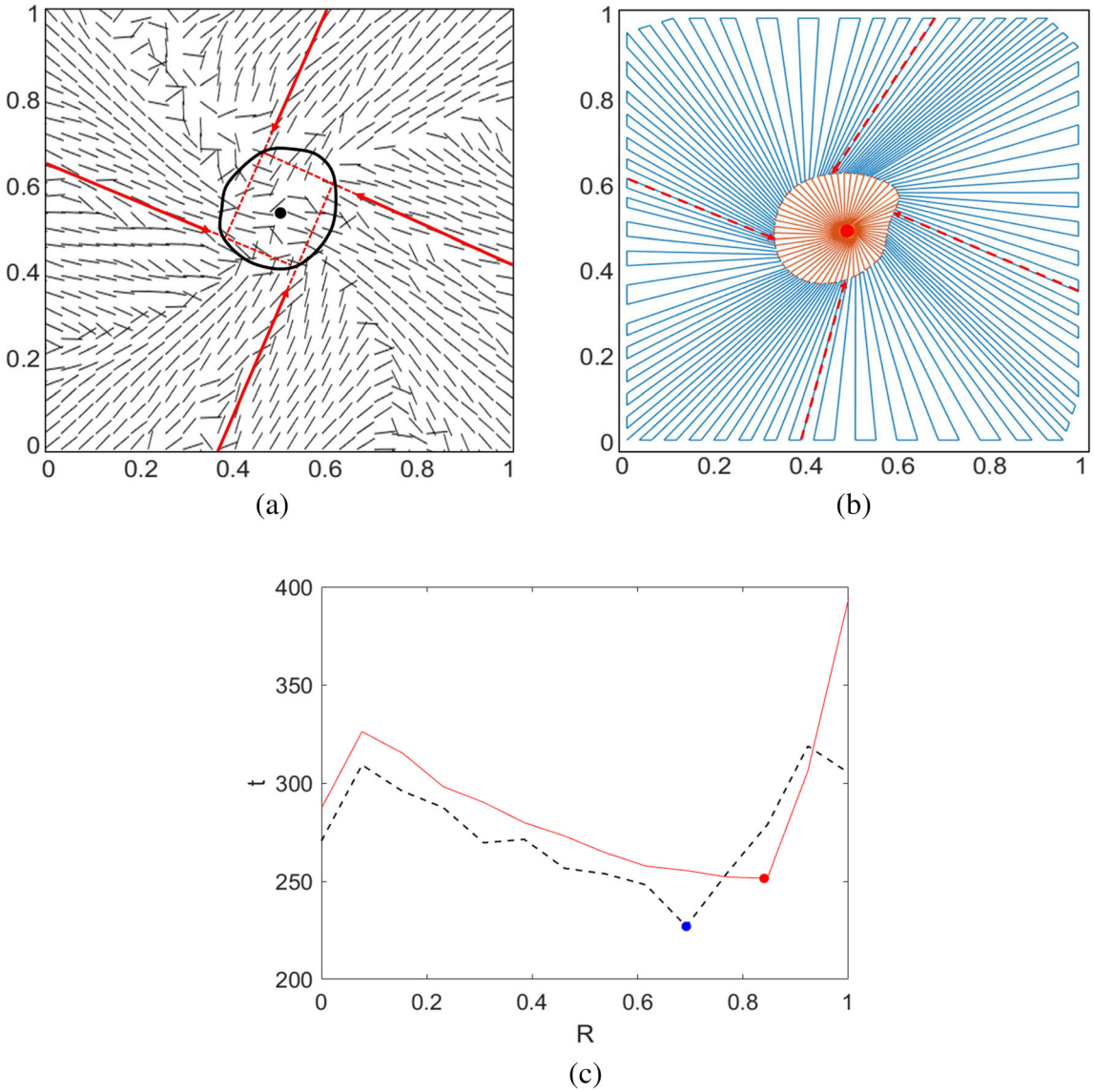


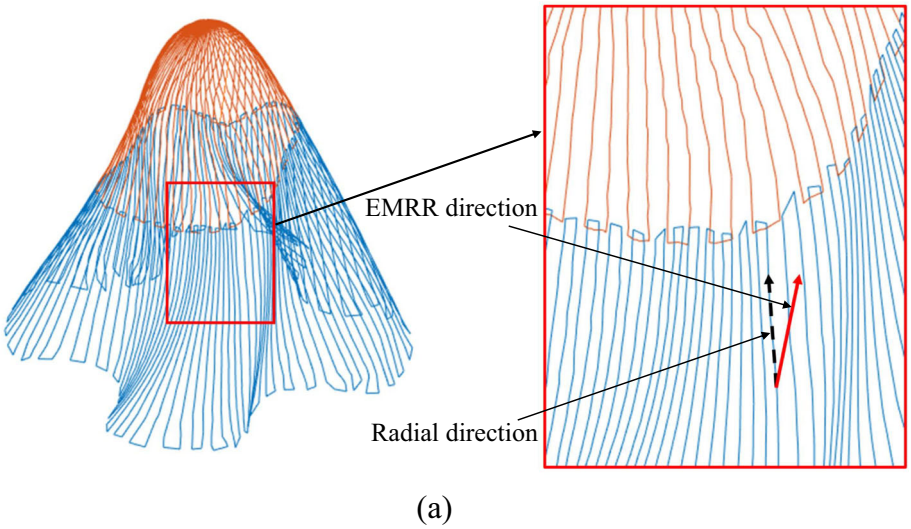
Fig. 27 CRZ and CRZT toolpath generation for the twisted surface (example 4). **a** Convergence region for the twisted surface. **b** CRZT path, important vectors, convergence region. **c** Solid line: CRZ partition, $R_1^* = 0.84$; dashed line: CRZT partition, $R_1^* = 0.72$

advantage of the CRZ. Our experiments show that for $h = 1$ mm, the CRZ is slower than ZZ and ISOP for a relatively flat (saddle) surface. This situation can be partially improved by using CRZT; however, the impact and applicability of CRZT is still an open problem.

Nevertheless, if the curvature of the surface is high and the required scallop height is relatively low (semi-finish or finish cut), we conjecture that the CRZ toolpath always wins in the case of a radial or an approximately radial pattern. The method is not necessarily applicable to an arbitrary surface. However, a test proposed in Section 4.2 is an excellent criterion of applicability.

Real cutting demonstrates that the roughness and the kinematic error of the CRZ surfaces are in the same range as that obtained by the conventional methods, while the machining time is always shorter (except for the cases mentioned above).

The proposed toolpath strategy is efficient when the part surface is partitioned into subregions characterized by different topologies of the VFPD. These topologies include the radial patterns around sharp peaks and valleys of complex-shaped sculptured surfaces, e.g., medical implants. With the current state of the manufacturing technology, the market demand for such parts is steadily growing. Therefore, the proposed technology is promising. There is no doubt that this method should be included in the set of so-called standard



