An investigation on adaptive sculpting for the thin-walled surface parts using on-machine measurement

Shengtao Lin and Zhengcai Zhao
Nanjing University of Aeronautics and Astronautics, Nanjing, China

Abstract

Purpose – Complex and exquisite patterns are sculpted on the surface to beautify the parts. Due to the thin-walled nature, the blank of the part is often deformed by the forming and clamping processes, disabling the nominal numerical control (NC) sculpting programs. To address this problem, a fast adaptive sculpting method of the complex surface is proposed.

Design/methodology/approach – The geometry of the blank surface is measured using on-machine measurement (OMM). The real blank surface is reconstructed using the non-uniform rational basis spline (NURBS) method. The angle-based flattening (ABF) algorithm is used to flatten the reconstructed blank surface. The dense points are extracted from the pattern on the image using the OpenCV library. Then, the dense points are quickly located on the complex surfaces to generate the tool paths.

Findings – By flattening the reconstructed surface and creating the mapping between the contour points and the planar mesh triangular patches, the tool paths can be regenerated to keep the contour of the pattern on the deformed thin-walled surface.

Originality/value – The proposed method can adjust the tool paths according to the deformation of the thin-walled part. The consistency of sculpting patterns is improved.

Keywords – On-machine measurement, Angle-based flattening, Complex surface, Adaptive sculpting

Paper type – Research paper

Nomenclature

NC Numerical control
OMM On-machine measurement
ABF Angle-based flattening
d_{i,j} Control points of NURBS surface
ω Weight factor
U, V Node vectors of NURBS surface
P(u,v) Function of NURBS surface
N_{i,k} B spline function
α Angle vector of the flat triangular mesh
φ Optimal angle vector for α
w Weight factor for α

© Shengtao Lin and Zhengcai Zhao. Published in Journal of Intelligent Manufacturing and Special Equipment. Published by Emerald Publishing Limited. This article is published under the Creative Commons Attribution (CC BY 4.0) licence. Anyone may reproduce, distribute, translate and create derivative works of this article (for both commercial and noncommercial purposes), subject to full attribution to the original publication and authors. The full terms of this licence may be seen at http://creativecommons.org/licences/by/4.0/legalcode

This work is partially supported by the National Natural Science Foundation of China (No. 51475233 and No. 51805258) and Fundamental Research Funds for Central Universities (NZ2016107).
1. Introduction
For the advantages of lightweight, compactness and high strength-to-weight ratio, thin-walled parts are widely applied in different fields, such as automotive, aerospace and jewelry. Complex patterns are sculpted on the surface of the part to beautify the part. However, due to the low rigidity, the blank of the thin-walled parts is often deformed after going through forming and clamping processes (Bi et al., 2019). Sometimes, the geometry of the part may vary from each other even within the same batch (Zhao et al., 2021). Since the deformation of the part is not taken into consideration during the design process. The NC programs generated based on the nominal part are unusable and cannot adjust the deformation, which may cause overcut and undercut (Sun and Jiang, 2018). Take the aircraft skin, for example, it is a typical thin-walled part with many pockets to be machined from the blank (Hao et al., 2019). The dimensional and shape tolerance is quite small compared to the large deformation of the blank. In this situation, the strict requirements for the consistency of the sculpted patterns and wall thickness may cause high scrap part ratio.

The critical issue in NC sculpting is how to adaptively sculpt the image patterns onto freeform thin-walled surfaces without knowing the position and geometry of the blank. Many researchers have focused on the modeling of deformation to predict and reduce machining errors. A finite element method (FEM)-based model is established to predict the deformation of the thin-walled part, and an iterative algorithm is used to calculate the milling force and deformation (Wang et al., 2015). Also, an FEM-based approach considers the springback deformation of the blank and predicts the milling error (Liu et al., 2017). The FEM is useful to execute with the aid of commercial simulation software. Nonetheless, it is time-consuming and requires a large amount of computational power to get the simulation results. Some works use mechanism analysis and modeling methods to calculate the deflection of the tool and blank. A deformation error prediction model is proposed to improve computational efficiency (Tuysuz and Altintas, 2018). The calculation method of deformation error that is based on the differential increments is proposed to reduce milling error using tool path modification (Li and Zhu, 2019). However, it is difficult to establish a precise model of deformation during the machining process. Since the OMM system develops rapidly for the strength of online measurement and high efficiency (Bastas, 2020), it is used to obtain the
geometry of the part during the machining process (Zhao et al., 2019). The geometry data acquired by OMM inspection are applied to compensate for the deformation error of an impeller by modifying the tool path (Huang et al., 2014). Most modeling approaches concentrate on elastic deformation, while plastic deformation occurs in many cases (Ding et al., 2020). Besides, it is difficult to realize the adaptive modification of dense cutting points (Ge et al., 2020). It is found that few works focus on the systematic procedures for adaptively sculpting on the deformed part.

