IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 5 December 2023, accepted 24 December 2023, date of publication 26 December 2023,
date of current version 9 January 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3347629

APPLIED RESEARCH

Research on Point Cloud Processing and Grinding Trajectory Planning of Steel Helmet Based on 3D Scanner

YUXIANG MENG, YU JIANG, YI LI, GUIBING PANG, AND QIANG TONG
College of Mechanical Engineering and Automation, Dalian Polytechnic University, Dalian 116034, China

Corresponding author: Qiang Tong (tongqiang.work@outlook.com)

This work was supported by the China Postdoctoral Science Foundation under Grant 2023M730452.

ABSTRACT As the continuous expansion of robot grinding objects, the traditional manual teaching and offline programming methods can no longer meet people's requirements for grinding trajectories. With the help of a 3D scanner, model data can be acquired quickly and accurately, which facilitates trajectory planning and effective grinding for workpieces. In this paper, a point cloud hole repair method is proposed based on B-spline surface. It integrates point cloud segmentation, feature point removal, smoothing and hole repair, aiming to solve the hole problem in steel helmet scanning. In addition, to generate a uniform grinding trajectory, a planning method is proposed based on an improved B-spline curve. This method combines curve homogenization with equidistant offset and uses k-nearest neighbor search. Iteratively identify and eliminate unreliable point data to determine the best point of contact. The effectiveness of the two methods is verified by trajectory planning on three types of steel helmets.

INDEX TERMS Robotic grinding, point cloud processing, holes repairing, trajectory planning, posture optimization.

I. INTRODUCTION

In recent years, robotic grinding systems have been widely used for the surface machining of free-type workpieces, such as wheel hubs and turbine blades [1], [2], [3]. By maintaining the contact point of the tool-workpiece and the tool normal direction (the tool position and posture), accurate surface grinding of the workpiece can be achieved [4]. It has replaced the traditional manual operation, greatly improves the processing efficiency and the working environment of workers.

For the surface grinding process of robotic steel helmets that we are interested in our research, the goal is to achieve an accurate and uniform removal of the surface material. It is similar to the robotic painting process in that they both achieve uniform coverage of the workpiece surface [5], [6]. However, compared with spraying, grinding has higher requirements for the accuracy of contact point information. The main characteristics of the grinding robot are: (1) The contact point conforms to the local geometric characteristics on the contact area. (2) The normal of the tool and the contact area is collinear. The common grinding trajectory planning methods are manual teaching, offline and online programming [7]. The manual teaching process is complex and inaccurate, and it is only suitable for simple working scenarios. Offline programming relies on computer-aided design (CAD) models of the workpiece. And it is difficult to process with actual workpieces with machining deformations or additional characteristics [8]. Online programming is usually used when it is difficult to obtain an accurate 3D model. The workpiece data are obtained by the sensor, and the motion trajectory is generated online. At present, manual teaching and offline programming are still the main modes in the robot manufacturing process [9]. However, the lack of accurate CAD models and complex parts are the biggest problems faced by both modes. Therefore, the development of sensor-based robotic grinding is an urgent need to solve the difficult and complex grinding tasks of automation [10], [11], [12].

The associate editor coordinating the review of this manuscript and approving it for publication was Bo Pu.

Many researchers have published their research on robotic grinding with different kinds of sensors. Wang et al. [13] proposed a grinding path planning method for robotic welds based on 3D profile scanners. All segments were modeled with a B-spline surface to obtain the parent surface, and welds were added to the parent surface to generate the grinding path. Yang et al. [14] designed a stripe projection system based on a digital light processing (DLP) projector, established a mathematical model of the weld, and solved the 3D weld path model and posture estimation. Vision sensors can get more information and handle more complex problems than other sensors. In the field of robot grinding, 3D sensors are commonly used, such as structured light cameras, laser displacement sensors and 3D scanners [15]. Compared to other 3D sensors, scanners are suitable for digitizing physical objects and can obtain 3D models with large arc changes.

The manufacturing process of steel helmets is generally stamping and forming, bending and welding, heat treatment, surface treatment and assembly. Before surface treatment, the surface of the steel helmet needs to be ground to ensure surface quality. However, the existing surface grinding method is mostly manual operation, and the work efficiency is low, and the environment is harsh. The intelligent robot grinding method based on vision sensor can automatically generate trajectory to achieve automatic grinding. Because there is a large gap between the model of stamping die and the steel helmet before surface treatment, it cannot be directly used for offline programming. As a typical three-dimensional manifold surface, the surface of the steel helmet has the characteristics of large arc change and small curvature. In this paper, a 3D scanner was used to scan the steel helmet to obtain the point cloud data of the grinding surface.

