Contour-parallel tool path generation for three-axis mesh surface machining based on one-step inverse forming

Jinting Xu, Yujin Wang, Xiangkui Zhang and Shuyu Chang

Abstract
Recently, mesh surfaces have become the focus of considerable interest due to its simplicity for data exchange and geometric computation. However, for mesh surface machining, there are few tool path planning strategies but iso-planar method. In this article, a contour-parallel tool path method is presented for machining complicated mesh surfaces with holes or islands by introducing a so-called one-step inverse approach based on physical plastic deformation of metal materials. By rapidly simulating for the inverse forming process of sheet metal of auto-body panels, the machined surfaces are flattened into its initial flat blank on which the cutter contact points are calculated by iteratively offsetting the inner and outer contours of the blank. Then, using the mapping from the machined surfaces to the blank as a guide, the planar offset curves without self-intersection are inversely mapped onto the original mesh surfaces, forming the final contour-parallel tool paths. Benefiting the flattening method based on one-step inverse approach, the task of generating tool paths is reduced from a physical surface to a plane so that geometric computation of tool path generation is greatly simplified, especially for self-intersection elimination in offset curves. Moreover, a new data structure of offset curve is also introduced, which is used to effectively compute the self-intersections. The proposed method has been tested on several sample surfaces, and the experimental results show that the proposed method can deal with complicated mesh surfaces with holes or islands and is also more efficient compared with the traditional methods.

Keywords
Computer-aided manufacturing, numerical control machining, tool path generation, mesh surface, one-step inverse forming

Introduction
At present, triangular mesh surfaces have gained nearly universal acceptance and are supported by most of current computer-aided design (CAD)/computer-aided manufacturing (CAM) systems, in which complex mesh surfaces can be created directly from design surface or solid model and are stored as stereo lithography (STL) format that has been a de facto standard. Compared with parametric surfaces, in STL file, only the position of every vertex and the normal of every facet are recorded, so that mesh model can be handily translated between different CAD systems, and its related geometric computation is simpler and more robust than that of parametric surfaces due to the linear representation for design model. Recently, mesh surfaces have become the focus of considerable interest and been used as representation model for tool path planning. In most studies, mesh surface machining was often to be as a means to generate gouge- and collision-free tool path for machining free-form surface, and only one topology of iso-planar type is used. Although iso-planar methods are simple and robust, excessive short paths resulted from irregular or curved boundary of the surface make them very defective. Recently, several different planning strategies have been developed. Kim and Yang proposed a cutter location (CL) surface deformation–based approach that first deforms CL surface to a new surface and then inversely maps the curves generated by slicing the CL surface into iso-scallop height tool paths. Lee et al. proposed a cutter location (CL) surface deformation–based approach that first deforms CL surface to a new surface and then inversely maps the curves generated by slicing the CL surface into iso-scallop height tool paths. Lee et al.
extended further the concepts of the conventional guide plane and the drive surface and utilized a series of plane driven by side steps to slice the CL surface to generate iso-scallop tool paths. Makhanov et al.\textsuperscript{15,16} adopted grid generation technique to the tool path optimization and generated spiral tool path embedded into zigzag tool path on complex surface. Based on conformal maps, Sun et al.\textsuperscript{17} proposed a spiral cutting strategy that realized cutting of mesh surfaces without tool retractions. These latest methods have explored tool path generation further for meshes machining from various perspectives.

However, there is a limited number of studies that have been performed on contour-parallel machining of mesh surfaces. Contour-parallel tool paths are widely used for sculptured surfaces or free-form surface machining\textsuperscript{18} as it reduces the nonproduction time and also maintains the consistent up-mill or down-mill material removal during the cutting process.\textsuperscript{19} In contour-parallel tool paths, a crucial issue is to eliminate self-intersection interference arising out of contour offsetting. Since offset curves are often represented by the discrete points and their connecting piecewise linear line segments, this causes some line segments nearly meet but do not really intersect, and thus, eliminating self-intersections directly on the surfaces is not usually desirable. For free-form surface modeled by parametric domain. To deal with this issue, Li et al.\textsuperscript{21} have done some valuable works. They flattened the machined surfaces onto a plane by angle-based flattening (ABF) technique used widely in texture mapping.\textsuperscript{22} On the plane, the self-intersection is removed easily using 2D curve offsetting algorithm. Their method works well for simple mesh surfaces, but not suitable for the surfaces with holes or islands often encountered in practice because their flattening method is only applicable to simple surfaces without holes.