An NC sculpting method of adapting the image patterns to complex surfaces using OMM is proposed. Only one clamping is needed by using the proposed method, which avoids repeated clamping errors. The geometric shape data of the part are acquired by the on-machine probe in real-time, which is the source data of the section of part reconstruction. The reconstructed part surface is flattened into the planar mesh, on which the extracted points of the patterns are projected. The rest of this paper is organized as follows: The problem description and solving methodology are given in Section 2. Section 3 describes the implementation of NC sculpting based on OMM of the part surface. The results of a case study and discussions are presented in Section 4. Finally, conclusions are summarized in Section 5.

2. Problem description and solving methodology

2.1 Problem description

The key to sculpting image patterns on the part surface is how to quickly extract image contours and generate contour lines on the surface. The main problems are as follows:

(1) The geometry condition of the part to be sculpted is unknown due to the deformation which results from forming and clamping processes. It means that the machining allowance may be distributed unevenly on the part.

(2) It is difficult to adapt the nominal tool paths to the part surface with local deformation.

(3) Since the tool path is generated based on the pattern contour which contains scattered points, the NC code should guarantee the smoothness of the patterns (Sun and Altintas, 2021).

2.2 Solving methodology

To solve the above problems, a pattern sculpting approach is developed (see Figure 1). After the part is clamped, the data of the surface to be sculpted are obtained by OMM. Then, the reconstruction of the measured surface is realized by the means of non-uniform rational basis spline (NURBS) (Liang et al., 2020; Hashemain and Hosseini, 2018). The ABF algorithm is used to convert the 3D surface to the 2D planar domain. The patterns are extracted from the image using OpenCV and OpenGL library functions. Considering the restraints of distance and angle, fast sorting of image discrete points can be realized. The area coordinate algorithm is used to achieve the fast positioning of image contour points onto the surface.

3. Implementation of adaptive sculpting

3.1 Surface profile measurement and surface reconstruction

Currently, the main methods of surface profile measurement are contact and non-contact measurement. The most commonly used instrument for contact measurement is the coordinate measuring machine (CMM) (Altinisik and Bolova, 2021), including the touch probe and scanning probe (Bastas, 2020). The surface profile data measured by CMM are accurate, however, the clamping error will be introduced when the part is fixated on the machine tool.
Therefore, a touch probe is mounted on a computer numerical control (CNC) machine tool to measure the surface profile of the part (see Figure 2).

Before the NC measurement, the measurement path of the probe needs to be planned. Based on UG software, the function for measuring path generation is developed. First, choose the surface of the part to be sculpted and obtain the cross-section lines of the surface by planes. A series of point sets are obtained on the cross-section lines. Considering measured surface, point sets and safety distance, the measurement path can be generated automatically (see Figure 3).

The curve interpolation process is completed using NURBS functions. NURBS surface is determined by control points, weight factors and node vectors. Given \( (m + 1) \times (n + 1) \) control points \( d_{i,j} (i = 0,1, \ldots, m; j = 0,1, \ldots, n) \) and the corresponding weight factor \( w_{i,j} (i = 0,1, \ldots, m; j = 0,1, \ldots, n) \), a control grid is formed. Then, given the times \( k \) and \( l \) of the parameters \( u \) and \( v \), and the corresponding node vectors \( U = \{ u_0, u_1, \ldots, u_{m+k+1} \} \) and \( V = \{ v_0, v_1, \ldots, v_{n+l+1} \} \), a NURBS surface of \( k \times l \) can be determined. The equation is as follows:

\[
P(u, v) = \frac{\sum_{i=0}^{m} \sum_{j=0}^{n} w_{i,j} d_{i,j} N_{i,k}(u) N_{j,l}(v)}{\sum_{i=0}^{m} \sum_{j=0}^{n} w_{i,j} N_{i,k}(u) N_{j,l}(v)} \tag{1}
\]

where \( N_{i,k}(u) \) and \( N_{j,l}(v) \) are recursive polynomials of B spline basis functions.
3.2 Part surface flattening