There are two important steps in the grinding of steel helmets based on 3D scanners. The first is model restoration, because there are no obvious features on the surface of the steel helmet, the feature points need to be pasted when scanning. This leads to the point cloud of the steel helmet with holes. The second is tool trajectory planning, which generates a grinding path for the manipulator. The common methods of point cloud restoration include polar coordinate transformation algorithm, surface fitting algorithm, triangulation algorithm, etc. Zhikun et al. [16] proposed an approximate fitting and interpolation method based on cubic b-spline curves, and repaired the point cloud holes after longitudinal slicing in the axial plane of the point cloud. Zhang et al. [17] proposed a hole repair model based on a fuzzy inference system, which combined the geometric and curvature information of boundary points. The fast and accurate estimation of the elevation coordinates of the repair point was realized. However, these methods are not suitable for estimating manifold surface holes in unorganized point clouds. In terms of trajectory planning, a common strategy is to reconstruct the workpiece point cloud into a CAD model by reverse engineering. Zhou et al. [18] reconstructed the 3D model of the workpiece by using a linear iterative near-point algorithm, and extracted the 6-degree-of-freedom welding path of the edge point of the weld. Li et al. [19] reconstructed the 3D model of the worn part, identified the boundaries of the damaged area of the part, and realized the accurate toolpath generation for the remanufacturing operation. However, these methods are time-consuming and are not suitable for automatic path planning for spherical surfaces with large radians.

Aiming at the problem of hole repair and trajectory planning in the generation of steel helmet grinding trajectory. In this paper, a manifold surface hole repair algorithm based on B-spline surface and a trajectory generation algorithm based on improved B-spline curve is proposed. The overall planning process is shown in FIGURE 1 The rest of the paper is organized as follows: Section II describes the hole repair algorithm, including point cloud segmentation, feature point removal, smoothing and repair. Section III describes the trajectory planning algorithm, including attitude setting, generating seed curve, curve offset and attitude optimization. In Section IV, the helmet model is repaired with B-spline surface to generate uniform contact points with B-spline curves, resulting in a grinding trajectory. Finally, the accuracy and robustness of the algorithm are verified, and the verification results are given.

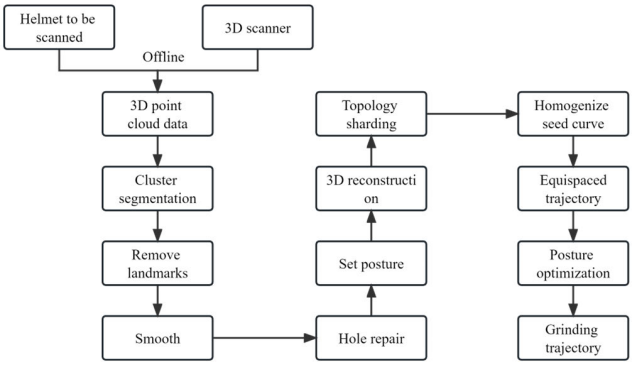


FIGURE 1. Flowchart of steel helmet grinding trajectory planning based on scanning point cloud.

II. PREPROCESSING AND HOLE REPAIRING FOR POINT CLOUDS

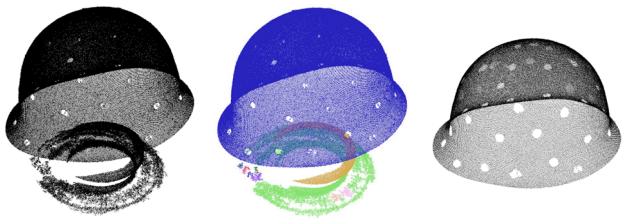
In grinding, the type of tool, feed rate, normal pressure and motion trajectory have a great impact on the machining quality. The focus of this study is on the generation of surface grinding trajectories of steel helmets without accurate CAD models. We used 3D scanner (POP2, REVOPOINT3D, Xian, China) to obtain the point cloud data of the steel helmet to be processed, its parameters were shown in TABLE 1. During the scan, we first pasted artificial feature points for the helmet. Then the helmet was placed on the rectangular bracket to keep it away from other objects. Finally, they were placed together on the turntable, and a tripod was used to fix the scanner at a suitable angle. The model scanning of the helmet was completed after the turntable rotated for 3 turns.

