The International Journal of Advanced Manufacturing Technology (2018) 98:2831–2847
https://doi.org/10.1007/s00170-018-2463-2

ORIGINAL ARTICLE

Smooth fillet-end cutter tool path generation method on triangular-mesh surface based on Modified Butterfly subdivision

Qirui Wang[1] · Yixiong Feng[1] · Yicong Gao[1] · Zhiwu Li[2] · Jianrong Tan[1]

Received: 7 March 2018 / Accepted: 17 July 2018 / Published online: 25 July 2018

Abstract
Triangular-mesh surfaces are widely applied in the design of aerospace, automobiles, ships, etc., because of their high computational efficiency and robustness compared with parametric surfaces. In order to improve the tool path continuity and reduce the cutter load fluctuation, a smooth tool path generation method on triangular-mesh surfaces based on Modified Butterfly subdivision is proposed. The machining model at the cutter-contact point is constructed based on the geometric analysis of the triangular-mesh surface and fillet-end cutter. The minimum cutter tilt angle and maximum machining stripe width are obtained to avoid the interference of cutter and workpiece. And then, the Modified Butterfly subdivision method is applied to the triangular-mesh surface to ensure the tool path interval within the strip width. The boundary of the surface is selected as the initial tool path based on which the subsequent tool paths are generated along the edge of triangular mesh, and the cutter location points are obtained according to the geometric model of the fillet-end cutter. The Discrete Domain of Feasible Orientation (DDFO) model is constructed to eliminate the singularity problem of the machining process, and Modified Butterfly subdivision is applied recurrently to guarantee the whole surface within the scallop height requirement. Finally, the spiral tool path with good continuity and smoothness is generated according to the specified cutter parameter and scallop height requirement, which can reduce the cutter load fluctuation.

Keywords NC machine · Tool path · Triangular-mesh surface · Modified Butterfly subdivision

1 State of the art

Free-form surfaces are widely used in the design of aerospace, automobile, ship, and other products. Free surfaces can be expressed not only with parametric equations, but also with polygon meshes. The expression precision of parametric equations is obviously superior to polygon meshes, but its complexity on the boundary leads to much more computational cost in joining or trimming two parametric surfaces. On the contrary, the simple and effective principles of joining or trimming two polygon surfaces guarantee the computational robustness. In addition, there are no high-order equations to be resolved in intersection of polygon surfaces. With the reverse engineering and rapid prototyping technology development, free-form surfaces composed of polygon meshes, especially triangular meshes, are widely used in the field of Computer Aided Design (CAD) and Computer Aided Manufacturing (CAM). Tool path planning on triangular-mesh surfaces has also drawn lots of researchers' attention.

Kim [1] proposed a method of generating offset contour, which is used to optimize quality of corner cutting and control the cutting force to improve the cutting stability. However, the surface features of free-form surfaces are complex and not suitable for the generation of the offset path. The Z-level algorithm [2] and contour algorithm [3] are widely deployed in high speed machining, effectively reducing machine tool load fluctuations in the process of high-speed machining. Nevertheless, defects of the Z-level algorithm and contour algorithm are also obvious: if the angle between the cutting surface and the contour surface is too small, the abnormal density of tool path may reduce the machining efficiency. Too much retraction of the machining process is avoided and thus machining efficiency is improved by Lee [4], which

✉ Yixiong Feng
fyxtv@zju.edu.cn

[1] State Key Laboratory of Fluid Power and Mechatronic Systems, Zhejiang University, Hangzhou 310027, China

[2] Institute of Systems Engineering, Macau University of Science and Technology, Taipa, Macau

takes the curvature of cutter contact points (CC point) into consideration to generate the tool path by non-uniformly offsetting the initial tool path and connecting each machining area. However, the complexity of splicing surfaces reduces the stability of the method. Sun [5] proposed a new method of generating spiral or contour-parallel tool path is proposed, which is inspired by the cylindrical helix or circle which are a set of parallel lines on the rectangular region obtained by unwrapping the cylinder. An intuitive man-machine interface is used by Balasubramaniam [6] to propose a tool path generation method which includes two phases: recording of access directions at the surface of the object and the post-processing phase. Zhou et al. [7] proposed a smooth tool path generation method based on a heat transfer model. The isotherm of the heat transfer model is mapped to the spiral tool path by calculating the cutting stripe width on the machined surface. Chen and Zhao [8] applied self-adaptive Loop subdivision to construct the finish machining model, in which the machined surface is classified into flat areas and steep areas and different machining strategies are implemented on each kind of area. Zhang et al. [9] proposed a method to generate the tool path with good continuity by iteratively subdividing the mesh surface, which effectively reduces the fluctuation of cutter load and improves the machining accuracy and efficiency. The ball-end cutter used in the methods above may lead to the too small path interval which will reduce the machining efficiency. In order to ensure the machining efficiency of the proposed method, fillet-end cutter is applied to generate the smooth tool path.

In recent years, lots of researchers have focused their attention on smoothing the tool orientation sequence. As inverse kinematic transformation of the machine tool changes along with its configuration, Kim and Sarma [10] realized that the same tool orientation sequence can lead to different movements of axes on different machine tools. On this basis, DAO theory was proposed by Castagnetti et al. [11]. In DAO theory, the whole kinematic chain is considered to construct the mapping relation between the error of tool orientation and error of machine tool axes. It can be realized that the most straightforward and efficient way to make the tool orientation sequence smooth is to optimize the motion of rotary axes. In order to optimize the machining performance, the machining time model based on feedrate, acceleration, and jerk profiles is constructed in [12]. Jun et al. [13] proposed a methodology and algorithms of optimizing and smoothing the tool orientation control for 5-axis sculptured surface machining. A searching method in the machining configuration space (C-space) is proposed to find the optimal tool orientation by considering the local gouging, rear gouging, and global tool collision in machining.

As non-linear optimization is involved in the tool orientation smoothing problem, a sequential quadratic programming (SQP) method was implemented to find the solution of optimizing the tool orientation sequence. However, the disadvantage of SQP method is obvious that the initial guess is absolutely stochastic, and the accuracy of optimal sequence depends on the discrepancy between the global optimization and initial guess. Besides, some graphic-based methods are implemented to optimize the tool orientation sequence. For example, feed direction and tool orientation are varied using the simplex method by Fard and Feng [14]. On the basis of admissible-map (A-map) theory, the smooth-map (S-map) theory is proposed by Li et al. [15].

There are also a lot of heuristic algorithms implemented in the tool orientation optimization. Ho et al [16] proposed a novel tool orientation smoothing method to control the nonlinear error. The quaternion interpolation algorithm is applied to obtain the orientation sequence based on the selected typical tool orientations. Then, the initial tool orientation sequence was smoothened after the interference detection. The particle swarm optimization (PSO) method has also been widely applied in tool orientation smoothing, such that the machining error can be successfully minimized. However, the performance of PSO method is unacceptably poor when the number of cutter locations is large [17].

As for the aforementioned state about the generation method of smooth tool path, the general thought is generating the initial tool path, then optimizing it. It requires a margin before generating the initial tool path. The instability and the unpredictability of the aforementioned methods may lead to dead end of optimization, even form some defects on the machined surface. In order to improve the stability of the optimization and eliminate the defects, Modified Butterfly subdivision is applied to generate new meshes such that the tool path along the edges can make whole surface within the scallop height requirement. At the same time, the tool orientation should be smoothened to guarantee the machining quality. There are three highlights of our work: (1) iteratively applying local subdivision to generate the initial tool path to make the whole surface within the scallop height requirement; (2) discretizing the admissible arc at each CC point to construct a Discrete Domain of Feasible Orientations (DDFO) model, based on which the tool orientation sequence is smoothened; (3) recalculating the cutting stripe width of every feed movement of the cutter and subdividing the triangular meshes which contains remnant islets to retrieve the degradation of cutting stripe width caused by smoothing the tool orientation sequence.

The rest of the paper is organized as follows. In the following section, the geometry features of the cutter and the workpiece at the CC point are analyzed. After that, in Sect. 3, the Modified Butterfly subdivision is applied to generate initial tool path and the tool orientation sequence is established. In order to smooth the initial path generated in Sect. 3, the DDFO model is constructed and the corresponding solving method based on Dijkstra algorithm is proposed in Sect. 4. Section 5 eliminates the cutting stripe width diminishment problem by

iteratively applying Modified Butterfly subdivision. Section 6 gives the simulation and comparison experiments to prove the validity of the proposed method. Finally, the conclusion of the method and the discussion of future work close this paper in Sect. 7.

2 Local geometry analysis of CC point

2.1 Geometry analysis of cutter surface

Ball-end cutter, flat-end cutter, and fillet-end cutter are widely used in NC machining, since they have their own advantages and disadvantages. The machining speed at the tip of ball-end cutter is zero, and the low machining speed near the cutter tip will lead to the poor machining quality. The small tool radius selected to avoid interference will make the tool paths abnormally dense and reduce the machining efficiency. The equivalent curvature of the flat-end cutter changes with the cutter posture, which makes it fit uneven surfaces, but the quality of the machined surface deteriorates with cutter abrasion. The fillet-end cutter has the advantages of the ball-end cutter and flat-end cutter, making it competent for the high precision machining of free surface.

Primarily, the work-piece coordinate system is defined as the basic system to describe positions of the cutter. At the CC point, a local coordinate system $CX_CY_CZ_C$ is defined as shown in Fig. 1. X_C represents the instantaneous feed direction, Z_C is the normal vector of the work-piece surface at CC point, and Y_C is determined by the right-hand rule. A cutter coordinate system is defined at the center point of the cutter bottom circle O which is set as the origin of the coordinate system. As for the determination of X_t and Y_t, the direction of X_t and Y_t should be initialized, respectively, the same as X_C and Y_C. The cutter rotates about axis Y_C for the tilt angle β with point O as the fulcrum and then rotates about axis Z_C for the angle rotary angle σ with point C as the fulcrum.