Due to the difficulty of surface calculation in 3D space, it is necessary to reduce the dimension of complex surfaces. Since reverse engineering (Yang et al., 2020) is a well-established technology for surface reconstruction, the triangular mesh is used to represent the surface, especially the STL file. The work (Sheffer and Sturler, 2001; Sheffer and Levy, 2005) mapped the triangular mesh surface to the 2D parametric domain only when the angle of each triangular mesh was constant or the deformation was minimal, thus providing an effective method to plan the image pattern in the 2D parametric domain. Considering the angle value as the parameter, the following mathematical model is established under the condition that the surface of the triangular mesh and the topological relationship between the triangular mesh and the surface are taken into account.

$$F^*(\alpha) = \sum_{i=1}^{n} \sum_{j=1}^{3} \left( \alpha'_i - \phi'_j \right)^2 w'_i$$

(2)

Use $i$ to represent the index value of the plane or space triangle, and use $j$ to represent the index of three angle values that are ordered counter-clockwise in the triangle, and use $n$ to represent the triangulation faces. $\alpha'_i$ is the angle of the plane grid triangle. $\phi'_j$ is the optimization angle corresponding to $\alpha'_i$, $w'_i = (\phi'_i)^{-2}$ is the weighting factor. The optimal angles $\phi'_j$ are derived from the following equation:
\[ \phi_j^i = \begin{cases} \beta_j^{(k)} \frac{2\pi}{\sum_{i} \beta_i^{(k)}} & k = 1 \ldots k_{\text{int}} \\ \beta_j^{(k)} & k \neq 1 \ldots k_{\text{int}} \end{cases} \]  

while \( k \) is the index number of the vertex, and \( k_{\text{int}} \) is the index number of the internal vertex. \( \beta_j^{(k)} \) is the angle of the triangle at the vertex \( j \). To ensure the surface and topology of the triangular mesh, three constraints are required. The Lagrange multiplier is used to eliminate the constraint, the final augmented objective function obtained is as follows:

\[
F(x) = F(\alpha, \lambda_1, \lambda_2, \lambda_3) = F'(\alpha) + \sum_{i=1}^{n} \lambda_{f_i}^{(1)} + \sum_{k=1}^{k_{\text{int}}} \lambda_{f_k}^{(2)} + \sum_{k=1}^{k_{\text{int}}} \lambda_{f_k}^{(3)}
\]

\( \lambda_1, \lambda_2 \) and \( \lambda_3 \) are Lagrange multipliers. To ensure the surface and topology of the triangular mesh are complete, the following three constraints are required:

\[
\begin{align*}
    f_i^{(1)} &= \alpha_i^1 + \alpha_i^2 + \alpha_i^3 - \pi = 0, i = 0, 1, \ldots, n; \\
    f_k^{(2)} &= \sum_{i} \alpha_i^{j(k)} - 2\pi = 0, k = 0, 1, \ldots, k_{\text{int}}; \\
    f_k^{(3)} &= \prod_{i} \sin(\alpha_i^{j(k)}+1)/\prod_{i} \sin(\alpha_i^{j(k)}-1) - 1 = 0.
\end{align*}
\]

The first constraint (1) ensures that each triangle face after mapping is valid. The second constraint (2) guarantees the plane of the inner vertex of the triangular grid. The third constraint (3) ensures the topological validity of the mesh (see Figure 4). The flattening result of two spatial surface meshes is shown as follows:

### 3.3 Extraction and sorting of pattern point set

The image patterns need to be processed before the sculpting, such as extracting the contours of image patterns and generating a set of discrete points. The NC sculpting lines are also the interpolation based on the ordered coordinate values in the tool path. After the discrete point set of the image is generated, the contour lines need to be divided to avoid collision when the tool moves between the contour lines. Besides, for the same contour line, there may be some scattered points that influence the smoothness of the tool paths. Therefore, each set of contour points needs to be sorted.