To obtain complete helmet point cloud data, the overall scanning object of the scanner is usually larger than the

TABLE 1. Device parameters for POP2.

Equipment Variables	Parameter Value
single frame accuracy	0.05 mm
single frame scanning range	210 × 130 mm
working distance	150 - 400 mm
minimum scanning size	20 × 20 × 20 mm
frame rate	10 fps
interface type	USB 3.0

helmet itself. This will lead to obtain other objects, such as the support structure and workbench of the helmet. In addition, since there are no obvious features on the surface of the steel helmet. It is necessary to manually paste the feature points on the surface to ensure effective scanning. This results in distortion of the helmet data near the feature point, resulting in holes, bumps, and noise spots. These must be removed before the trajectory is generated. The helmet region can be segmented by Euclidean clustering, noise and isolation points can be removed by filtering, and feature points can be removed and repaired by the algorithms described in Sections II-B and II-D. The point cloud processing tools are mainly PCL (Point Cloud Library) 1.13.0. FIGURE 2 depicts the preprocessing process for point clouds.



(a) Original point cloud. (b) Cluster segmentation. (c) Feature point removal.

(d) Smoothing. (e) Hole repair. (f) Down-sampling.

FIGURE 2. The preprocessing process of the point cloud.

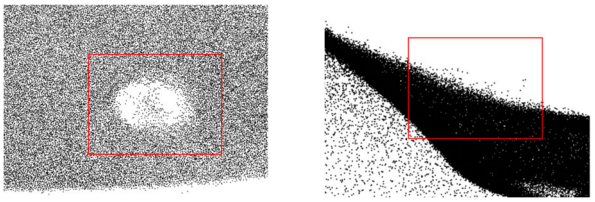
A. SEGMENTATION

The scale of the data obtained by 3D scanning is too large, which is not conducive to subsequent processing. Before segmentation, the voxel down-sampling method was used to uniformly down-sampling the point cloud data. This reduces the density of point cloud data and allows for more efficient processing and storage of point clouds.

In this paper, the KD tree proximity search method was used to cluster and segment the steel helmet point cloud. The main idea of this division is explained below. Search for all points between neighbors that were less than a set threshold (1.5 mm) and clustered them in the same set. The number of points in the set were also recorded at the same times. After the search was completed, the number of points was used as the extraction criterion. The largest points number of sets was the required helmet point cloud data, and the segmentation effect is shown in FIGURE 2(b).

B. REMOVE FEATURE POINTS

Due to the smooth surface and consistent color of the helmet, there was a lack of matching feature points during the 3D scanning process. It cannot be stitched well together, and the feature points need to be pasted manually. As a result, the surface of the obtained steel helmet point cloud has characteristic point holes of different sizes, which cannot be effectively identified. Moreover, the distribution of point cloud data at the edge of the hole is messy, as shown in FIGURE 3. Feature points data need to be identified and deleted.



(a) Front view of holes. (b) Side view of holes.

FIGURE 3. The preprocessing process of the point cloud (the red box in b is the side view of the red box hole in a).

In this paper, the boundary detection algorithm based on the maximum angle criterion was used to determine whether the sampling point and its neighborhood point were on the same side. If it was on the same side, it was the edge point, and vice versa. Based on this, a feature point removal algorithm was proposed. In boundary detection, we set the search radius to 3mm and the angle threshold to 180 °. All extracted boundary points were classified using the Euclidean clustering method in Section II-A, and the boundary point set was denoted as $L_b = \{l_{b1}, l_{b2}, l_{b3}, \ldots l_{bn}\}$. We sorted L_b according to the number of points, found the maximum set l_{bmax}, and defined it as the true boundary of the steel helmet. Since it was not needed for the following calculation, l_{bmax} was culled. The KD tree method was used to search for the point cloud near L_b to be deleted. Considering the different sizes of feature holes, the search radius should be dynamic, so it was set to:

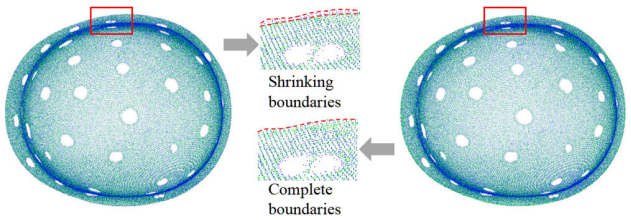
$$r_i = k \frac{\sum_{i=1}^{n_{\text{boud}}} n_i / n_{\text{boud}}}{n_i} \tag{1}$$

where k is the scaling factor, n_i is the number of points for each boundary set, and n_{boud} is the number of boundary sets. The duplicate points in the search process were ignored to obtain the data of the artificial feature points to be deleted. By deleting these, the real steel helmet data can be obtained without the influence of feature points, as shown in FIGURE 2(c).