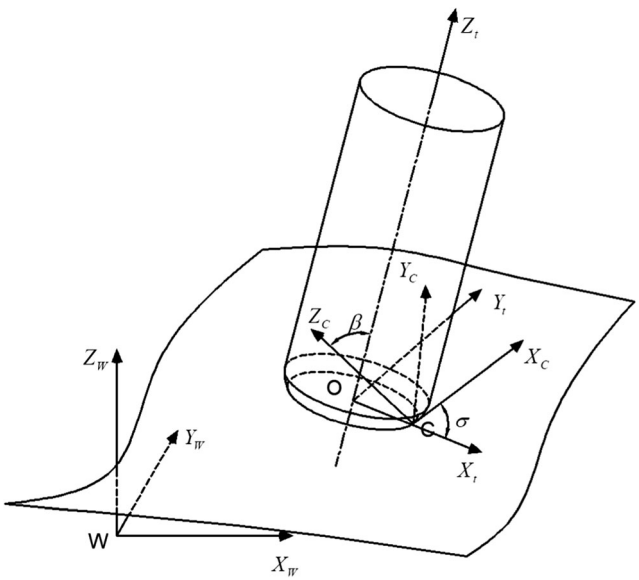

Fig. 1 Five-axis machining of fillet-end cutter

The work-piece is actually machined by the cutter envelop surface and is formed by the instantaneous characteristic curve, which is the actual cutting edge. Assuming that, in our case, the global collision has been already considered, normally the rotary angle σ is set to be 0 to make the tilt angle dominate the cutter posture. With the cutter radius R and charmer radius r given, the equivalent radius of the cutter at CC point can be estimated as [18]

$$r_e = \frac{1}{\kappa_e} = \frac{R + r\cdot \sin\beta}{\sin\beta} \tag{1}$$

where κ_e is the equivalent curvature of the cutter at CC point.

2.2 Geometry analysis of work-piece surface

On the triangular-mesh surface, the curvatures at each vertex have magnificent impact on the cutting stripe width and the feed direction. The vertex sequence $V = \{V_1, V_2, \ldots, V_N\}$ which contains all the vertexes of the surface is constructed to record the position of each vertex. And each edge is expressed as a tuple $E_{ij} = \{V_i, V_j\}$, with which the triad of triangular mesh is established as $F_{ijk} = \{E_{ij}, E_{jk}, E_{ik}\}$. All the following analysis and calculation are operated on the aforementioned data sets to generate the spiral smooth tool path.

Since the vertex V_i does not belong to only one triad of triangular mesh, each triangular mesh in the 1-ring neighborhood of V_i influences the curvature at vertex V_i. The larger the triangular mesh is, the greater its impact is. As the normal of each triangular mesh can be calculated accurately, the normal at the vertex V_i can be estimated based on the normal of meshes in the 1-ring neighborhood of V_i, as expressed by Eq. (2) [19].

$$\mathbf{n_i} = \frac{\sum_{x_i \in F_k} |F_k| \mathbf{n}_{\mathbf{F_k}}}{\left\| \sum_{x_i \in F_k} |F_k| \mathbf{n}_{\mathbf{F_k}} \right\|} \tag{2}$$

where $|F_k|$ is the area of the triangular mesh which contains vertex V_i, and $\mathbf{n}_{\mathbf{F_k}}$ is the normal vector of the triangular mesh which contains vertexV_i.

In order to estimate the curvature along the direction $\overrightarrow{V_iV_j}$, an osculating circle is constructed with $|V_iV_j|$ as a chord and the normal at the point V_i as a diameter. Thus, the curvature at the point V_i along the direction $\overrightarrow{V_iV_j}$ can be estimated, as expressed by Eq. (3):

$$\kappa_{ij} = \frac{2\left(V_\mathbf{i} - V_\mathbf{j}\right)\cdot \mathbf{n_i}}{\left\| V_\mathbf{i} - V_\mathbf{j} \right\|^2} \tag{3}$$

where $\boldsymbol{n}_i$ is the unit normal at the point V_i, and the mean curvature of point V_i can be estimated in Eq. (4).

$$H_i = \frac{1}{4A_{mixed}} \sum_{V_j \in Neighbor(V_i)} \left(\cot\alpha_{ij} + \cot\beta_{ij}\right)\left(\mathbf{V_i} - \mathbf{V_j}\right) \cdot \mathbf{n_i} \quad (4)$$

where α_{ij} and β_{ij} are the opposite angles of the edge $|V_iV_j|$ in triangles that contain the edge $|V_iV_j|$, as shown in Fig. 2. And A_{mixed} is the mixed area of triangles which contains the vertex V_i. The mixed area of a triangle T can be estimated by the following pseudo-code [20].

```
A_mixed = 0
For each triangle T of 1-ring neighborhood of vertex V_i
{
    If T is non-obtuse
        A_mixed += A_Voronoi (A_Voronoi = 1/8 (cot α_ij + cot β_ij) ||V_i - V_j||)
    Else {
        If the angel of triangle T at vertex V_i is obtuse
        A_mixed += area(T) / 2
        Else
        A_mixed += area(T) / 4
        }
}
```

As the scallop height requirement is set to be h, the area along the cutting stripe with the scallop under h is defined as the valid cutting stripe. As the normal curvature at the point V_i is $\boldsymbol{n_i} \times (V_i - V_j) = 2H_i - \kappa_{ij}$, the width of valid cutting stripe along the edge $|V_iV_j|$ can be estimated as expressed by Eq. (5).

$$d_{ij} = \sqrt{\frac{8h}{\kappa_{tool} - \left(2H_i - \kappa_{ij}\right)}} \quad (5)$$

where κ_{ij} is the curvature along the direction $\overrightarrow{V_iV_j}$ and κ_{tool} is the curvature of the equivalent cutting edge on the cutter.

3 Generation of the initial tool path using Modified Butterfly subdivision

3.1 The tool orientation analysis based on the interference of the cutter and the work-piece

For the sake of avoiding the interference between the cutter and the work-piece, the tilt angle and rotary angle at each CC point are confined to control the cutting condition. The tilt angle β defines the angle of the cutter rotating about the axis Y_C, and similarly, the rotary angle σ defines the angle of the cutter rotating about the axis Z_C.

The equivalent cutting edge of the fillet-end cutter is a quasi-arc lying on the envelop surface of the cutter, and it changes with the cutter posture. From Eq. (5), it can be found that if the rotary angle σ remains unchanged, the cutting stripe width changes with the change of the tilt angle β. With different tilt angles β_1 and β_2 ($\beta_1 > \beta_2$), it can be found that the corresponding radii of the equivalent cutting edge have the relation: $r_{e1} < r_{e2}$. Thus, if the scallop height h is preset, the corresponding cutting stripe widths satisfy $D_1 < D_2$, as shown in Fig. 3. According to the aforementioned analysis, it can be concluded that the cutting stripe width D and the tilt angle β are correlated reciprocally with the given rotary angle γ. Similarly, if the tilt angle β is given, the cutting stripe width D has negative relation with the rotary angleγ.

In order to simplify the model, we set the rotary angle $\gamma = 0$, and the tilt angle is adjusted to control the cutting stripe width and to avoid interference. At the cutter-contact point, the equivalent curvature of the cutting edge on the fillet-end cutter

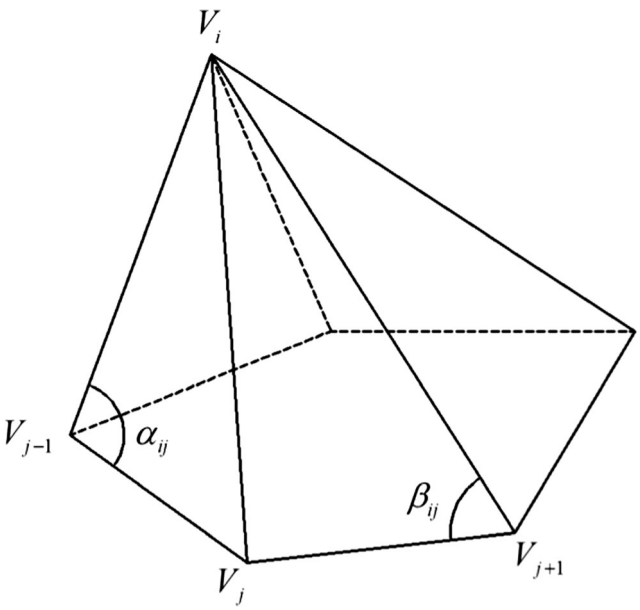

Fig. 2 The vertex model of the mesh surface

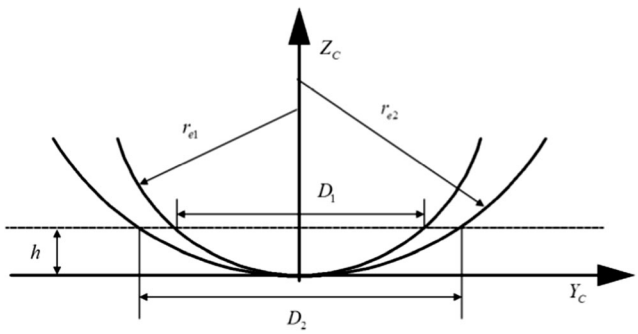

Fig. 3 The equivalent cutting edge of the fillet-end cutter

along the feed direction is $\kappa_{t1} = 1/r$, and the equivalent curvature perpendicular to the feed direction is

$$\kappa_{t2} = \kappa_e = \frac{\sin\beta}{r \cdot \sin\beta + R} \tag{6}$$

It is obvious that $\kappa_{t1} > \kappa_{t2}$, and the corresponding equivalent radii have inverse relation, i.e., $R_{t1} < R_{t2}$. In order to avoid interference, the equivalent radius of the fillet-end cutter shall be set less than the curvature of the surface at the cutter-contact point, i.e.,

$$R_{t1} \le \frac{1}{\kappa_{\max}} \quad \text{(Along the feed direction)} \tag{7}$$

$$R_{t2} \le \frac{1}{\kappa_{\max}} \quad \text{(Perpendicular to the feed direction)} \tag{8}$$

where $\kappa_{\max}$ is the maximum principal curvature of the machined surface at the cutter-contact point. By $R_{t1} < R_{t2}$, the requirement $R_{t2} \le \frac{1}{\kappa_{\max}}$ is the sufficient condition for the avoidance of interference. Thus, the minimum non-interference tilt angle is $\beta_{\min} = \arcsin\left(R/\left(\frac{1}{\kappa_{\max}} - r\right)\right)$. Considering the negative correlation of the tilt angle β and the cutting stripe width D, the maximum stripe width along the edge V_iV_j can be obtained by substituting $\beta_{\min}$ to Eqs. (7) and (8).