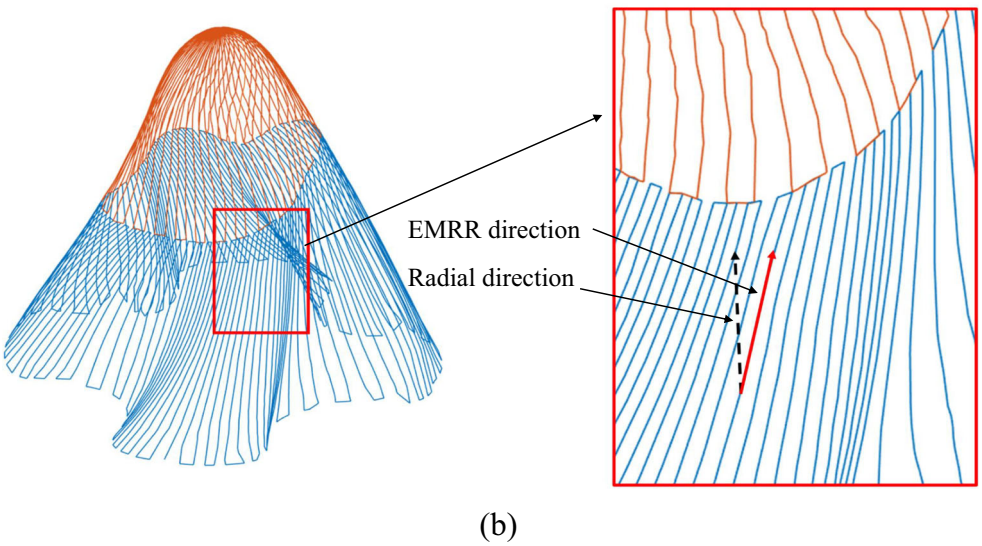


Fig. 28 CRZ and CRZT (example 4). **a** CRZ toolpath. **b** CRZT toolpath

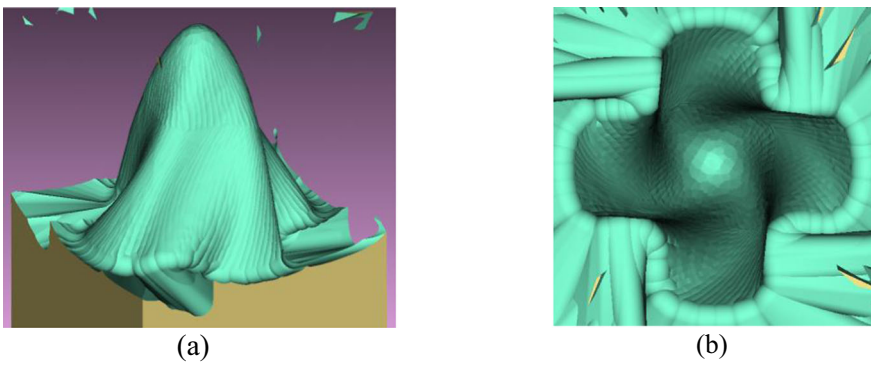


Fig. 29 Virtual cut, CRZT, twisted surface (example 4). **a** CRZT virtual machining (side view). **b** CRZT virtual machining (top view)

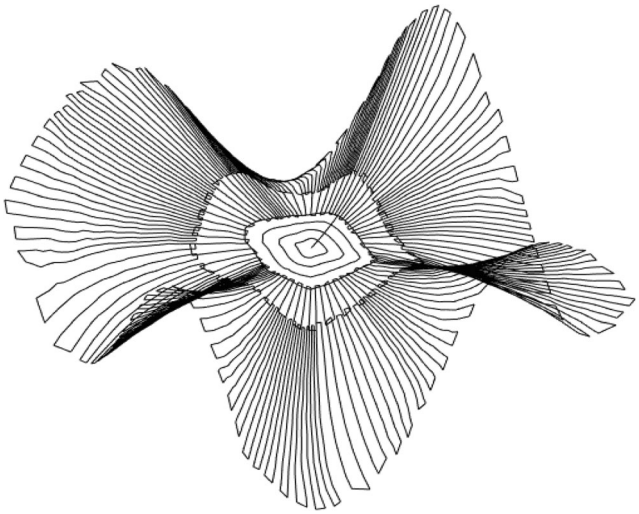


Fig. 30 CRZ toolpath with the CP pattern around the pole (example 2) (40 s advantage)

machining patterns, such as the zigzag, iso-planar, and spiral patterns.

Funding information This research is supported by the Center of Excellence in Biomedical Engineering, Thammasat University, Thailand.