C. DENOISING AND SMOOTHING

The process of obtaining point cloud data through the scanner is susceptible to the influence of the external environment and equipment. The obtained data contain a certain amount of noise, which will affect the subsequent operation. Therefore, denoising and smoothing are important parts of point cloud preprocessing. There are two main types of anomalous noise points, one is the long-distance outlier from the target point cloud, and the other is the short-distance outlier close to the target point cloud [20]. The first type of points has been removed in Section II-A, and the second type of points are mainly treated in this section.

Considering that feature points are invalid point cloud data, which will affect the smoothing effect of point clouds, we put the smoothing process after feature point removal. Smoothing of point clouds is usually performed by parametric or non-parametric regression. Parametric regression requires prior knowledge of regression formulas, which can represent the data well, but in most cases cannot be parameterized with a definite function [21]. Therefore, non-parametric regression is the preferred choice for point cloud smoothing. They commonly used in point cloud processing include gaussian filtering, moving least squares (MLS), median filtering, and bilateral filtering [22]. Both Gaussian filtering and MLS have good results, but Gaussian filtering will cause a certain boundary shrinkage, and MLS will run slowly. In this paper, Gaussian filter was used to process the overall data, and MLS algorithm was used to process the boundary point cloud. While maintaining the integrity of the smooth boundary (compared to Gaussian filtering), the smoothing efficiency was improved (2.271 s faster than MLS at 212984 points), as shown in FIGURE 4. The blue is the original point cloud, and the green is the smoothed point cloud.



(a) Gaussian filtering effect. (b) Gaussian filtering hybrid MLS effect.

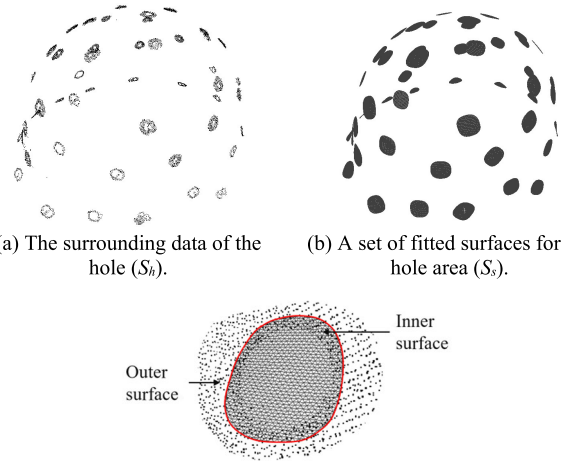
FIGURE 4. Point cloud smoothing results.

D. HOLE REPAIR

After segmentation, feature point removal and smoothing, the 3D point cloud has nearly circular holes at the feature points, making the overall data incomplete, as shown in FIGURE 2(c). In this paper, an algorithm based on B-spline surface was used to reconstruct the surface of the data around the hole. The surrounding data were fitted by the B-spline curve and the B-spline surface was segmented to obtain the hole data discretely.

1) ACQUISITION OF SURFACE FITTING DATA

After the removal of the feature points, the hole boundary becomes more obvious. It can be accurately extracted by the normal-based boundary extraction algorithm. The extracted boundary points were divided by Euclidean clustering. And the set of the largest number of points (the real boundary of the steel helmet) was removed, so as to determine the boundary of the hole to be repaired. Then, the KD tree method was used to search for nearby point clouds to obtain the hole edge set $S_h=\{s_{h1},s_{h2},s_{h3},\ldots s_{hn}\}$, as shown in FIGURE 5(a). S_h was used as the basic data for B-spline surface fitting. This is largely consistent with the flow in Section II-B.



(a) The surrounding data of the hole (S_h). (b) A set of fitted surfaces for the hole area (S_s).

(c) Projection of s_{h1} in a local coordinate system.

FIGURE 5. The data around feature holes.