3.2 Local subdivision of the triangular-mesh surface using Modified Butterfly subdivision

Spiral tool path [21] runs continuously along the edges of triangular meshes could reduce the cutter retractions, tool wear, and machining time. However, in our case, due to the limit of the tilt angle β, the cutting stripe width of each feed step has an upper limit, and this is why the central area of some triangular meshes cannot fulfill the scallop height requirement, as shown in Fig. 4. These areas which are named as "remnant islets" may cause the deterioration of the machining quality and local interference. As new edges generated on the triangular meshes could increase the density of the tool path to eliminate the remnant islets on the machined surface, subdivision algorithm could be applied to eliminate remnant islets on the surface. It should be noted that the subdivision method is not applied to every mesh as the way it is used in graphic optimization, but just applied to triangular meshes, those cannot fulfill the scallop height requirement.

The four common-used subdivisions, i.e., Loop subdivision [22], Catmull-Clark subdivision [23], Butterfly subdivision [24], and Doo-Sabin [25] subdivision, have their own specialties. Doo-Sabin subdivision may change the mesh type and that would be an obstacle to generate a neat spiral tool path. Catmull-Clark subdivision can keep the surface C^2 continuous, but its subdivision strategy does not work well on triangular meshes. Loop subdivision is a kind of approximating method which could smoothen the surface but cannot maintain the sharp features, and it will enlarge the machining error to the unacceptable degree. By contrast, Modified Butterfly subdivision is an interpolating method which can maintain the original features and works well on triangular meshes.

In order to eliminate the remnant islets to make the whole surface within the scallop requirement, each triangular mesh is firstly inspected whether there is a remnant islet inside. Before the tool path is finally generated on the edges of triangular meshes, we do not know which edge is used. So, we need to propose a criterion which could guarantee a triangular mesh ΔF_{ijk} is within the scallop height requirement no matter which edge of ΔF_{ijk}. In ΔF_{ijk} as shown in Fig. 5, assuming that V_iV_j is used to generate the tool path, the cutting stripe d_{ij} width should cover the whole triangular mesh ΔF_{ijk} to fulfill the scallop requirement. In other words, if the vertex V_k is not covered in the cutting stripe, the scallop height cannot be guaranteed and the triangular mesh needs to be subdivided. Considering the situation of generation tool path on $|V_iV_k|$ and $|V_jV_k|$, the triggering condition of subdivision could be expressed as Eq. (9)

$$\begin{aligned}&\left((|V_iV_k| \cdot \cos V_i) \ge \frac{1}{2}d_{ij}\right) \| \left(\left(|V_jV_k| \cdot \cos\angle V_k\right) \ge \frac{1}{2}d_{ik}\right) \\ &\quad \| \left(\left(|V_iV_j| \cdot \cos\angle V_j\right) \ge \frac{1}{2}d_{jk}\right) = 1\end{aligned} \tag{9}$$

where d_{ij}, d_{ik}, and d_{jk} are, respectively, the cutting stripe width along the edge $|V_{ij}|$, $|V_{ik}|$, and $|V_{jk}|$. Then, the Modified Butterfly subdivision is applied to generate new meshes until all the remnant islets are eliminated, i.e.,

$$\begin{aligned}&\left((|V_iV_k| \cdot \cos V_i) \ge \frac{1}{2}d_{ij}\right) \| \left(\left(|V_jV_k| \cdot \cos\angle V_k\right) \ge \frac{1}{2}d_{ik}\right) \\ &\quad \| \left(\left(|V_iV_j| \cdot \cos\angle V_j\right) \ge \frac{1}{2}d_{jk}\right) = 0\end{aligned} \tag{10}$$

In Modified Butterfly subdivision, the position of the newly generated point on the edge is determined by two end points of the edge and points in its 1-ring neighborhood. The subdivision strategy can be divided into four categories.

(1) If two end points of the edge are both regular, the subdivision follows the mask shown in Fig. 6a.
(2) If one end point of the edge is regular and the other is irregular, the subdivision follows the mask as shown in Fig. 6b. The weight of the regular point is $Q = 0.75$, and the weight of the irregular point can be estimated as expressed by Eq. (11):

$$s_i = \frac{0.25 + \cos\frac{2\pi i}{n} + 0.5\cos\frac{4\pi i}{n}}{n} (i = 1, 2, 3, \ldots, n-1) \tag{11}$$

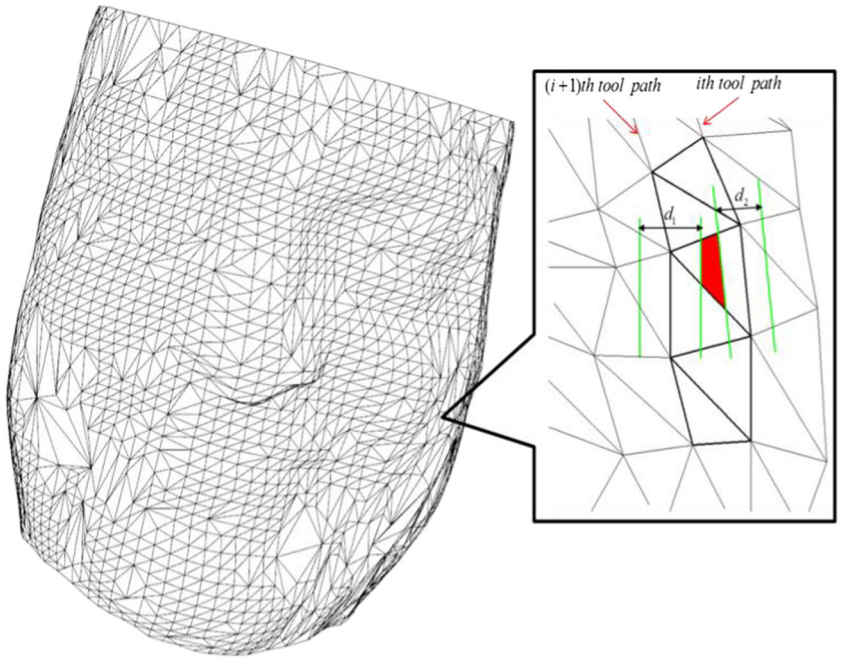

Fig. 4 The generation of remnant islets

where n is the number of points in the irregular point's 1-ring neighborhood.

(3) If two end points are both irregular, the interpolating point can be generated by three steps. Firstly, one end point is assumed to be regular, then an interpolating point can be generated following the mask as shown in Fig. 6b; secondly, the other end point is assumed to be regular, and another interpolating point can be generated similarly; finally, the midpoint of interpolating points generated from the aforementioned two steps is interpolated on the edge.

(4) If the edge is on the boundary of the surface, the interpolating point can be generated according to Eq. (12), and the weights of the related points are shown in Fig. 6c, i.e.,

$$V_E = -\frac{1}{16}V_{l-1} + \frac{9}{16}V_l + \frac{9}{16}V_{l+1} - \frac{1}{16}V_{l+2} \tag{12}$$

where V_l and V_{l+1} are the end points of the edge, while V_{l-1} and V_{l+2} are the end points of adjacent edges on the boundary.

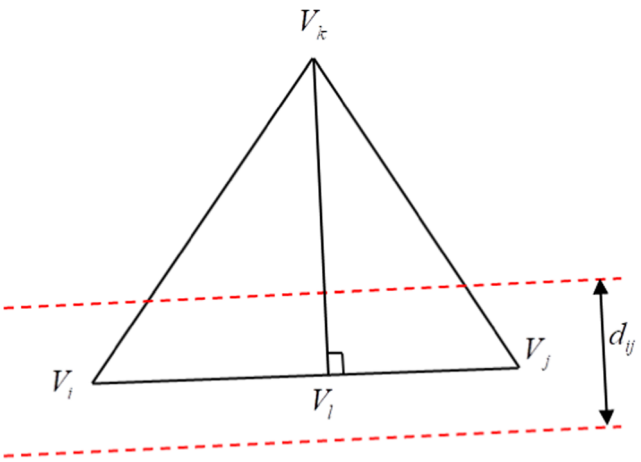

Fig. 5 An arbitrary triangular mesh on the surface

3.3 The generation of the initial tool path

The designed surface is machined by the cutter along the edges of meshes to generate the spiral tool path. The more continuous the tool path is, the less cutter lifting there will be. And fluctuation of the cutting load will be reduced by the increasing continuity of the tool path, and thus, the abrasion of the cutter can be alleviated to improve the machining accuracy and extend the service life of the cutter. As every triangular mesh of the surface is within the scallop height requirement after the Modified Butterfly subdivision is applied, the initial tool path will be generated along the edges of the triangular meshes.

Definition 1: Reference point. The end point of the generated tool path is defined as the reference point which is the reference in the generation of the next CC point, as shown in Fig. 7.

Definition 2: Reference row. The latest row of CC tool path is defined as the reference row which is the offset reference to generate the new row of CC tool path, as shown in Fig. 7.