Appendix. The kinematic error

To compare the accuracy of the proposed method, we evaluate the kinematic errors induced by non-linear trajectories of the machine for examples 3 and 4. Note that since the surfaces are approximated by the STL mesh, the testing includes several sources of error. The kinematic error depends on the accuracy of the approximation of the desired surface by the STL mesh (quality of triangulation). Furthermore, the procedure includes barycentric interpolation, transforming the triangulated surfaces into the Cartesian system. This transformation invokes certain numerical errors. If a CC curve on the real surface is characterized by a smooth variation of the rotation angles, the CC curve on the STL surface is piecewise linear. The corresponding normals and rotation angles could change abruptly, leading to substantial kinematic errors. A certain toolpath may pass through such singularities and generate large kinematic errors, whereas a different configuration may eventually avoid these errors. However, we have included these inaccuracies in our evaluation.

Table 6 Kinematic error for surfaces in examples 3 and 4

Kinematic error (mm)	Dental crown surface Example 3		Twisted surface Example 4	
	ε_{max}	ε'	ε_{max}	ε'
ZZ	0.192	0.0035	0.192	0.0088
CP	0.190	0.0054	0.161	0.0149
SRZ	0.191	0.0133	0.184	0.0147
ISOP	0.195	0.0049	0.192	0.0070
CRZ	*0.187*	*0.0046*	*0.119*	*0.0066*

The toolpaths produced by NX have not been included since they have been generated from the solid models rather than from the STL. Although our experiments show that their kinematic errors are in the same range, the Appendix compares the errors obtained only on the STL meshes.

Furthermore, we exclude the kinematic error at the pole. Our approach is not to cross the peak (singularity) which inevitably leads to large kinematic errors [72, 73]. The trajectory loops invoked by the singularity can destroy the workpiece or even cause a global collision. Therefore, when the SRZ and CRZ reach the singularity point, they turn back, following the next track (see our demo in the Abstract). The CP cuts around the singularity as in Section 7. Finally, if the ZZ or ISOP path runs near the singularity, we withdraw the tool, turn it in the air, and continue the cut from the other side (plunging [74], see also Fig. 31). Note that there exist a number of efficient

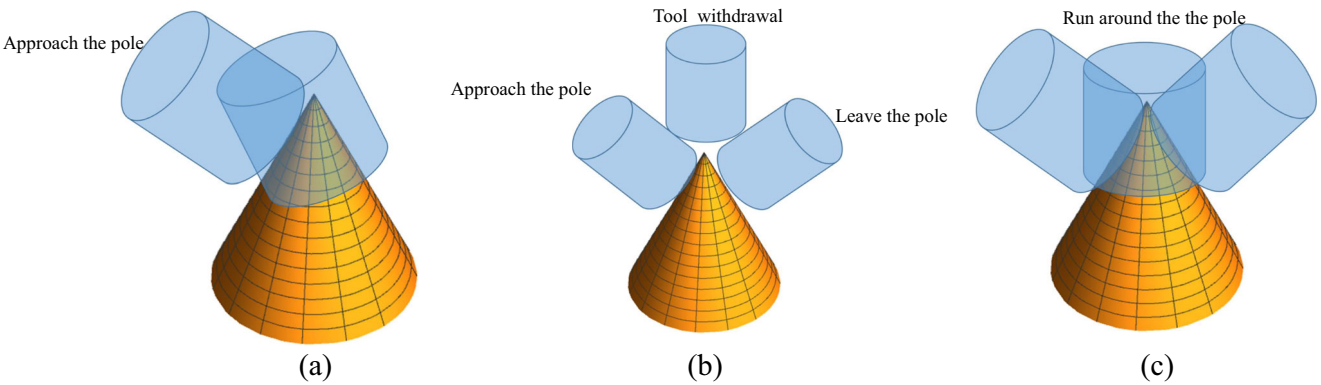


Fig. 31 Practical treatment of the singularity. **a** ZZ path. **b** SRZ and CRZ path. **c** CP path