In view of convenience and efficiency of generating the contour-parallel tool paths on complicated mesh surfaces with holes, this article adopts Li’s tool path principle and extends its ability to process the mesh surfaces with holes. A new flattening method is introduced in tool path generation by a so-called one-step inverse approach (IA) simulating the metal sheet inverse forming process. Moreover, an approach based on mesh topology to calculating the self-intersection points is also given to improve 2D curve offsetting algorithms. The rest of this article is organized as follows: Basic theory of flattening algorithm based on one-step IA is introduced in section “Flattening method based on one-step IA,” and section “Tool path generation” discusses the technical details of the developed method. Experimental results are presented in section “Experimental results,” and section “Conclusion and future work” gives the concluding mark and future work.

**Flattening method based on one-step IA**

The so-called one-step inverse approach (IA) is described as follows: As shown in Figure 1, given the mesh model of the final workpiece, the material property, thickness of the initial blank and the boundary conditions (blank-holder force (BHF), draw-bead restraining force (DBRF), friction force and constraints), requirement is to predict positions of the nodal in the initial flat blank and the distributions of the thickness h, strain ε, stress σ and so on in one step in the final workpiece.

Before this, one-step IA is only used in sheet forming to predict the optimum blank shape and sizes and reasonable estimation of forming severity (i.e. strain distributions and thickness) from desired final workpiece. It is the first time to be used in numerical control (NC) machining. Previous work\textsuperscript{23} of Li et al. has demonstrated that one-step IA can obtain the flattened meshes with lower area distortion and deal with irregular meshes with holes very well compared with other mesh parameterization method such as ABF+++, and it can replace ABF to generate the tool paths on more complicated mesh surface. For integrity of our method, the basic theory one-step IA is reviewed as follows and the details of the mathematical description can be referred to our team’s works.\textsuperscript{23,24}

According to the principle of virtual work, the equation of energy of final workpiece is described as

$$W = \sum_{e} \left( \int_{\gamma_e} \langle \varepsilon' \rangle \{\sigma\} dV - \int_{\gamma_e} \langle \sigma' \rangle \{f\} dV \right) = 0 \tag{1}$$

where,} \varepsilon' and} \{u'\} are the virtual strain and virtual displacement,} \{\sigma\} is the Cauchy stress and} \{f\} is the equivalent nodal force which is caused by the force between die and roughcast interaction and the friction resistance. For arbitrary elasto-plastic deformable element in equilibrium state, the principle of virtual work must be satisfied as the sufficient and necessary condition for the stress field. Thus, in the material coordinate system of the final mesh surface, the energy equation of element e can be described as

$$\int_{\gamma_e} \delta(\varepsilon') \{\sigma'\} dv - \int_{\gamma_e} \delta(\sigma') \{f_{ext}\} dv = 0 \tag{2}$$
where $\delta(u^c)$ and $\delta(u^o)$ are virtual strains and virtual displacements, respectively, $\{\sigma^e\}$ is the Cauchy stresses and $\{f_{ext}^e\}$ is the equivalent external forces, which is expressed as

$$\{f_{ext}^e\} = \{f^e_o\} + \{f^e_i\} + \{f^e_S\}$$

where $\{f^e_o\}$, $\{f^e_i\}$ and $\{f^e_S\}$ are friction force, BHF and draw-bead force, respectively, whose definition can be referred to the study by Zhou et al.\(^{24}\) According to equations (1) and (2), the following nonlinear equilibrium system can be obtained

$$\{R(U)\} = \sum_{i=1}^{n} (\{F_{int}^i\} - \{F_{ext}^i\}) = 0 \quad (3)$$

where $F_{int}^i$ is the global internal force vector of element $e_i$. Since one-step IA is a strong nonlinear problem induced by large displacement, large strain and material nonlinearity, Newton–Raphson (N-R) method is an efficient tool for the solution of this problem. The purpose of N-R solution is to achieve the equilibrium in the final workpiece, and the convergence criterion is given by

$$\|R_{i}(u^c) - R_{i}(u^{c-1})\|_n \leq \epsilon \quad (4)$$

where $R_{i}$ is the residual force vector in nodal tangent plane and $\epsilon$ is a given iteration precision.

### Tool path generation

Using the above flattening method, the machined surfaces are nicely flattened onto the plane. On this plane, the inner and outer boundaries of the blank are selected as the initial drive curves for curve offsetting. The developed tool path method adopts pair-wise offset (PWO) algorithm\(^{25-28}\) to generate the planar guide curves and improves further its performance using topology relations of mesh.