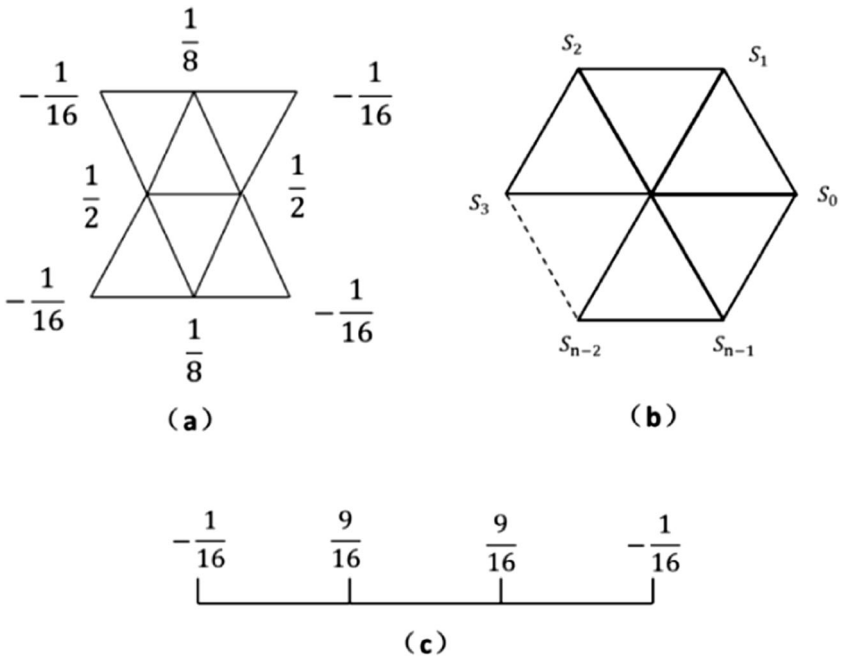

Fig. 6 The mask of Modified Butterfly subdivision

In order to manage the tool path data, the data set of each vertex is constructed as

$$G(V_i) = \{P(V_i), U(V_i)\} \tag{13}$$

where $P(V_i)$ is the position data of vertex V_i in the coordinate system $WX_WY_WZ_W$, and $U(V_i)$ is the label recording whether vertex V_i is used or not. Thus, the data set of each vertex can be expressed using the following pseudo-code:

```
class  V_i
{
    public:
    double x, y, z;\\ recording the position of the vertex
    bool u; \\ recording whether the vertex is used or not
}
```

As the tool path generated on the surface is spiral, the recognition of the boundary is the precondition of the tool path generation. The recognition of the boundary can be carried on by the following three steps:

(1) During the traverse of all the triangular meshes, each triangular mesh is expressed as a triad with three edges contained in it, i.e., $F_{ijk} = \{E_{ij}, E_{jk}, E_{ik}\}$, and the belongingness of each edge is inspected from an arbitrary edge. The searching stops when there is an edge belonging to only one triangular mesh, and the edge is selected as the start edge of the CC tool path. Thus, vertexes of the start edge are marked as used, and one of the vertexes is selected as the reference point.
(2) Similarly, the belongingness of each vertex in 1-ring neighborhood of the reference point is inspected and the edge which belongs to only one triangular mesh is selected and added to the tool path.
(3) The searching stops until the boundary edge with a reference point and an unused vertex cannot be found and then a close boundary of the triangular-mesh surface can be generated. The boundary is set as the first row of the tool path, and both reference point list and reference row list are filled in order to generate the next row of the tool path.

The CL position $\mathbf{O}_t$ and orientation vector $\mathbf{T}_t$ of the fillet-end cutter can be expressed in coordinate $OX_tY_tZ_t$, as expressed by Eq. (14).

$$\begin{aligned} \mathbf{O}_t &= (0\ 0\ 0\ 1)^T \\ \mathbf{T}_t &= (0\ 0\ 1\ 0)^T \end{aligned} \tag{14}$$

And they could be switched to the form in coordinate $CX_CY_CZ_C$, as expressed by Eqs. (15) and (16).

$$\mathbf{O_C} = \begin{pmatrix} -(R\cos\beta + r\sin\beta)\cos\gamma \\ -(R\cos\beta + r\sin\beta)\sin\gamma \\ R\sin\beta - r\cos\beta \\ 1 \end{pmatrix} \tag{15}$$

$$\mathbf{T_C} = \begin{pmatrix} \sin\beta\cos\gamma \\ \sin\beta\sin\gamma \\ \cos\beta \\ 0 \end{pmatrix} \tag{16}$$

Thus, the transformation matrix from the coordinate $CX_CY_CZ_C$ to the coordinate $WX_WY_WZ_W$ is

$$\mathbf{M} = \begin{pmatrix} \mathbf{i_c} & \mathbf{j_c} & \mathbf{k_c} & \mathbf{C_{i,j}} \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{17}$$

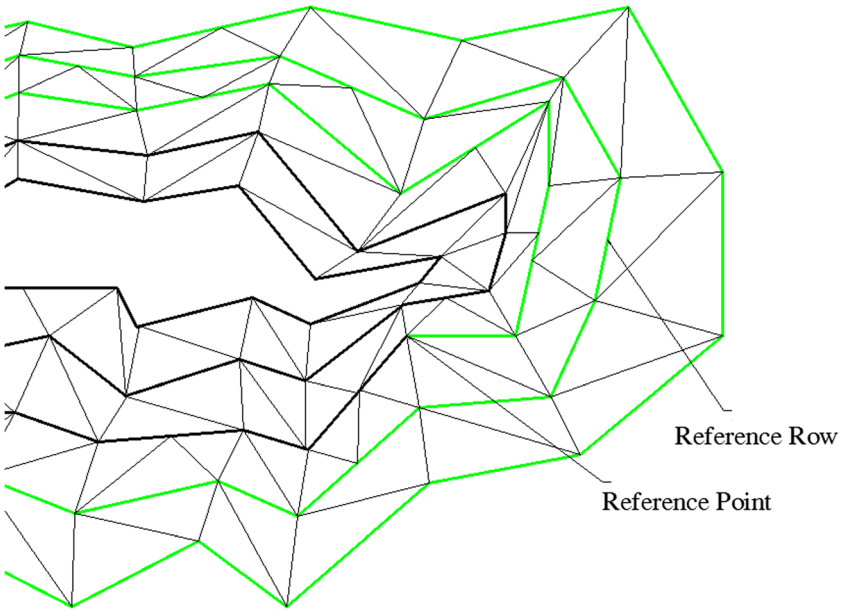

Fig. 7 Definitions of reference row and reference point

where $\boldsymbol{i_c}, \boldsymbol{j_c}, \boldsymbol{k_c}$ are the expression forms of the base vectors of coordinate $CX_CY_CZ_C$ in coordinate $WX_WY_WZ_W$, and $\mathbf{C_{i,j}}$ is the origin of the coordinate $CX_CY_CZ_C$ expressed in coordinate $WX_WY_WZ_W$. Thus, the CL data of tool path in coordinate $WX_WY_WZ_W$ is

$$\begin{aligned} \mathbf{O_w} &= \mathbf{MO_c} \\ \mathbf{T_w} &= \mathbf{MT_c} \end{aligned} \tag{18}$$

4 Smoothing of the tool orientation sequence

The initial tool path generated according to Sect. 3 could guarantee every triangular mesh within the scallop height requirement. However, abrupt changes of tool orientation and rotary axes would lead to the existence of scratches on the machined surface [26]. For the sake of machining quality, smoothing the tool orientation sequence is an important issue to improve the performablity of the tool path. If the tool orientation is only determined by the inclination angle, the admissible domain can be visualized as an arc, named a feasible arc, lying between the corresponding minimal and maximal inclination angles as shown if Fig. 8. When the minimal inclination angle is reached, the cutter may penetrate into surface with a smaller inclination angle. Similarly, the front side of a cutter could touch a part surface when the maximal inclination angle is exceeded. In the cutter's feasible range which is defined by the maximal and minimal inclination angles, the cutting interference can be avoided.

Smoothing the tool orientation sequence is to minimize the movements of rotary axes while avoiding singularity problems; hence, the smoothing method should concentrate on how the rotary axes operate considering inverse kinematics of machine tool.

4.1 The construction of the discrete domain of feasible orientation (DDFO) model

Establishing the mapping relation of tool orientation and its corresponding movements of two rotary axes is the basic issue of smoothing the tool orientation sequence. A tool orientation vector T is firstly considered on the Gaussian sphere in the workpiece-system as shown in Fig. 9. As concluded in [27], the posture of cutter is only determined by movements of rotational axes. Whether the intersecting point of axes A and C coincides with the origin point of work-piece coordinate system makes no difference in tool orientation analysis. So, a unit sphere with its center at the origin point of work-piece coordinate system is constructed, as shown in Fig. 9. According to the configuration of AC-type machine tool, T can rotate about the X-axis and Z-axis in the workpiece-system. The initial tool orientation T is set to be parallel to Z-axis,

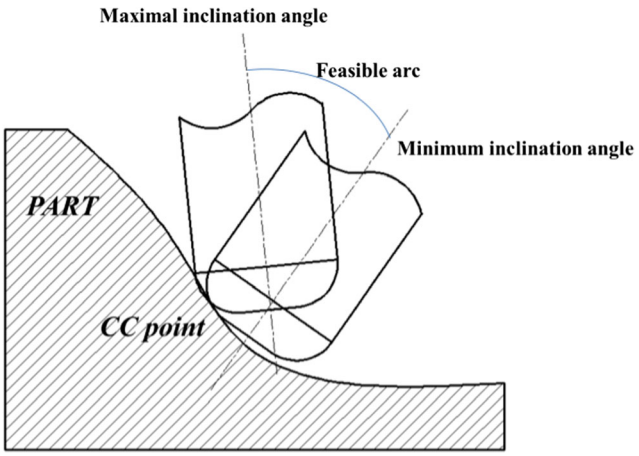

Fig. 8 The feasible arc at an arbitrary CC point

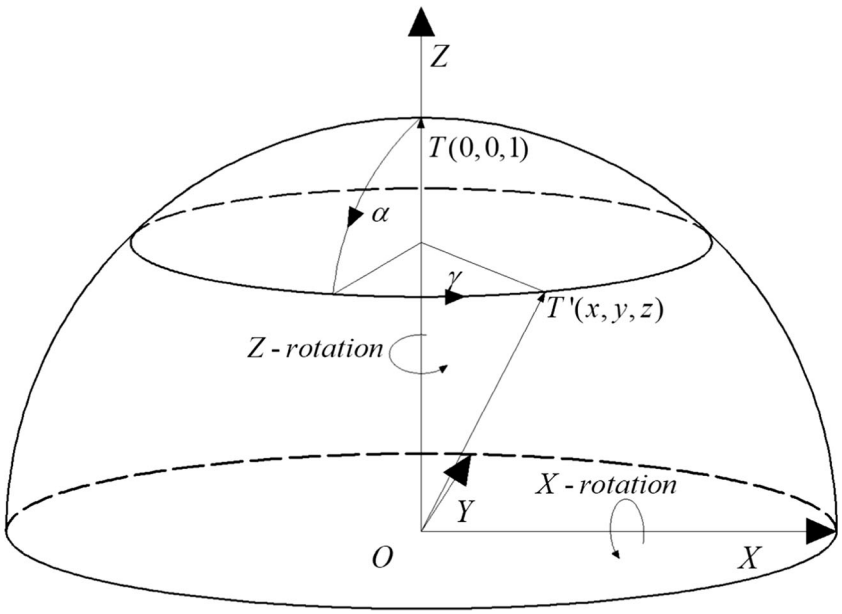

Fig. 9 The tool orientation unit vectors on the Gaussian sphere

i.e., $T = (0, 0, 1)$. An arbitary vector $T^{'} = (x, y, z)$ in workpiece-system can be reached from T by two rotational movements: (1) T rotates for an angle α about X-axis, and (2) T' rotates for an angle γ about Z-axis, as shown in Fig. 9. It should be clearly illustrated that both rotary movements are operated simultaneously. And the angles α and γ can be obtained by Eq. (19):

$$\begin{cases} \sin\alpha \cdot \sin\gamma = x \\ \sin\alpha \cdot \cos\gamma = -y \\ \cos\alpha = z \end{cases} \tag{19}$$

where x, y, and z are the coordinate values of $T^{'}$. In the above two rotary movements, the first rotation is named as X-rotation. Similarly, the second one is named Z-rotation. As an arbitrary vector, $T^{'}$ in the workpiece-system can be reached by rotating the initial vector T, $T^{'}$ can be expressed as $T^{'}(\alpha, \gamma)$. As for the AC-type machine tool with an orthogonal tilting table, the X-rotation and Z-rotation are, respectively, rotation movements of A-axis and C-axis. In other words, an arbitrary tool orientation vector can be expressed as

$$T^{'}(A, C) = T^{'}(\alpha, \gamma) \tag{20}$$

where A and C are, respectively, rotary angles of A-axis and C-axis from the initial tool orientation $T(0,0,1)$.