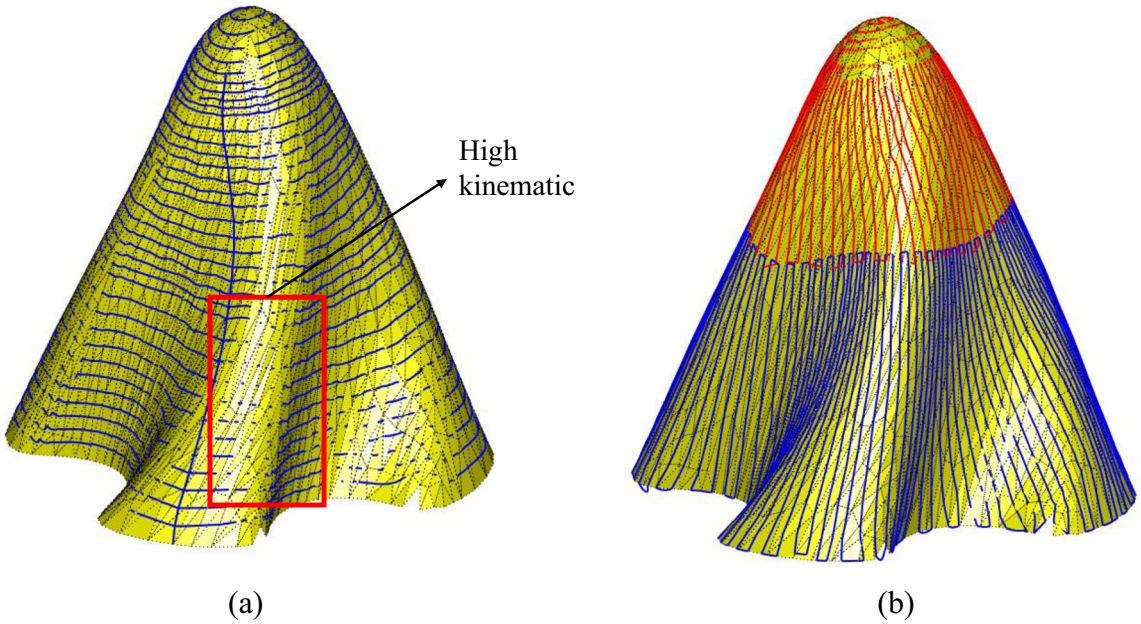


Fig. 32 Tool trajectories on the twisted surface. **a** CP path. **b** CRZ path

methods to avoid singularities [75–77]; however, this experiment compares a *regular* kinematic error that appears when cutting a relatively smooth surface characterized by the radial VFPD. Statistically, the singularity is an outlier, which must be excluded.

Table 6 shows $\varepsilon_{\max}$ and ε' obtained by the ZZ, CP, SRZ, ISOP, and CRZ toolpaths. Each toolpath has a different length. Therefore, we measure the kinematic error per unit length of the forward step (0.5 mm). The size of the forward step was selected so that $h \approx 0.1$.

The results show that the maximum and average kinematic errors by the CRZ are in the same range as the errors produced by competing methods; however, *the machining time is consistently shorter*.

The kinematic error is graphically illustrated in Fig. 32a and b. The CP trajectories substantially deviate from the desired surface (in example 4) due to the large kinematic error, whereas the CRZ tool trajectories follow the low curvature and, therefore, are lying on the surface.