2) SURFACE FITTING

The obtained point cloud data are the periphery ring of the area to be repaired. We used the cubic B-spline surfaces to fit it. The surface was converted to a triangular mesh surface by setting the split density, and some of the vertices of the mesh were the repaired hole point cloud data.

For the surrounding set s_{h1} of the area to be repaired, the local coordinate system was constructed by principal component analysis (PCA). The bounding box obtained by PCA was projected onto the XOY plane of the local coordinate system as the initial iterative surface. Since the scanner obtains a disorderly distribution of point clouds, there is no need to consider the point cloud arrangement in the processing. We used a cubic B-spline surface to iteratively fit s_{h1}, given the node vectors for the parameter axes u and v:

$$U = \left[u_0, u_1, \cdots, u_{m+p}\right] \tag{2}$$

$$V = \left[v_0, v_1, \cdots, v_{n+q}\right] \tag{3}$$

where p and q are the degrees of the parameters u and v, m and n represent the number of u and v, and the expression of the B-spline surface can be described as:

$$p(u,v) = \sum_{i=0}^{m}\sum_{j=0}^{n} P_{ij}F_{i,p}(u)F_{j,q}(v)\ , \quad (u,v) \in [0,1]\times[0,1] \tag{4}$$

where $P_{ij}(i = 0,1,2,\ldots m,\ j=0,1,2,\ldots n)$ is the set of $(m+1)(n+1)$ control points in space, $F_{i,p}(u)$ and $F_{j,q}(v)$ is the B-spline basis function, which is determined by the node vectors U and V according to the de Boor-Cox recursive formula, respectively [23]. The basis function of the B-spline surface is:

$$F_{i,p}(u) = \frac{1}{p!}\sum_{k=0}^{p-i}(-1)^k C_{p+1}^k (t+p-i-k)^p,$$
$$u \in [0,1], i = 0,1,2,\cdots,m \tag{5}$$

$$F_{j,q}(v) = \frac{1}{q!}\sum_{k=0}^{q-j}(-1)^k C_{q+1}^k (t+q-j-k)^q,$$
$$v \in [0,1], j = 0,1,2,\cdots,n \tag{6}$$

When p and q are both 3, (4) is the expression of the cubic B-spline surface, and (5)(6) are its basis functions. All subsets in S_h were fitted to obtain the surface set S_s of the hole region, as shown in FIGURE 5(b).

3) SURFACE SEGMENTATION

Since the initial iterative plane is the projection of the S_h subset bounding box, its area is larger than that of the subset projected on the local coordinate system. This will lead to a low overlap rate between the partially surface and the helmet point cloud after iterative fitting, and the wrong fitting data will occur. To get rid of the erroneous surface peripheral data, the subset of S_h needs to be segmented. We used the cubic B-splines to fit S_h to obtain the mean boundaries of the subsets $L_m=\{l_{m1},l_{m2},l_{m3},\ldots l_{bn}\}$, where the expression of the B-spline is as follows:

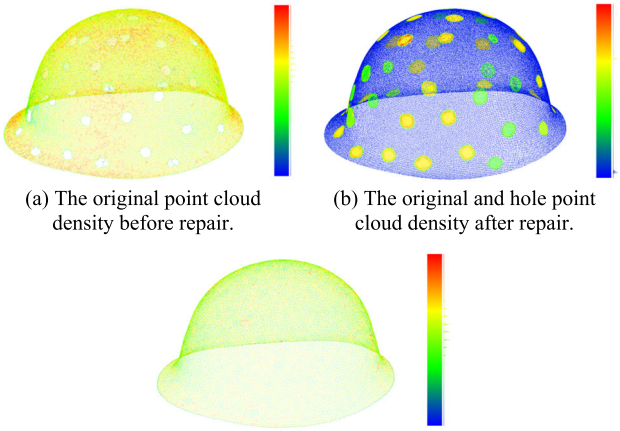
$$P(t) = \sum_{i=0}^{n} P_i F_{i,k}(t), (0 \le t \le 1) \tag{7}$$

where $P_{ij}(i=0,1,2,\ldots m,\ j=0,1,2,\ldots n)$ is the set of $(n+1)$ control points in space, $t=\{t_0,t_1,t_2,\ldots t_m\}$ is $(m+1)$ order vector, $F_{i,k}(t)$ is the basis function of the k-order B spline. When $k=3$, (7) is the expression of the cubic B-spline. S_s was triangulated to obtain a mesh set $S_t=\{s_{t1},s_{t2},s_{t3},\ldots s_{tn}\}$, and L_m was used as the dividing line to split S_t into two surfaces, as shown in FIGURE 5(c). And the inner surface was selected as the repair surface of the hole.