As the feasible range of the tool orientation can be regarded as an arc which is defined by the feasible inclination angle boundaries $[\tau_{\min}, \tau_{\max}]$, such that τs at each CC point can be adjusted to achieve the optimal tool orientation sequence.

Inspired by [26], it is realized that discretizing the feasible domain at each CC point is a reasonable approach to find the smooth tool orientation sequence. Before constructing the smoothing model, the distance between two adjacent tool postures, namely the posture distance, should be defined in detail.

As two rotary axes of a machine tool are simultaneously driven by their respective motors, the posture distance can be defined as the maximal rotary angle of A-axis and C-axis, as expressed by Eq. (21). Additionally, when major arcs exist in the rotary movements of the machine tool, the posture distance can be further illustrated as Eqs. (22) and (23), i.e.,

$$D_m = \max(\Delta A_m, \Delta C_m) \tag{21}$$

where

$$\Delta A_m = |A_{m+1} - A_m| \tag{22}$$

$$\Delta C_m = \begin{cases} |C_{m+1} - C_m| (|C_{m+1} - C_m| \le \pi) \\ 2\pi - |C_{m+1} - C_m| (|C_{m+1} - C_m| > \pi) \end{cases} \tag{23}$$

The cost index D_m is defined according to Chebyshev distance. Considering the periodicity of the C-axis rotary movement, if the angle of Z-rotational $\Delta\gamma$ is larger than 180°, the reverse rotation which is less than 180° will be chosen to be executed by the CNC system, as expressed by Eq. (23). In different machining cases, appropriate weight coefficients could scale the distance definitions given in Eqs. (22) and (23). Hence, the objective function can be expressed by Eq. (24):

$$\min_{m} \left\{ \sum_{m=1}^{n-1} D_m | \tau_m \in [\tau_{m\min}, \tau_{m\max}] \right. \\ \left. \tau_{m+1} \in [\tau_{m+1\min}, \tau_{m+1\max}] \right\} \tag{24}$$

Due to the non-smoothness of the cost function given in Eq. (24), convex and gradient optimization methods are not

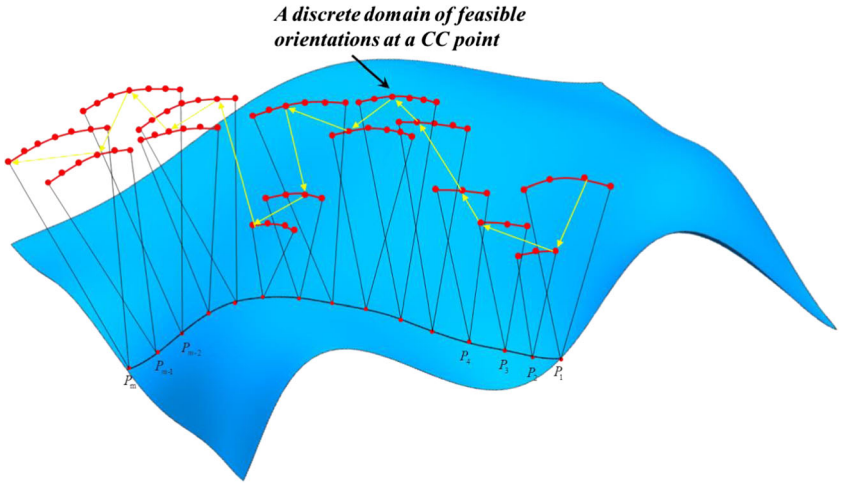

Fig. 10 Discrete Domain of Feasible Orientation (DDFO) model

effective to solve the optimization problem. Besides, the accuracy of heuristic methods [28–30] is determined by discrepancy between the initial guess and the optimal solution, and the optimization may be embogged into local optimal solution. Additionally, expensive computational cost of the aforementioned heuristic algorithms is unacceptable for industrial applications. In order to get rid of the dependence on the initial solution and improve computational efficiency of the optimizing process, the tool orientation sequence smoothing method is proposed based on the Discrete Domain of Feasible Orientation (DDFO). We found it difficult to find the optimal tool orientation sequence in several continuous feasible arcs, but the smoothing problem can be significantly simplified when each feasible arc is discretized. Columns of nodes are shown in Fig. 10 as an example of DDFO model. With feasible arcs discretized into columns of nodes, the selected node from each arc can be combined to express the tool orientation sequence of a tool path.

4.2 Finding the optimal solution in DDFO model

For the sake of initializing the DDFO model while keeping the integrality of it, an end node and a start node are added, as shown in Fig. 11. The optimal solution can be achieved by finding the shortest path from start node to the end node. The optimization based on Dijkstra algorithm [31] begins from the start node which is initialized as zero. One specific node in the next column can be assigned by inheriting values of the edges and nodes associated with it, as shown in Fig. 11. For example, the value of the node $(A,C)_2^2$ is calculated as the minimum of four values $[\left((A,C)_1^1 + D_{11\text{–}22}\right);\ \left((A,C)_1^2 + D_{12\text{–}22}\right);\ \left((A,C)_1^3 + D_{13\text{–}22}\right);\ \left((A,C)_1^4 + D_{14\text{–}22}\right);\ ]$. After assigning all the nodes of DDFO model in the same way, the node assigned with lowest value in the last column is selected and the optimal tool orientation sequence can be acquired by following the selected node to the start node.

In some specific cases, the shortest path contains edges with too large assignments, i.e., singularity problems exist in the solution even though the accumulated rotational movement of the solution is optimal. In order to avoid too large movement and ensure the machining quality, edges with too large assignments shall be excluded. For the sake of eliminating singularity problem of the machining process, a penalization strategy for too large rotational movements is proposed.

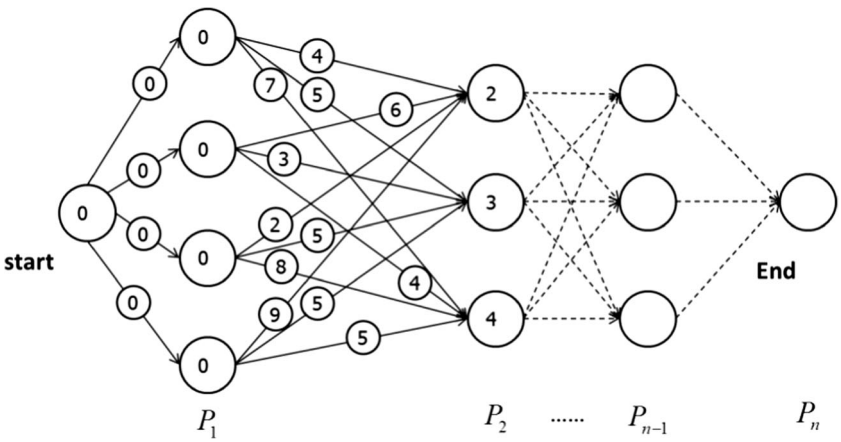

Fig. 11 Finding the optimal solution applying Dijkstra algorithm

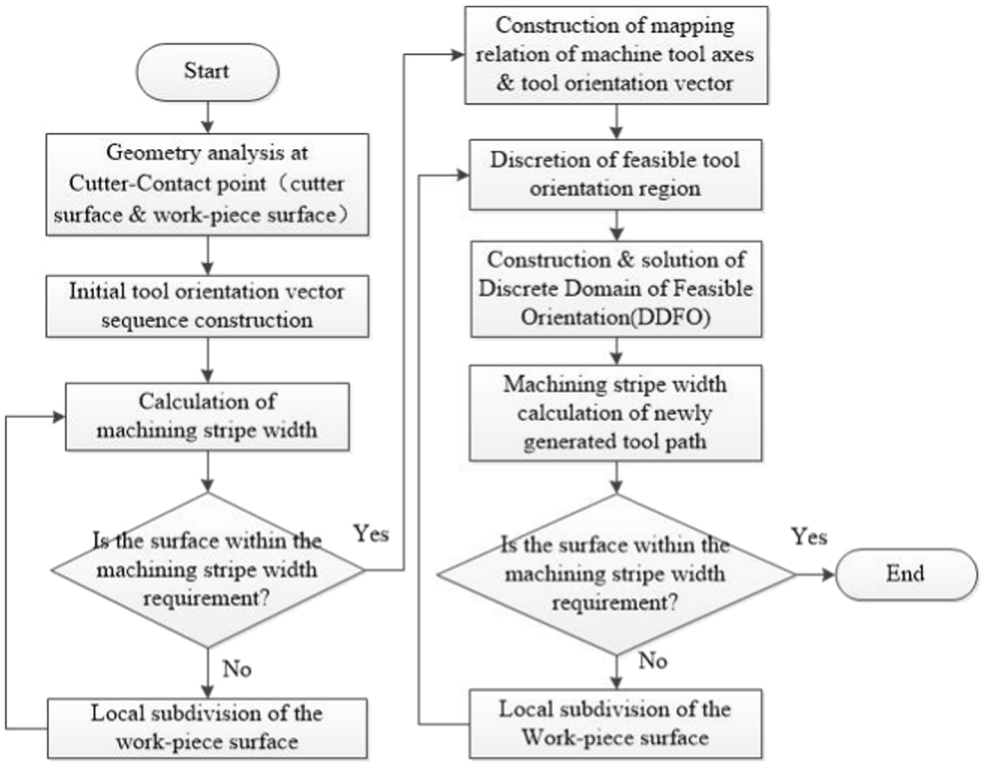

Fig. 12 Flow chart of the whole optimization process

When the posture distance exceeds the maximal rotation limit, the corresponding edge would be weighted a penalization coefficient to reduce the probability of selecting it. Equation (25) expresses the exponential penalty function applied on D_m which could help exclude edges those exceed the maximal change limit, i.e.,

$$E_{penalized} = D_m \cdot \left(\frac{\delta}{\delta_{\max}}\right)^n \tag{25}$$

where δ is the rotation angle between two adjacent tool postures, $\delta_{\max}$ is the penalty threshold, and n is the penalty intensity coefficient.