References

1. Dragomatz D, Mann S (1997) A classified bibliography of literature on NC milling path generation. Comput Aided Des 29:239–247
2. Lasemi A, Xue D, Gu P (2010) Recent development in CNC machining of freeform surfaces: a state-of-the-art review. Comput Aided Des 42:641–654
3. Makhanov SS (2010) Adaptable geometric patterns for five-axis machining: a survey. Int J Adv Manuf Technol 47:1167–1208
4. Feng H-Y, Li H (2002) Constant scallop-height tool path generation for three-axis sculptured surface machining. Comput Aided Des 34: 647–654
5. Ding S, Mannan MA, Poo AN, Yang DCH, Han Z (2003) Adaptive iso-planar tool path generation for machining of free-form surfaces. Comput Aided Des 35:141–153
6. Suresh K, Yang DCH (1994) Constant scallop-height machining of free-form surfaces. J Eng Ind 116:253–259
7. Lo C-C (1999) Efficient cutter-path planning for five-axis surface machining with a flat-end cutter. Comput Aided Des 31:557–566
8. Lee E (2003) Contour offset approach to spiral toolpath generation with constant scallop height. Comput Aided Des 35:511–518
9. Marshall S, Griffiths JG (1994) A new cutter-path topology for milling machines. Comput Aided Des 26:204–214
10. Chiou CJ, Lee YS (2002) A machining potential field approach to tool path generation for multi-axis sculptured surface machining. Comput Aided Des 34:357–371
11. Liu W, Zhou L-S, An L-L (2012) Constant scallop-height tool path generation for three-axis discrete data points machining. Int J Adv Manuf Technol 63:137–146
12. Hu P, Chen L, Tang K (2017) Efficiency-optimal iso-planar tool path generation for five-axis finishing machining of freeform surfaces. Comput Aided Des 83:33–50
13. Zou Q, Zhang J, Deng B, Zhao J (2014) Iso-level tool path planning for free-form surfaces. Comput Aided Des 53:117–125
14. Kim T, Sarma SE (2002) Toolpath generation along directions of maximum kinematic performance; a first cut at machine-optimal paths. Comput Aided Des 34:453–468
15. Hu P, Tang K (2016) Five-axis tool path generation based on machine-dependent potential field. Int J Comput Integr Manuf 29:636–651
16. Makhanov SS (1999) An application of the grid generation techniques to optimize a tool-path of industrial milling robots. J Comput Math Phys 39:1589–1600
17. Makhanov SS, Ivanenko SA (2010) Grid generation as a new concept of CNC-based part optimization. In: IMACS Word Congress on Computational Mathematics and Simulations, 21-25 August, Switzerland