1. **Contour extraction and segmentation**

   The image needs to be processed to obtain the contour of the sculpting pattern using the OpenCV library. For image pattern extraction, the edge of the pattern is detected by the Canny operator. The contour number of edges and the point number of each contour line can be obtained using the contour search function. After the point set in the contour is read, the extraction and segmentation of the point set in the contour can be realized.

2. **Pattern scatter point sorting based on distance and angle**

   The point set obtained from the above image pattern is dense, and the distance between the points is 1 or \( \sqrt{2} \) unit pixels. However, there will be errors if the point sets are stored in the wrong order. There will be cross-lines that will make the final tool path not smooth
The existing sorting algorithms based on distance and angle require large computational power and time. For this reason, a region point set partitioning method based on the k-d tree is proposed. The set of nearest neighboring points of each point is established, which reduces the number of searches for nearest normal distance sorting.

Let the discrete pattern point set $P$ be $\{P_1, P_2, P_3, \ldots, P_n\}$. The $K$ points closest to a certain point $P_i$ in the point set $P$ are called the $K$-neighborhood point set of the point $P_i$ (see Figure 6). The image contours can be sorted in the $k$-d tree at each point by distance and angle.

(see Figure 5).
As shown in Figure 7, eight data points are selected and the nearest algorithm is used. Figure 7a and b show the wrong and right sorting order, respectively. On this basis, the angle algorithm combines with the nearest search algorithm. The algorithm is developed as follows:

**Step1:** Select a point \( P_1 \) arbitrarily, and solve the second point \( P_2 \) in the \( K \)-neighborhood of \( P_1 \) by anti-clockwise. \( P_1, P_2 \) are marked as searched points in the point set \( P_k \);

**Step2:** The closest point \( P_3 \) is searched in the \( K \)-neighborhood of \( P_2 \), and the cosine value of the angle between the \( P_1P_2 \) and \( P_2P_3 \) is determined. If \( \cos \alpha \in [0, 1] \), the point \( P_3 \) meets the conditions. The point \( P_3 \) is marked, and skip to step 4; Else, the point \( P_3 \) does not meet the conditions; Skip to step 3;

**Step3:** Search the second neighboring point of \( P_2 \) in the \( K \)-neighborhood, and determine the cosine value of the angle between \( P_1P_2 \) and \( P_2P_3 \). If \( \cos \alpha \in [0, 1] \), the conditions are met, the point \( P_3 \) is marked. Else, find the third neighboring point in the \( K \) neighborhood. Set the new \( P_3 \) and iterate until a point that satisfies the condition is found. If the traversal is completed, skip to step 5;

**Step4:** Set \( P_1 = P_2 \), \( P_2 = P_3 \), and skip to step 2;

**Step5:** End.

### 3.4 Planar point cloud inverse mapping

To realize the NC sculpting, it is necessary to locate the discrete point set of contour to the planar mesh, and reverse mapping relationship to the 3D part surface mesh to generate tool path lines. Therefore, the main research of this section is to map the discrete point sets of the image pattern to the part surface to realize the rapid generation of the tool contact path. To locate the contour points of the pattern into a spatial mesh, the discrete point sets are firstly located in a planar triangular mesh. Since the size of the pattern should fit the part surface area, the image contour point set needs to be transformed, including scaling, translation and rotation. The area coordinate method is used to determine the transformation relationship between the pattern point sets and planar triangular mesh.

1. Coordinate transformation

To determine the transformation relationship, the pattern point sets and planar triangular mesh both need to be unified into the same coordinate system. The scaling, translation and rotation of the pattern point sets can be achieved according to the following equations.
Translation equation:
\[
\nabla X = X_{\text{Center}} - X'_{\text{Center}} \\
\nabla Y = Y_{\text{Center}} - Y'_{\text{Center}}
\]

(\(X_{\text{Center}}, Y_{\text{Center}}\)) is the center point of the planar triangular mesh, (\(X'_{\text{Center}}, Y'_{\text{Center}}\)) is the center point of the discrete pattern point sets.