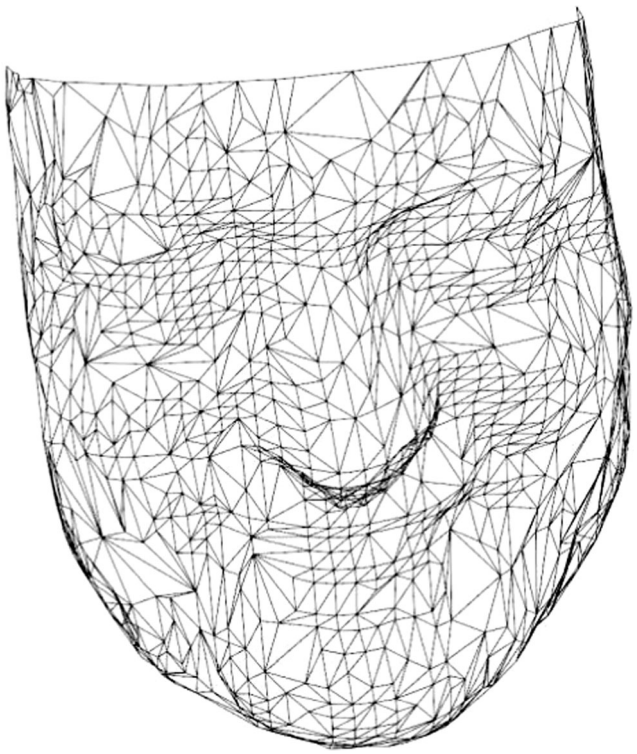

Fig. 13 The face model with triangular meshes

5 The tool path regeneration based on the modified butterfly subdivision

The smoothing method is discussed in Sect. 4, and the tool orientation sequence is optimized by using DDFO model. As the equivalent cutting radius of the fillet-end cutter changes with its posture, the optimized tool orientation sequence may

Table 1 Machining parameters of the simulation experiment

Parameter names	Parameter values
The size of the work-piece	120 mm × 120 mm
The radius of the cutter	$R = 3$ mm
The chamfer radius of the cutter	$r = 1$ mm
Scallop height requirement	$h = 0.2$ mm
Penalty threshold	$\delta_{\max} = 2^\circ$
Penalty coefficient	$n = 2$

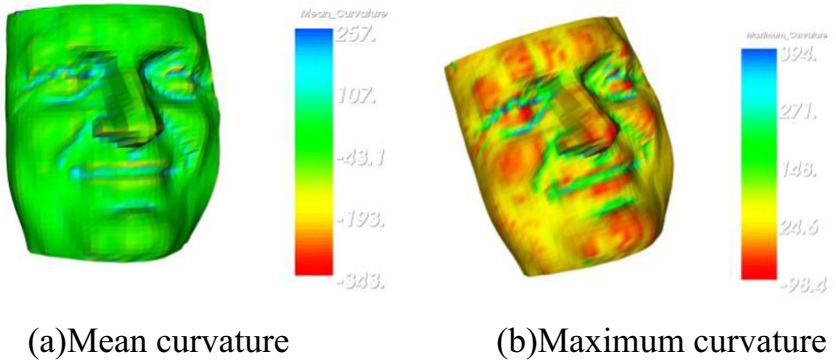

Fig. 14 Curvature analysis of the face model. **a** Mean curvature. **b** Maximum curvature

change the cutting conditions at some CC points. According to Eqs. (5) and (6), there is a negative correlation between the tilt angle β and the cutting stripe width d_{ij}. If the optimized tilt angle β is enlarged to make the tool orientation sequence smooth, the tilt angle may be larger than the initial optimal tilt angle $\beta_{\min}$. Thus, the cutting stripe width diminishes correspondingly at some CC points.

In order to eliminate the diminishment of the cutting stripe width, denser triangular meshes are required to regenerate the tool path. Coincidentally, the local subdivision utilized to generate the initial tool path in Sect. 3.2 could fulfill our requirements. Thus, the local subdivision is iteratively applied to make the meshes denser such that the increasing density of the tool path can eliminate the decreasing stripe width caused by tool orientation optimization. After iterative applications of local subdivision and tool orientation smoothing, the spiral tool path is regenerated according to the method in Sect. 3.3. The aforementioned optimization can be summarized in Fig. 12.

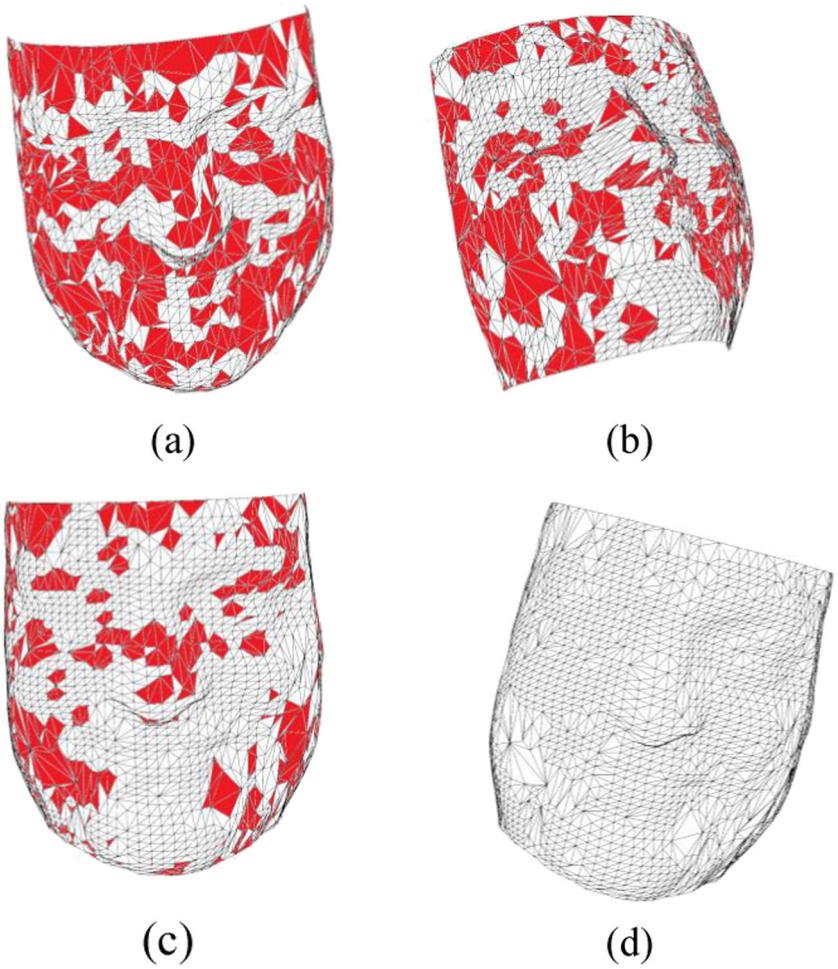

Fig. 15 The meshes contain remnant islets after each subdivision on the face model

6 Simulation and experiment

The proposed method is applied on a face model which contains 1425 vertexes and 2734 triangular meshes. Sharp features exist in eyes-part and nose-part, while smooth features exist in mouth-part and cheek-part. As the aforementioned distribution of features tells, the characteristic of the method regarding maintaining sharp and smooth features can be tested on the face model, as shown in Fig. 13. Our method is applied on the JDVT600 machine tool, and the parameters of the machine tool are listed as follows.

Max rotational speed of the machine tool $N_C = 20r/\min$;

The feed speed usually used along the tool path

$$M_c \in [2500mm/\min, 12000mm/\min];$$

In the first interpolation, the step size of the tool path can be estimated according to the mesh size of the work-piece, i.e., $S = 0.72mm$.

Thus, the maximal tolerable rotation of a single step is $\varphi_{\max} = \frac{S}{M_C \cdot N_C = 2.16^\circ}$. In order to make the calculation process simple and clear, we take the penalty threshold $\delta_{\max} = floor(\varphi_{\max}) = 2^\circ$. The experiment parameters are listed in Table 1.

At first, the unit normal vector at each vertex is estimated according to Eq. (6). The corresponding curvatures, including maximum principal curvature and mean curvature, are analyzed at each vertex of the model, as shown in Fig. 14. The local subdivision eliminates the remnant islets of the triangular meshes to guarantee the whole surface within the scallop height requirement.