18. Bieterman MB, Sandstrom DR (2003) A curvilinear tool-path method for pocket machining. J Manuf Sci Eng 125: 709–715
19. Chen L, Hu P, Luo M, Tang K (2018) Optimal interface surface determination for multi-axis freeform surface machining with both roughing and finishing. Chin J Aeronaut 31:370–384
20. Moodleah S, Bohez EJ, Makhanov SS (2016) Five-axis machining of STL surfaces by adaptive curvilinear toolpaths. Int J Prod Res 54: 7296–7329
21. Kumazawa GH, Feng HY, Barakchi Fard MJ (2015) Preferred feed direction field: a new tool path generation method for efficient sculptured surface machining. Comput Aided Des 67–68:1–12
22. Liu X, Li Y, Ma S, Lee CH (2015) A tool path generation method for freeform surface machining by introducing the tensor property of machining strip width. Comput Aided Des 66:1–13
23. Teng Z, Feng H-Y, Azeem A (2006) Generating efficient tool paths from point cloud data via machining area segmentation. Int J Adv Manuf Technol 30:254–260
24. Chen ZC, Dong Z, Vickers GW (2003) Automated surface subdivision and tool path generation $3\frac{1}{2}\frac{1}{2}$-axis CNC machining of sculptured parts. Comput Ind 50:319–331
25. Tuong NV, Pokorný P (2010) A practical approach for partitioning free-form surfaces. Int J Comput Integr Manuf 23:992–1001
26. Wang N, Tang K (2008) Five-axis tool path generation for a flat-end tool based on iso-conic partitioning. Comput Aided Des 40:1067–1079
27. Gordon WJ, Hall CA (1973) Construction of curvilinear co-ordinate systems and applications to mesh generation. Int J Numer Methods Eng 7:461–477
28. The Siemens PLM software. https://www.plm.automation.siemens.com/en/docs/nx/11.shtml. Accessed 13 Aug 2018
29. Ozturk E, Tunc LT, Budak E (2009) Investigation of lead and tilt angle effects in 5-axis ball-end milling processes. Int J Mach Tools Manuf 49:1053–1062
30. Makhanov SS, Anotaipaiboon W (2007) Advanced numerical methods to optimize cutting operations of five-axis milling machines. Springer
31. Zhang K, Tang K (2014) An efficient greedy strategy for five-axis tool path generation on dense triangular mesh. Int J Adv Manuf Technol 74:1539–1550
32. Moodleah S, Makhanov SS (2015) Five-axis machining using a curvilinear tool path aligned with the direction of the maximum removal rate. Int J Adv Manuf Technol 80:65–90
33. Flusser J, Zitova B, Suk T (2009) Moments and moment invariants in pattern recognition. John Wiley & Sons
34. Floater MS, Hormann K (2005) Surface parameterization: a tutorial and survey. In: Advances in multiresolution for geometric modelling. Springer, Berlin, Heidelberg, pp 157–186
35. Sheffer A, Praun E, Rose K (2007) Mesh parameterization methods and their applications. Found Trends Comput Graph Vis 2:105–171
36. Tutte WT (1963) How to draw a graph. Proc Lond Math Soc 3:743–767
37. Lévy B, Petitjean S, Ray N, Maillot J (2002) Least squares conformal maps for automatic texture atlas generation. ACM Trans Graph 21:362–371
38. Sheffer A, Lévy B, Mogilnitsky M, Bogomyakov A (2005) ABF++: fast and robust angle based flattening. ACM Trans Graph 24:311–330
39. Desbrun M, Meyer M, Alliez P (2002) Intrinsic parameterizations of surface meshes. Comput Graph Forum 21:209–218
40. Floater MS (1997) Parametrization and smooth approximation of surface triangulations. Comput Aided Geom Des 14: 231–250
41. Xu J, Sun Y, Wang S (2013) Tool path generation by offsetting curves on polyhedral surfaces based on mesh flattening. Int J Adv Manuf Technol 64:1201–1212