### Self-intersection point calculation

To calculate self-intersections efficiently, traditional offsetting methods often employ a grid structure over the initial offset curves to record the bounding bins that enclose each segment. Simultaneously, for each bin, the segments spanning over the bin are also recorded. Using a simple test for overlapping bin, a great number of attempted intersections can be avoided.\(^{19,26}\) For mesh surfaces, the above grid structure is unnecessary and can be replaced by the planar mesh surfaces itself. In the present method, when computing the initial offset curves using PWO algorithm, the offset point and its corresponding triangle, which contains the point, are stored as the following data structure $P_i$

$$P_i = \{p^e_o, \Delta^p\} \quad (5)$$

where $p^e_o$ is the offset point and $\Delta^p$ is the triangle enclosing $p^e_o$. Storing $\Delta^p$ is not only for calculating points of self-intersection efficiently, but also for mapping conveniently the offset curves onto the machined surfaces in the following process. For an offset segment $S_{i,j+1} = \{P_i, P_{i+1}\}$, the triangles $\{\Delta^p\}$, which are spanned over by $S_{i,j+1}$, can be rapidly determined using the topological relations of mesh and are stored as

$$S_{i,j+1} = \{\{p^e_o, p^e_o + 1\}, \{\Delta^p\}\} \quad (6)$$

In this judging process, for each triangle $\Delta^p_j$, the segments $\{S_{i,j+1}\}$ spanning over $\Delta^p_j$ are also recorded and are stored by the following data structure $\Delta^p_j$

$$\Delta^p_j = \{\Delta^p_j, \{S_{i,j+1}\}\} \quad (7)$$

Using the above data structures (equations (6) and (7)), overlapping triangles $\{\Delta^p\}$ of different offset segments can be rapidly identified and then the self-intersection points in the initial offset curves are calculated among $\{S_{i,j+1}\}$ in each triangle of $\{\Delta^p\}$, thus avoiding the attempt to intersect all segments with all other segments. Since the topology relations of the meshes have been already created while reading the mesh model from the STL files, the procedure of traditional grid partition can be removed from PWO algorithm so that the performance of the algorithm will be further improved.

### Offsetting strategy for boundaries

In the developed tool path approach, the offset distance or the path interval is determined according to the theorem in the study by Huang and Oliver\(^{29}\) and changes with the surface curvature. As mentioned earlier, the inner and outer boundaries of the planar blank are selected as the initial drive curves and the offset curves are generated by PWO algorithm. Note that in the present method, for the convenience of processing, the inner boundaries are offset outward only once and other remaining machined region is implemented by inward offsetting the outer boundaries, as shown in Figure 2. The local and global self-intersection points in the initial offsets are calculated by the method of section

![Figure 2. First offset of outer and inner boundaries.](image)
“Self-intersection point calculation” and are removed automatically by PWO algorithm.

The intra-intersections between the offsets of the outer and inner boundaries and the intra-intersections of the offsets of the inner boundaries themselves are also simultaneously detected using the method of section “Self-intersection point calculation.” If there are the intra-intersections among the offsets of the inner boundaries, the offsets will be merged into a united offset curve after eliminating the invalid loops. As shown in Figure 2, the offsets of left two islands, \( G_1 \) and \( G_2 \), intersect, forming two intra-intersections at which \( G_1 \) and \( G_2 \) are split into two chains, the invalid loop in red and the valid chain. The invalid loop is eliminated by the orientation criterion, namely, retain the loops with the same orientation as the original contour and reject the others, and these two valid chains are joined into a new offset curve \( \{ G_1, G_2 \}_\text{valid} \) of left two islands. Likewise, between the offsets of the right island \( G_3 \) and the offset of the outer boundary \( S_1 \), there are two intra-intersections, as shown in Figure 2. Different from processing of \( G_1 \) and \( G_2 \), the valid chains of \( G_3 \) and \( S_1 \) are united into new outer offsets \( \{ G_3, S_1 \}_\text{valid} \) which are then used as starting curve for next offsetting. The above procedures of offsetting curves, eliminating self-intersections and merging offset curves are repeated until offset curves cover entirely the planar mesh surfaces.