The boundary of the surface is detected to be the first row of tool path and the initial tool path is generated after applying Modified Butterfly subdivision three times, and the meshes which still need to be subdivided (with red color) after each subdivision are shown in Fig. 15a–d. In order to obtain the tool orientation sequence data, the CC data of initial tool paths generated by the method of Sect. 3.3 is transformed to CL data according to the coordinate conversion method in Sect. 3.4. The whole surface generated by the initial tool path is within

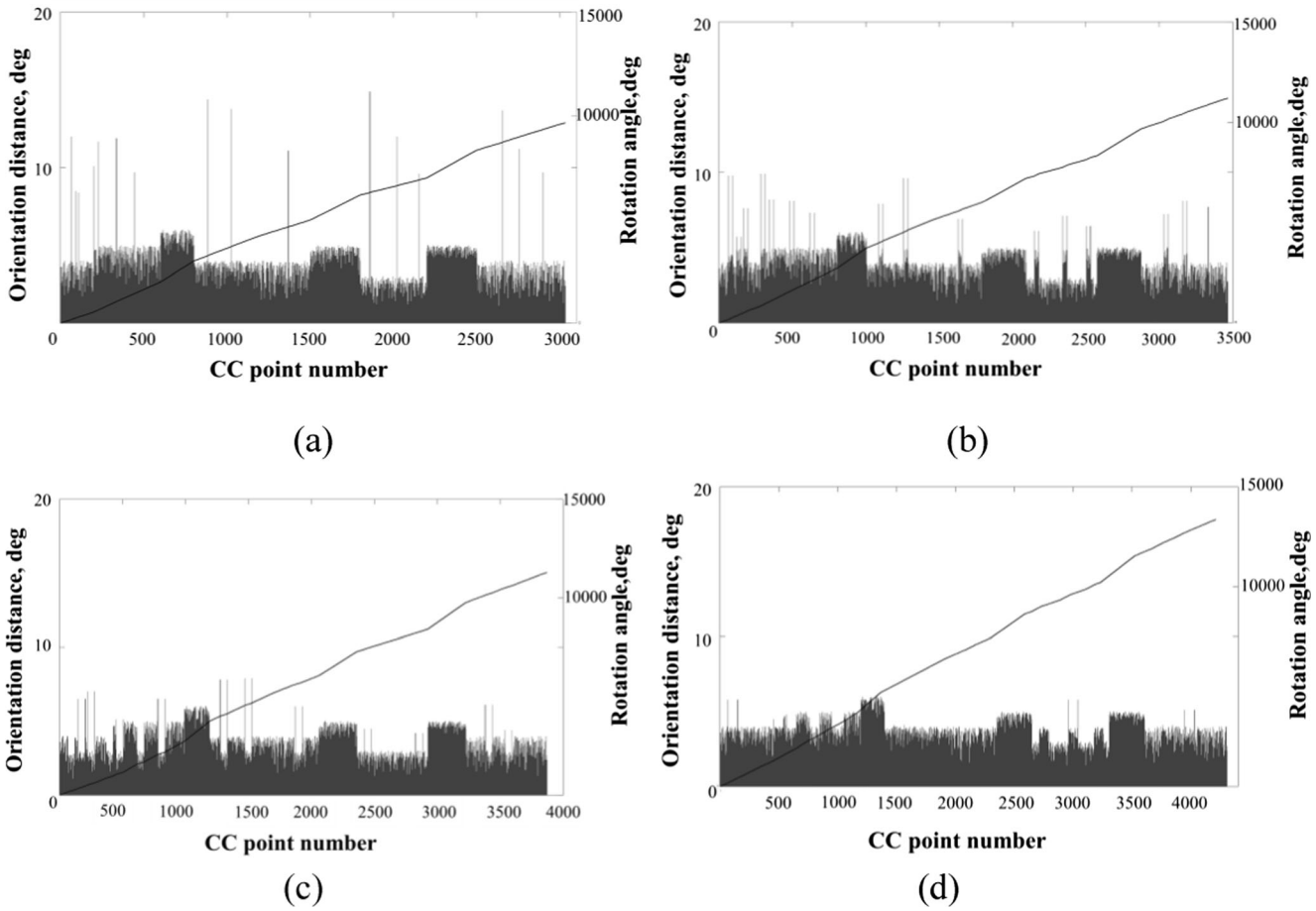

Fig. 16 The tool orientation sequence data (**a** the orientation sequence data of initial path; **b**–**d** the orientation sequence data after the 1st, 2nd, and 3rd application of subdivision)

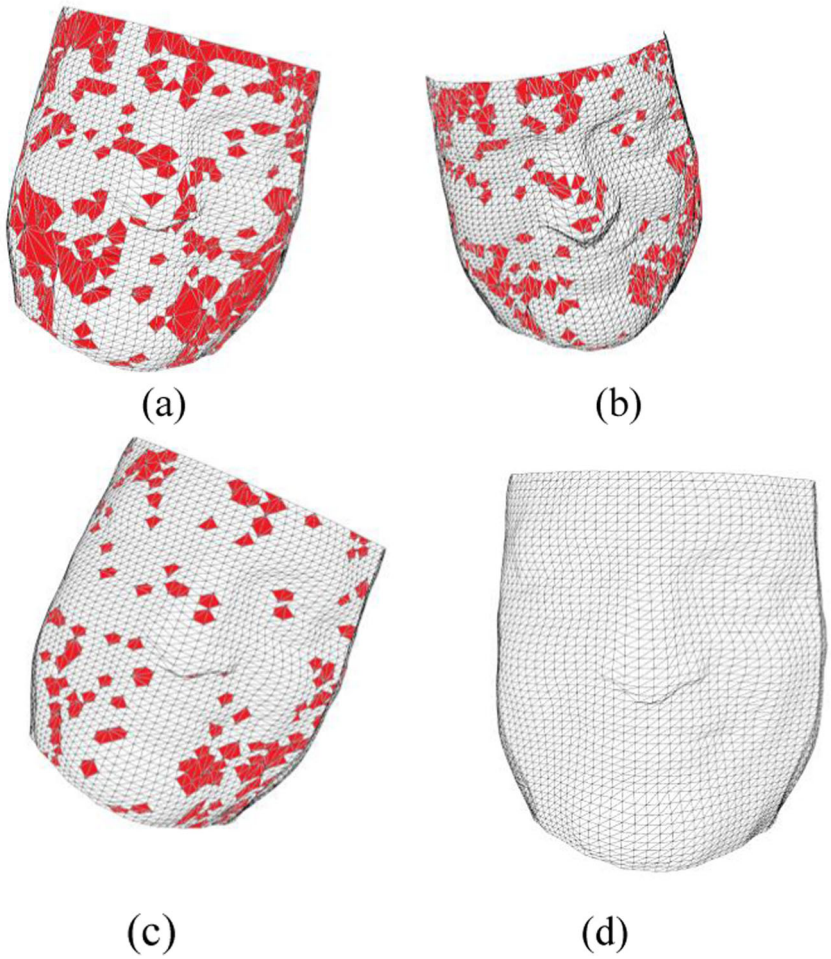

Fig. 17 The meshes contain remnant islets after each iteration of smoothing and re-calculation of the cutting stripe width

the scallop height requirement, but some large tool orientation changes which could cause singularity problems still exist.

The smoothing method in Sect. 4 is applied to eliminate too large tool orientation changes such that the singularity problem is avoided and the machining quality is guaranteed. However, the change of tool orientations at some CC points during the smoothing process can also diminish the cutting stripe width, which means that some triangular

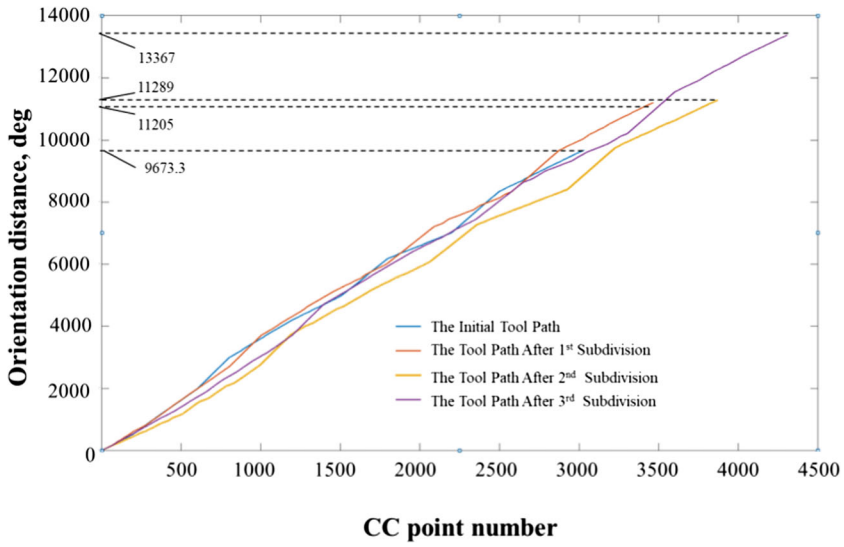

Fig. 18 The comparison of the accumulated rotations after each smoothing

Fig. 19 The smoothed tool path

meshes cannot fulfill the scallop height requirement and some remnant islets will show up as shown in Fig. 4. According to the aforementioned state, the Modified Butterfly subdivision has to be iteratively applied to generate denser meshes such that the interval of the tool path could decrease to make the whole surface within the scallop height requirement. The tool orientation sequence data after each subdivision are shown in Fig. 16, where the bars represent the orientation distance of adjacent CC points and the curve represents the accumulated rotation angle of the machine tool. Correspondingly, the meshes that contain remnant islets (with red color) after each smoothing and re-calculation are shown in Figs. 17a, d.