42. Yuwen S, Dongming G, Zhenyuan J, Haixia W (2006) Iso-parametric tool path generation from triangular meshes for free-form surface machining. Int J Adv Manuf Technol 28:721–726
43. Ren F, Sun Y, Guo D (2009) Combined reparameterization-based spiral toolpath generation for five-axis sculptured surface machining. Int J Adv Manuf Technol 40:760–768
44. Sun Y, Ren F, Zhu X, Guo D (2012) Contour-parallel offset machining for trimmed surfaces based on conformal mapping with free boundary. Int J Adv Manuf Technol 60:261–271
45. Shu CF, Jain RC (1993) Direct estimation and error analysis for oriented patterns. CVGIP Image Underst 58:383–398
46. Schlemmer M, Heringer M, Morr F, Hotz I, Bertram MH, Garth C, Kollmann W, Hamann B, Hagen H (2007) Moment invariants for the analysis of 2D flow fields. IEEE Trans Vis Comput Graph 13: 1743–1750
47. Corpetti T, Mémin E, Pérez P (2003) Extraction of singular points from dense motion fields: An analytic approach. J Math Imaging Vis 19:175–198
48. Koch S, Kasten J, Wiebel A, Scheuermann G, Hlawitschka M (2016) 2D Vector field approximation using linear neighborhoods. Vis Comput 32:1563–1578
49. Liu W, Ribeiro E (2012) Detecting singular patterns in 2D vector fields using weighted Laurent polynomial. Pattern Recogn 45: 3912–3925
50. Gonzalez RC, Woods RE (2002) Digital image processing, 2nd edn. Prentice Hall
51. Zhang Q, Yan H (2007) Fingerprint orientation field interpolation based on the constrained Delaunay triangulation. Int J Inf Syst Sci 3:438–452
52. Hu MK (1962) Visual pattern recognition by moment invariants. IRE Trans Inf Theory 8:179–187
53. Flusser J, Suk T (1994) Affine moment invariants: a new tool for character recognition. Pattern Recogn Lett 15:433–436
54. Campisi P, Neri A, Panci G, Scarano G (2004) Robust rotation-invariant texture classification using a model based approach. IEEE Trans Image Process 13:782–791
55. Srikanth C, Deekshatulu B, Rao C, Bhagvati C (2009) Classification and identification of Telugu Aksharas using moment invariants and C4.5 algorithms. Int J Comput Intell Res 5:225–232
56. Shu CF, Jain RC (1994) Vector field analysis for oriented patterns. IEEE Trans Pattern Anal Mach Intell 16:946–950
57. Lin R-S, Koren Y (1996) Efficient tool-path planning for machining free-form surfaces. J Eng Ind 118:20–28
58. Petitjean S (2002) A Survey of methods for recovering quadrics in triangle meshes. ACM Comput Surv 34:211–262
59. Goldfeather J, Interrante V (2004) A novel cubic-order algorithm for approximating principal direction vectors. ACM Trans Graph 23:45–63
60. Sun Y, Xu J, Jin C, Guo D (2016) Smooth tool path generation for 5-axis machining of triangular mesh surface with nonzero genus. Comput Aided Des 79:60–74
61. Chen X, Schmitt F (1992) Intrinsic surface properties from surface triangulation. In: European Conference on Computer Vision. Springer, Berlin, Heidelberg, pp 739–743
62. Martin RR (1998) Estimation of principal curvatures from range data. Int J Shape Model 4:99–109
63. Cohen-Steiner D, Morvan J-M (2003) Restricted Delaunay triangulations and normal cycle. In: Proceedings of the 19th Annual Symposium on Computational Geometry. ACM, pp 312–321
64. Alliez P, Cohen-Steiner D, Devillers O, Lévy B, Desbrun M (2003) Anisotropic polygonal remeshing. In: ACM Transactions on Graphics (TOG). ACM, pp 485–493
65. Rusinkiewicz S (2004) Estimating curvatures and their derivatives on triangle meshes. In: Proceedings. 2nd International Symposium