Scaling equation:
\[
S = \lambda \cdot \min \left( \frac{x_{\max} - x_{\min}}{x'_{\max} - x'_{\min}}, \frac{y_{\max} - y_{\min}}{y'_{\max} - y'_{\min}} \right)
\]

(\(x_{\max} - x_{\min}, y_{\max} - y_{\min}\), \(x'_{\max} - x'_{\min}, y'_{\max} - y'_{\min}\)) are the longest distance of the planar mesh and the discrete pattern point sets, respectively.

Rotation equation:
\[
R = \begin{bmatrix}
\cos \theta & -\sin \theta \\
\sin \theta & \cos \theta
\end{bmatrix}
\]
where \(\theta\) is the rotation angle of the image point around the \(Z\)-axis.

(2) Regional grid division

To determine the position of each pattern discrete point in the triangular mesh, the division of regional square grids is studied. First, the side length of the square grid needs to be determined. The equation is as follows:
\[
L_S = s \sqrt{(x_{\max} - x_{\min})(y_{\max} - y_{\min})}
\]

After the length of the square side is determined, the number of grids along the \(X\) and \(Y\) axes needs to be determined. The equation is as follows:
\[
R_X = (\text{int}) \left( \frac{x_{\max} - x_{\min}}{L_S} \right) + 1 \\
R_Y = (\text{int}) \left( \frac{y_{\max} - y_{\min}}{L_S} \right) + 1
\]

Therefore, all triangular mesh points and scattered pattern points are located within the grid (see Figure 8).

(3) Fast positioning of image points to planar mesh triangular patches

After the regional grid is divided, the triangle patch for each pattern point \(p_i\) is determined using the following equation:
\[
I = \text{int} \left( \frac{x_j - x_{\min}}{R_x} \right), J = \text{int} \left( \frac{y_j - y_{\min}}{R_y} \right)
\]

The index number of the grid where the pattern discrete point is located can be determined, and then the search can be performed in the grid at the \(i-th\) row and the \(j-th\) column. As the mesh vertex closest to the pattern point \(p_i\) is designed, the triangle patch index and the corresponding area coordinate value are calculated according to the area coordinate method.
The adoption of the area coordinate method can distinguish whether the image point $p_i$ is within a certain triangle patch (see Figure 9).

The three points A, B and C of the triangle are in the same plane, and any point $p$ in the plane can be represented by the following equation:

$$P - A = u \cdot (B - A) + v \cdot (C - A)$$

If point $P$ is in triangle ABC, the following three conditions must be satisfied:
1. $u \geq 0$
2. $v \geq 0$
3. $u + v \leq 1$

Let vector $B - A = (X_1, Y_1, Z_1)$, vector $C - A = (X_2, Y_2, Z_2)$, vector $P - A = (X_3, Y_3, Z_3)$, then the above equation can be changed:

$$K \begin{bmatrix} u \\ v \end{bmatrix} = b$$

Let $K = \begin{bmatrix} X_1 & X_2 \\ Y_1 & Y_2 \\ Z_1 & Z_2 \end{bmatrix}$, $b = \begin{bmatrix} X_3 \\ Y_3 \\ Z_3 \end{bmatrix}$. $B - A$ and $C - A$ is linearly unrelated, and the matrix $K$ is a full rank matrix, then $\begin{bmatrix} u \\ v \end{bmatrix} = K^+ \cdot b$ ($K^+ = (K^T K)^{-1} K^T b$).

The order of three points in each triangle patch of the 3D triangular mesh strictly corresponds to that of three points in the corresponding triangle patch of the planar mesh. According to the partition of the square grid and the location of the discrete points, the $[u, v]^T$ of each image point is recorded. Therefore, the following equation can be used to locate the image point set on the part surface quickly. The result of mapping the point in the planar triangle to the point in the corresponding spatial triangle is shown in Figure 10.
3.5 The generation of NC sculpting code

This research is based on five-axis NC sculpting to ensure that each cutter contact point has the same sculpting depth. However, the discrete points on the part surface are not the actual cutter contact point (Xu et al., 2020). The following equation is needed for compensation.

\[ P_{IN} = P - d_{IN} n_p \]  

(15)

Among them, \( P \) is the part surface point; \( P_{IN} \) is the actual tool contact point; \( d_{IN} \) is the sculpting depth. \( n_p \) is the normal vector of the discrete pattern point. The value \( n_p \) can be obtained by the area coordinate method. The equation is as follows:

\[ n_p = u \times n_{PA} + w \times n_{PB} + v \times n_{PC} \]  

(16)

where \( u, w, v \) are the area coordinates of each pattern point in the triangle mesh and \( n_{PA}, n_{PB}, n_{PC} \) are the three vertex normal vectors in a spatial triangular patch.