In order to diminish too large machine tool axis rotations, the meshes with remnant islets are locally subdivided using Modified Butterfly subdivision. In this case, too large rotations disappear but the new-generated meshes need more cutting movements to make the whole surface within the scallop height requirement. In other words, the accumulated rotations are deteriorated to cripple too large rotations, but denser meshes are needed to guarantee the machining quality. The comparison of the accumulated rotations after each time of

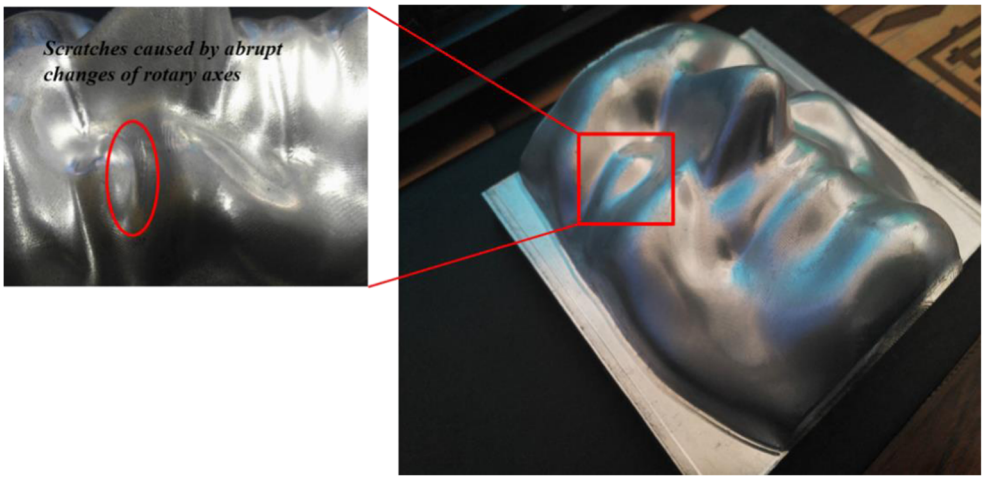

Fig. 20 The machining experiment of tool path generated according to [9]

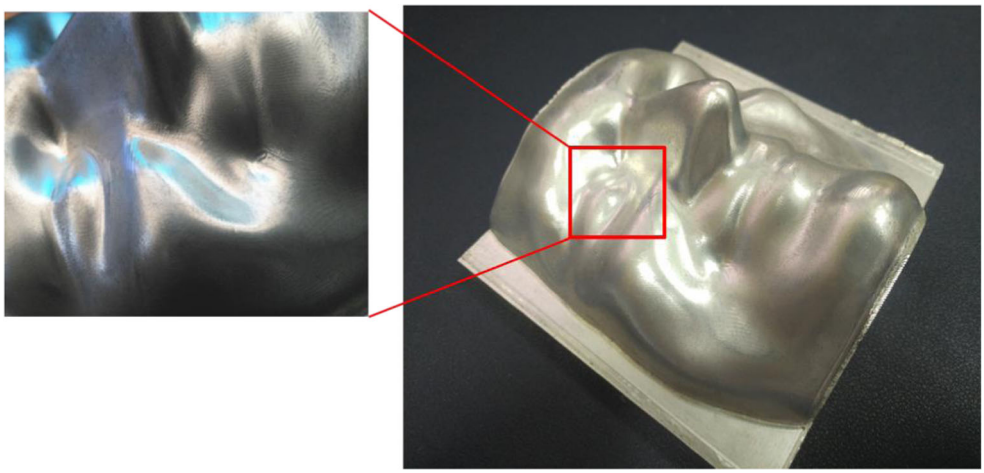

Fig. 21 The machining experiment of tool path generated according to the proposed method

smoothing and subdividing is shown in Fig. 18, indicating that the number of rotary movements increases as interpolated points and new meshes are generated by the Modified Butterfly subdivision. Correspondingly, the accumulated rotations of the machine tool increase by 38.1%, but it is not explosive and can be accepted for the sake of assuring the machining quality of the work-piece. Finally, the smoothed tool path is generated according to the method in Sect. 3.3, as shown in Fig. 19.

To prove the validity of the proposed method, a comparison experiment was conducted on JDVT 600 machine tool. At first, tool paths are generated according to our previous work [9] without considering the kinematic configuration of the machine tool. A ball-end cutter with a radius of 3 mm was used and the tool path runs along the subdivided triangular meshes. Some singularity problems arise at some features whose curvatures have change abruptly. Details of "eye part" on the face model are shown in Fig. 20 where the scratches caused by abrupt changes of rotary axes can be observed. As a contrast, the tool path generated according to the proposed method in this paper is applied on the same face model. Details of the "eye part" on the face model can be shown in Fig. 21, where it could be found that the scratches are eliminated. Due to smoothing of the tool orientation sequence considering the machine tool's kinematic configuration, abrupt changes of rotary axes are eliminated and scratches are removed along with them.

7 Conclusions

This paper proposes a method to generate a smooth tool path on a triangular-mesh surface. At first, the geometry features of the cutter and the work-piece at the CC point are analyzed. Then, the Modified Butterfly subdivision is applied to generate the initial tool path and the tool orientation sequence is established. However, there are still too large tool orientation changes in the tool orientation sequence. Realizing that, the DDFO model is constructed and the optimization method based on Dijkstra algorithm is employed to smoothen the tool orientation sequence. Finally, the Modified Butterfly subdivision is iteratively applied to eliminate the cutting stripe width diminishment problem caused by the smoothing of tool orientations. To prove the validity of the proposed method, a simulation and comparison experiments are given. The experiment results show that the smoothing method can eliminate the singularity problem of the initial path in the premise of fulfilling the scallop height requirement.

Note that smoothing the tool orientation is inevitable to make the tool path applicable for machining while eliminating the singularity problems. Different machine tools have different configurations, and the mapping and optimizing equations are also different. Therefore, generalizing the proposed method to different types of machine tools deserves more attention.

Funding This work was supported by the National Natural Science Foundation of China (Nos. 51675477,51775489), Zhejiang Provincial Natural Science Foundation of China (No. LZ18E050001), and Innovation Foundation of the State Key Laboratory of Fluid Power and Mechatronic Systems.

References

1. Kim HC (2010) Tool path generation for contour parallel milling with incomplete mesh model [J]. Int J Adv Manuf Technol 48(5): 443–454
2. Chuang CM, Yau HT (2005) A new approach to z -level contour machining of triangulated surface models using fillet-end mills [J]. Comput Aided Des 37(10):1039–1051
3. Wei ZC, Wang MJ, Zhu JN, Gu LY (2011) Cutting force prediction in ball end milling of sculptured surface with Z -level contouring tool path [J]. Int J Mach Tools Manuf 51(5):428–432
4. Lee E (2003) Contour offset approach to spiral tool path generation with constant scallop height [J]. Comput Aided Des 35(6):511–518
5. Sun Y, Xu J, Jin C, Guo D (2016) Smooth tool path generation for 5-axis machining of triangular mesh surface with nonzero genus[J]. Comput Aided Des 79:60–74
6. Balasubramaniam M, Ho S, Sarma S, Adachi Y (2002) Generation of collision-free 5-axis tool paths using a haptic surface[J]. Comput Aided Des 34(4):267–279
7. Bo Z (2013) Processing calculation method of 5-axis NC spiral trajectory based on parametric surface mapping [J]. Chin J Mech Eng 49(5):100
8. Chen T, Gang Z (2015) Tool path generation for loop subdivision surface based finish machining [J]. J Beijing Univ Aeronaut Astronaut 41(4):663–668
9. Zhang Z, Feng Y, Ren B, Hagiwara I (2016) Exploratory study of spiral NC tool path generation on triangular mesh based on local subdivision [J]. Int J Adv Manuf Technol 83(5):835–845
10. Kim T (2002) Sarma SE tool path generation along directions of maximum kinematic performance: a first cut at machine-optimal paths. Comput Aided Des 34(6):453–468
11. Castagnetti C, Duc E, Ray P (2008) The domain of admissible orientation concept: a new method for five-axis tool path optimisation [J]. Comput Aided Des 40(9):938–950
12. Lavernhe S, Tournier C, Lartigue C (2008) Optimization of 5-axishigh-speed machining using a surface based approach. Comput Aided Des 40(10–11):1015–1023
13. Jun CS, Cha K, Lee YS (2003) Optimizing tool orientations for 5-axis machining by configuration-space search method[J]. Comput Aided Des 35(6):549–566
14. Barakchi Fard M, Feng HY (2009) Effect of tool tilt angle on machining stripe width in five-axis flat-end milling of free-formsurfaces. Int J Adv Manuf Technol 44:211–222
15. Li LL, Zhang YF, Li HY et al (2010) Generating tool-path with smooth posture change for five-axis sculptured surface machining based on cutter's accessibility map [J]. Int J Adv Manuf Technol 53(5):699–709
16. Ho MC, Hwang YR, Hu CH (2003) Five-axis tool orientation smoothing using quaternion interpolation algorithm [J]. Int J Mach Tool Manu 43(12):1259–1267

17. Hsieh HT, Chu CH (2013) Improving optimization of tool path planning in 5-axis flank milling using advanced PSO algorithms [J]. Robot Comput Integr Manuf 29(3):3–11
18. Rao N, Ismail F, Bedi S (1997) Tool path planning for five-axis machining using the principal axis method[J]. Int J Mach Tools Manuf 37(7):1025–1040
19. Meyer M, Desbrun M, Schröder P et al (2002) Discrete differential-geometry operators for triangulated 2-manifolds[J]. Visual Math 3(8–9):35–57
20. Zou Q, Zhang J, Deng B, Zhao J (2014) Iso-level tool path planning for free-form surfaces [J]. Comput Aided Des 53(5):117–125
21. You CF, Sheen BT, Lin TK (2001) Robust spiral tool-path generation for arbitrary pockets[J]. Int J Adv Manuf Technol 17(3):181–188
22. Deng C, Ma W (2012) Weighted progressive interpolation of loop subdivision surfaces [J]. Comput Aided Des 44(5):424–431
23. Loop C, Schaefer S (2008) Approximating Catmull-Clark subdivision surfaces with bicubic patches [J]. ACM Trans Graph 27(1):1–11
24. Novara P, Romani L, Yoon J (2016) Improving smoothness and accuracy of modified butterfly subdivision scheme [J]. Appl Math Comput 272(P1):64–79
25. Xu Z, Kondo K (2002) Local subdivision process with doo-Sabin subdivision surfaces[C]. Shape Model Int IEEE Comput Soc 7
26. Plakhotnik D, Lauwers B (2014) Graph-based optimization of five-axis machine tool movements by varying tool orientation[J]. Int J Adv Manuf Technol 74(1–4):307–318
27. Xu K, Tang K (2017) Optimal workpiece setup for time-efficient and energy-saving five-Axis machining of freeform surfaces[J]. J Manuf Sci Eng 139(5):051003
28. Yusup N, Zain AM, Hashim SZM (2012) Evolutionary techniques in optimizing machining parameters: review and recent applications (2007–2011) [J]. Expert Syst Appl 39(10):9909–9927
29. Yildiz AR (2013) Cuckoo search algorithm for the selection of optimal machining parameters in milling operations [J]. Int J Adv Manuf Technol 64(1–4):55–61
30. Yildiz AR (2013) A new hybrid differential evolution algorithm for the selection of optimal machining parameters in milling operations [J]. Appl Soft Comput 13(3):1561–1566
31. Massart DL, Dijsktra A, Kaufman L (1978) Evaluation and optimization of Labortary methods and analytical procedures [J]. Tech Instrum Anal Chem 1