Specifically, if we took s_{t1} as an example, set the average boundary of the fit to l_{m1}. The local coordinate system of the subset s_{t1} was constructed by PCA, and s_{t1} and l_{m1} were projected onto the XOY plane of the coordinate system. Traverse the vertex coordinates in s_{t1} to judge whether the vertex is located within l_{m1}. The vertices within it were placed in the same set as l_{m1} until the end of the traversal. This set is the repaired point cloud data of the hole.

E. POINT CLOUD DOWN-SAMPLING

It is difficult for the density of the repaired hole point cloud to be consistent with the original point cloud, as shown in FIGURE 6(a)(b). To synchronize the densities, we set the triangulation density of S_s to a higher level, and then down-sampling the point clouds at the same time. In this paper, a voxel filter down-sampling algorithm was used to alleviate the density inconsistencies between both. It can be seen from FIGURE 6(c) that the synchronized point cloud has a more uniform density distribution.



(a) The original point cloud density before repair.

(b) The original and hole point cloud density after repair.

(c) Synchronous original and hole point cloud density.

FIGURE 6. Point cloud density diagram (the average distance between the current point and the neighborhood points was used to show the point cloud density).

III. TRAJECTORY PLANNING

Before the robot grinds, the grinding path and contact points must be generated in advance, and the tool posture must be calculated. To ensure the smoothness and accuracy of robot movement, two principles must be followed [24]:

- The robot motion path must ensure that the tool contact process conforms to the local geometry on the contact area. For path inflection points that only keep C0 continuous, it is necessary to avoid or smooth the movement posture as much as possible.
- The normal of the surface contact point must be collinear with the normal of the contact point on the tool to ensure the accuracy of removal.

In this section, we generate a uniform grinding trajectory through posture tuning, surface reconstruction and topological slicing, isometric trajectory generation and tool posture optimization.

A. MODEL POSTURE TUNING

The posture of the reconstructed surface model depends on the scanned spatial posture under the scanner. Since there is no guarantee that the scan space will be in the same posture every time it is scanned. The posture deviation of the reconstructed surface model is large, which is not conducive to subsequent calculation and analysis.

To unify the model posture during trajectory planning, the PCA was used to construct the directed bounding box (OBB) of the surface mesh. Then the local coordinate system of the model was obtained. Its XYZ axis were divided into direction

vectors $\boldsymbol{Y}_l$, $\boldsymbol{X}_l$ and $\boldsymbol{Z}_l$ according to its eigenvalue size, and normalized. Taking the vertex of OBB as the origin, the length and width as the YX axis, and the origin to the adjacent point as the positive direction, the model posture matrix of the right-hand system was constructed:

$$\boldsymbol{R}_l = [\boldsymbol{X}_l, \boldsymbol{Y}_l, \boldsymbol{Z}_l] \tag{8}$$

In the eight coordinate systems composed of OBB vertices, there are four vertices whose directions from the origin to the three adjacent points are in the positive direction of the coordinate system, as shown in FIGURE 7 To determine a fixed vertex as the origin, the distance from the centroid of the point cloud to the surrounding box surface was used as the judgment criterion. The formula for calculating the centroid coordinates of the point cloud is as follows:

$$\boldsymbol{P}_c = \frac{1}{n}\sum_{i=0}^{n} \boldsymbol{r}_i \tag{9}$$

where r_i=(x_i,y_i,z_i),i=1,2,... n is the coordinates of each point, and n is the number of points. The distances of P_c from a, b, c and their opposite plane were calculated, and the deviations of the opposite plane distances were 2.876, 12.489 and 2.943 mm, respectively. It can be seen that there is a significant deviation between b and its opposite plane. The plane far away from the center of mass is selected as the reference. By setting the perpendicular direction from the centroid to reference plane as the negative Y axis, the two coordinate systems O2 and O3 could be separated, as shown in FIGURE 7. Due to the small deviation of the model in the X and Z directions, the method is not suitable for their judgment. In addition, according to the characteristics of the helmet point cloud, the direction from the boundary to the centroid was set to be the positive Z axis. The coordinate systems could be separated again to obtain the local coordinate system O3 of the model. Transform the initial point cloud $\boldsymbol{P_{in}}$ of the model by (10) to obtain the corrected point cloud $\boldsymbol{P_{out}}$.

$$\boldsymbol{P_{out}} = \boldsymbol{R}_l^{-1}\boldsymbol{P}_{in} - \boldsymbol{P}_{O3} \tag{10}$$

where $\boldsymbol{P_{O3}}$ is the origin coordinate of O3.