The vertex normal vector of an arbitrary space triangle mesh can be obtained according to the following equation:

\[ n_{Pi} = \frac{\sum_{\varepsilon=1}^{k} \alpha_{\varepsilon} A_{\varepsilon} n_{fe}}{\sum_{\varepsilon=1}^{k} \alpha_{\varepsilon} A_{\varepsilon}} \]  

(17)

\( n_{Pi} \) is the normal vector of an arbitrary point in a triangular mesh. As shown in Figure 11, the point \( Pi \) is the \( i \)-th vertex in the triangular mesh and there are \( k \) neighborhood triangular patches. In the neighboring triangular facet of the point \( Pi, A_{\varepsilon} \) is the area of \( \varepsilon \)-th triangular patch, \( n \) is the \( \varepsilon \)-th normal, \( \alpha_{\varepsilon} \) is the interior angle of the vertex \( Pi \) in the \( \varepsilon \)-th triangle patch.

4. Case study

To validate the feasibility of the proposed adaptive NC sculpting method, a pattern sculpting test on the egg surface is conducted in this section. The reconstruction result of the OMM point cloud data is shown in Figure 12.
The dimension of the egg surface is reduced from 3D to 2D using the ABF algorithm. This algorithm can quickly flatten the 3D surface into the 2D parameter domain, providing a good basis for the rapid mapping of the pattern point sets to the part surface. For spatial grids and planar domain grids, the MeshLab and OpenGL is used to display them respectively (see Figure 13). The results are as follows:

Based on the contour search function and Canny operator in the OpenCV library, the recognition, extraction and segmentation of image contours point sets are obtained. The C++ code has been implemented, and the final segmentation result is shown in Figure 14:

Based on the area coordinate method, fast positioning of image pattern point sets onto the flattened surfaces is achieved. The result is shown in Figure 15.
After quick positioning of the above pattern point sets onto a surface, the track line of the tool contact points on the surface is obtained. However, in the sculpting process, each tool contact point is offset along the normal by a certain distance according to the sculpting depth (see Figure 16). The red points are the tool contact points. The black ones are the actual tool path points. And the blue ones are normal for each pattern point.

These methods are implemented by C++ and validated by sculpting on the eggshell. All tests are run on a 64-bit Windows workstation with a 2.2 GHz processor and 32 GB memory. Measurement and sculpting tests were carried out on a DMG Linear 5-axis machine center, which has a maximum spindle speed of 42,000 r/min. The tool used for sculpting the eggshell is a ball-end milling cutter with a diameter of 1 mm and the sculpting depth is 0.15 mm. As shown in Figure 17, a pigeon with “Adaptive Machining” and a college badge is sculpted onto the eggshell, respectively. From the results, the proposed method can produce a decent sculpting effect on the part surface.

5. Conclusion
An adaptive sculpting method based on the OMM is developed. A sculpting test on a random egg is conducted, and the result shows that the shape of the designed pattern is well-preserved on the eggshell using the proposed method. Besides, the conclusions of this paper are as follows:
The automatic generation module of the measurement path is developed in UG software. The geometric data of the part surface can be obtained by OMM. For the measured point cloud data, the part surface reconstruction is realized using the NURBS method, which provides convenience for the parameterization of the complex surface.

The 3D complex surface is converted into a 2D planar domain to simplify the tool path planning. An image processing technique is proposed to extract and segment the image pattern points. To achieve the positioning of image pattern point sets onto a complex surface, the area coordinate method is proposed and explained in detail.

The inverse mapping of 2D image pattern points to 3D triangular meshes is implemented, which converts planar patterns points into spatial cutter contact points. The generated NC sculpting code maintains the vividness of the image.

References


Corresponding author
Zhengcai Zhao can be contacted at: zhengcaizhao@nuaa.edu.cn

For instructions on how to order reprints of this article, please visit our website: www.emeraldgrouppublishing.com/licensing/reprints.htm
Or contact us for further details: permissions@emeraldinsight.com