**Generating tool paths**

Once the planar guide curves are generated, they can be inversely mapped onto the machined surfaces and generate the tool paths for machining mesh surfaces. According to the triangle \( \Delta^j_0 \) containing the offset point \( p^j_0 \), recorded in data structure \( \{ p^j_0, \Delta^j_0 \} \), the corresponding triangle \( \Delta^m_0 \) on the machined mesh surfaces can be obtained easily. Using the mapping of \( \Delta^j_0 \) to \( \Delta^m_0 \), \( f: \Delta^j_0 \rightarrow \Delta^m_0 \), the corresponding point \( p^m_0 \) of \( p^j_0 \) on the machined surfaces can be obtained by the following equation

\[
f: \left( \Delta^j_0 \rightarrow \Delta^m_0 \right) p^j_0 = p^m_0
\]  

When mapping the guide curves, the step length has to be controlled such that the machining requirements can be satisfied. For mesh surfaces, in most CAD/CAM systems, the approximation error of the mesh surface to the original surface can be specified by the users such that a desirable triangulation to be adequate for NC machining is obtained conveniently, so only offset points and the intersections of the offsets and the edges are selected as cutter contact (CC) points. If the approximation error of mesh surface is equal to or less than an allowable chordal error of the step length, the step length of the generated tool path can be controlled reasonably. Then, the CC points \( \{ p^o_m \} \) are converted into the CL points \( \{ p^o_m \}_\text{CL} \), and after removing the gouge interfering in the tool path using the method in the studies by Chen and Shi\(^5\) and Kiswanto et al.,\(^1\) CL points are interpolated by straight-line segments or a B-spline curve that is then post-processed into \( G \) code and is used for NC machining of mesh surfaces.

**Experimental results**

**Implementation**

All the algorithms mentioned above has been coded in C++ and implemented on a personal computer (Intel 3.0 GHz CPU, 4.0 GB physical memory). Geometric data of the surfaces are first retrieved from CAD model in Unigraphics 7.5 by selecting surfaces from the workpiece. The selected surfaces are triangulated into the mesh surfaces using the triangulation function provided by UG systems. Next, the mesh surfaces are flattened onto the plane using the method given in section “Flattening method based on one-step IA.” On this planar mesh surfaces, the proposed tool path method in section “Tool path generation” is used to generate contour-parallel tool paths for machining of mesh surfaces. The machining parameters used in experiment are given as follows:

Cutter: ball-end tool with diameter of \( \Phi \) 10 mm; Scallop height: 0.5 mm.

**Comparison of one-step IA and ABF++**

Two types of distortion, angular distortion and area distortion, usually used for evaluating the flattening algorithm in the literature, are measured. Angular distortion is defined as

\[
D_{\text{ang}} = \frac{E(\alpha)}{3n_t}, \quad E(\alpha) = \sum_{t \in T} \sum_{k=1}^{3} \frac{1}{\omega_k} (\alpha^t_k - \beta^t_k)^2
\]  

where \( n_t \) is the number of triangle, \( \alpha^t_k \) are the unknown planar angles, \( \beta^t_k \) are the optimal angles, the index \( t \) represents the triangle in the mesh model and \( k \) stands for the angles in each triangle. The weights \( \omega_k \) are set as \( 1/\beta_k^2 \).\(^23\) And area distortion is measured with area root mean square (RMS). Table 1 gives the distortion statistics of the flattening results of one-step IA and ABF++ on the model, as shown in Figure 3. It can be seen that one-step IA minimizes the angle distortion like ABF++ and has lower area distortion. Figure 3(b) and (c) shows the flattening meshes of one-step IA and ABF++, respectively. It is noted that one-step IA gives better shape of the flattened boundary that might be

<table>
<thead>
<tr>
<th>Models</th>
<th>Methods</th>
<th>Angle (RMS)</th>
<th>Area (RMS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 3(a)</td>
<td>One-step IA</td>
<td>0.00286324</td>
<td>0.035874</td>
</tr>
<tr>
<td>#v: 264; #f: 471</td>
<td>ABF++</td>
<td>0.00214762</td>
<td>0.159237</td>
</tr>
</tbody>
</table>

RMS: root mean square; IA: inverse approach; ABF: angle-based flattening.
preferable for contour offsetting because poor boundary curve will lead to more self-intersections in initial offsets than smooth boundary, as shown Figure 3(b).