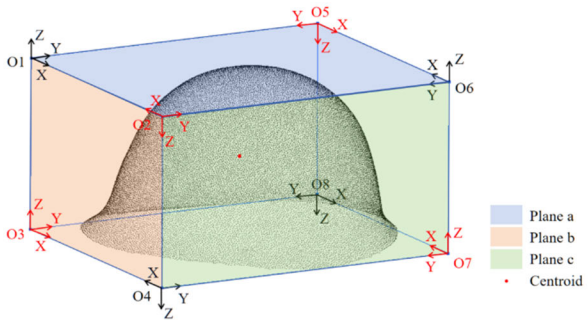
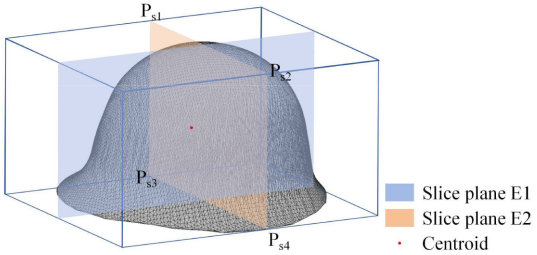


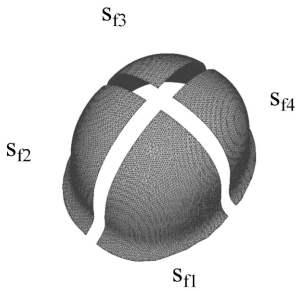
FIGURE 7. Schematic diagram of the bounding box.

B. SURFACE RECONSTRUCTION AND TOPOLOGICAL SHARDING

The triangular mesh is a way to store and express data in the form of facets to represent objects. This approach makes the surface geometry of 3D objects easier to process and analyze compared to point cloud data [25]. In this paper, the greedy projection triangulation algorithm was used to reconstruct the surface of the steel helmet. Firstly, the preprocessed point cloud was projected onto the two-dimensional plane for triangulation to establish the topological relationship. Then, the relationships between points in 2D space were mapped to 3D space to establish topological relationships in 3D space and complete surface reconstruction, as shown in FIGURE 8(a).



(a) Three-dimensional reconstruction of the model.



(b) Topological segmentation.

FIGURE 8. Schematic diagram of the shards of the mesh model.

FIGURE 8(b) shows the topological sharding of the model. This article used OBB to obtain the location of the slice plane. Firstly, the intersection point of the bounding box and the model was calculated in the Y-axis direction, and the two points P_{s3} and P_{s4} were obtained. Then they were translated 155.3 mm (the height of the bounding box in the Z-axis direction) along the Z-axis to obtain P_{s1} and P_{s2}, which formed the slice plane E1, as shown in FIGURE 8(a). E2 was constructed along the X axis in the same way. Finally, the steel helmet model was sharded by projection method based on the slice planes E1 and E2. A set of shards for the model S_f=$\{s_{f1}$,s_{f2},s_{f3},$s_{f4}\}$ were got.

C. TRAJECTORY GENERATION

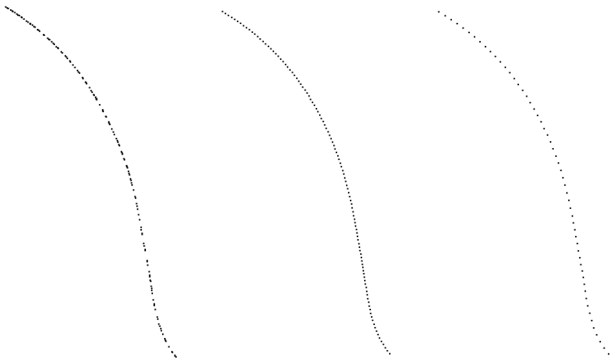
Since the robot cannot move continuously along the motion path, a series of tiny straight segments are used to approximate the curved motion. This means that the motion path needs to be discretized into a series of points, which can be referred to as "contact points". Considering that the curvature of each point on the surface of the steel helmet

is small and approximately equal, the uniform point on the curve is used to replace the contact points of equal chord error. This section studies the generation of equidistant contact points on the helmet model.