on 3D Data Processing, Visualization and Transmission. 3DPVT 2004. IEEE, pp 486–493
66. Cipolla R, Giblin P (2000) Visual motion of curves and surfaces. Cambridge University Press
67. Telea A, Van Wijk JJ (1999) Simplified representation of vector fields. In: Proceedings Visualization '99: Celebrating Ten Years. IEEE Computer Society Press, pp 35–42
68. Wang Q-H, Zhang X-M, Li J-R, Tang C-S (2015) Prediction of toolpath redundancy for NC machining of free-form surfaces based on automatic recognition of steep-wall features. Int J Prod Res 53: 4304–4316
69. Knorr EM, Ng RT (1999) Finding intensional knowledge of distance-based outliers. Proceedings 25th International Conference on Very Large Data Bases (VLDB 1999). pp 211–222
70. ISO 4278: Geometrical Product Specifications (GPS) – Surface Texture (1997) Profile method terms. Defin Surf Texture Parameters 1st ed
71. The Matlab Library. https://www.mathworks.com/products/matlab.html. Accessed 13 Aug 2018
72. Makhanov SS, Munlin M (2007) Optimal sequencing of rotation angles for five-axis machining. Int J Adv Manuf Technol 35:41–54
73. Anotaipaiboon W, Makhanov SS (2011) Minimization of the kinematics error for five-axis machining. Comput Aided Des 43:1740–1757
74. Sun C, Wang Y, Huang N (2015) A new plunge milling tool path generation method for radial depth control using medial axis transform. Int J Adv Manuf Technol 76:1575–1582
75. Tournier C, Duc E (2005) Iso-scallop tool path generation in 5-axis milling. Int J Adv Manuf Technol 25:867–875
76. Wang Y, Yan C, Yang J, Lee CH (2017) Tool path generation algorithm based on covariant field theory and cost functional optimization and its applications in blade machining. Int J Adv Manuf Technol 90:927–943
77. Lin Z, Fu J, Yao X, Sun Y (2015) Improving machined surface textures in avoiding five-axis singularities considering tool orientation angle changes. Int J Mach Tools Manuf 98:41–49