Performance of improved pair-wise offset algorithm

In proposed method, traditional pair-wise offset (TPWO) algorithms are used to generate the initial offset curves. The self-intersections are computed by the method of section “Self-intersection point calculation” instead of using the traditional grid partition and are then eliminated by the orientation criterion. Since the procedure of the traditional grid or enclosing box partition is removed from the PWO algorithm, it is expected that the efficiency of the PWO algorithm will be further improved. Table 2 compares the execution times of the TPWO algorithm with improved pair-wise offset (IPWO) algorithm on two mesh models, as shown in Figure 4. The number of vertex and triangle of the two mesh models and their sizes are given in Table 2, respectively. It can be seen that IPWO reduces 18.3% and 21.6% execution times compared with TPWO. About 20% execution time can be saved by IPWO instead of using TPWO.

Table 2. Execution times (ms) for two models in Figure 4.

<table>
<thead>
<tr>
<th>Models</th>
<th>IPWO</th>
<th>TPWO</th>
<th>Savings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 4(a); #v: 1239; #f: 2372; size: 113 \times 106 \times 19 mm</td>
<td>103</td>
<td>126</td>
<td>18.3%</td>
</tr>
<tr>
<td>Figure 4(b); #v: 3750; #f: 7068; size: 80 \times 66 \times 200 mm</td>
<td>294</td>
<td>375</td>
<td>21.6%</td>
</tr>
</tbody>
</table>

IPWO: improved pair-wise offset; TPWO: traditional pair-wise offset.

Analysis and comparison with other methods

Benefiting from the mechanical background of one-step IA pointed out in the study by Li et al., the proposed tool path method has no excessive restrictions on quality of the triangle meshes and can process more complex meshes in industry. These practical mesh models usually have invalid topology such as holes or ill-shaped triangles and are quite different from the scanning repositories used by the academic research in texture mapping of computer graphics (CGs). Therefore, compared to the tool path method based on the flattening algorithms of CG, the developed method is relatively robust for the irregular mesh surfaces. Moreover, the new data structures of offset curve introduced by this article take full advantage of the topology relation of the mesh surface to compute the self-intersections in
the initial offset curves, thus avoiding the procedure of the traditional grid partition, and experimental results show that this new data structure can effectively identify and compute the self-intersections and save about 20% execution times compared with those TPWO algorithms.25–27

Since the proposed tool path method is to map the offset curves on the planar meshes onto the machined surfaces to generate three-dimensional (3D) contour-parallel tool paths, it inherits the advantages of two-dimensional (2D) offset path of pocketing, and compared with iso-planar tool path of one-way and zigzag type,5,30 the generated tool paths have shorter nonproduction paths and consistent up-mill or down-mill stock material removal in cutting process and can conform nicely to the boundary of the surfaces, thus avoiding producing excessive short tool paths. However, the scallop height on whole surface is not constant compared with the iso-scallop tool paths31 that meet the same cutting requirements such that the proposed method usually generates a longer tool path, but the maximum scallop on the surfaces does not exceed the limit of the user-specified scallop height.

Moreover, the basic theory of the proposed method can be used to eliminate the self-intersection arising out of iso-scallop tool path generation because iso-scallop path curve generation is also to offset geometrically the boundaries of the surfaces or curves on the surfaces by
the path intervals on each CC points.\textsuperscript{18} Similar to the process of contour-parallel tool path generation, the discrete representation of iso-scallop path curve causes that it is difficult to eliminate the self-intersections directly on the surfaces; thus, the raw iso-scallop path curves can first be mapped onto the planar surfaces on which the self-intersections are then removed by 2D offsetting algorithm. Undoubtedly, this can simplify the iso-scallop tool path generation.

Conclusion and future work

A contour-parallel tool path method is developed for mesh surface machining. The machined surface is mapped onto the planar domain, which reduces the task of eliminating complex self-intersection interfering from physical surface to 2D plane. Undoubtedly, this simplifies considerably the processing of self-intersection interference. The flattening method based on one-step IA is introduced, which is used in the proposed method to address those complicated mesh surfaces with holes in industry. Experimental results show that one-step IA can effectively flatten the mesh surfaces with holes and gives better shape of the flattened boundary. Moreover, new data structure of offset curve can avoid the grid partition procedure in the TPWO algorithms and save about 20\% execution times, which has been confirmed by our experiments. It is noted that present method is only applicable to three-axis machining of mesh surface. For five-axis machining, the orientation of the cutter must be considered for generating gouge- and collision-free tool path, which is our work to do in the future research.

Declaration of conflicting interests

The authors declare that there is no conflict of interest.

Funding

This work was supported by the National Natural Science Foundation of China (grant nos 51105058 and 11102035).

References