1) SEED CURVE GENERATION

In the process of motion path generation, the selection of seed curves is the most important. In this paper, the cross-sectional curve of E1 intersecting the model was selected as the initial reference curve. But the boundary points generated by the intersection of the triangular mesh and the slice were not uniform, as shown in FIGURE 9(a). To generate a uniform contact point trajectory, this paper used the B-spline curve interpolation represented by (7) to fit the point set. The basis function of the curve is:

$$F_{i,k}(t) = \frac{1}{k!}\sum_{j=0}^{k-i}(-1)^j C_{k+1}^{j}(t+k-i-j)^k, \quad t \in [0,1], i = 0, 1, 2, \cdots, n \tag{11}$$



(a) Initial situation. (b) Homogenization (interval 2mm). (c) Homogenization (interval 4mm).

FIGURE 9. The intersection of model s_{f1} with a cross-section.

When $k = 3$, there is a basis function of cubic b-spline curve:

$$\begin{cases} F_{0,3}(t) = \frac{1}{6}(1-t)^3 \\ F_{1,3}(t) = \frac{1}{6}(3t^3 - 6t^2 + 4) \\ F_{2,3}(t) = \frac{1}{6}(-3t^3 + 3t^2 + 3t + 1) \\ F_{3,3}(t) = \frac{1}{6}t^3 \end{cases}, \quad t \in [0,1] \tag{12}$$

Substituting the (12) into (7), the cubic B-spline curve formula is obtained:

$$P(t) = P_0F_{0,3}(t) + P_1F_{1,3}(t) + P_2F_{2,3}(t) + P_3F_{3,3}(t) \tag{13}$$

In a multi-segment cubic B-spline curve, if there are m control points, the curve will be generated as m-3 segments. FIGURE 10 is a schematic diagram of a single third-order B-spline curve. P_0, P_1, P_2, and P_3 are the four fitting control points of the first curve. M_1 and M_2 represent the midpoint of the straight lines P_0P_2 and P_1P_3, respectively. The starting point S and the end point E of the first curve are located at one-third of P_1M_1 and P_2M_2, respectively [26].

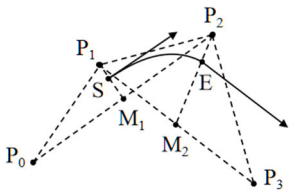


FIGURE 10. Schematic diagram of single third-order B-spline curve.

The traditional curve interpolation is usually calculated by uniform interval t-value interpolation. However, this method depends on the density between points. When the density is different, the interpolation results are quite different, and the expected uniform effect cannot be achieved in this paper. To solve this problem, we propose an equidistant (point interval) advancing t-value algorithm, and the interpolation effect is shown in FIGURE 9(b)(c). The specific steps are as follows:

1. The average interval method was used to estimate the arc length of the fitted curve. t was set 0.05 to obtain 20 points on a single curve, and calculating the spacing between adjacent points. Then, the length of the first curve l_{t1} was obtained by summing all the point spacing.

2. Repeating step 1 to calculate the length of the whole curve L_t=$\{l_{t1}, l_{t2}, l_{t3}, \ldots l_{bm}\}$. The step length d_s was set, and the d_s was corrected by (14) to obtain d_s'.

$$d_s' = \frac{L_t}{\left\lceil \frac{L_t}{d_s} \right\rceil} \tag{14}$$

3. d_s' was used to divide l_{t1}, and a uniform point set S_{Pr}=$\{\mathrm{P}(t_{r1}), \mathrm{P}(t_{r2}), \mathrm{P}(t_{r3}), \ldots \mathrm{P}(t_{rn})\}$ was obtained on l_{t1}, where:

$$n = \frac{l_{t1}}{\lfloor d_s' \rfloor} \tag{15}$$

$$t_{ri} = \frac{id_s'}{l_{tj}}, (i = 0, 1, 2, \cdots, n, j = 0, 1, 2, \cdots, m) \tag{16}$$

The remaining length L_e of the current curve was recorded by $L_e = l_{t1} \bmod d_s'$ and added to the calculation of the length of the next curve.

4. In the second curve, (15)(16) were modified into (17)(18), and the curve was divided. L_e was reset by $L_e = (l_{t2} + L_e) \bmod d_s'$.

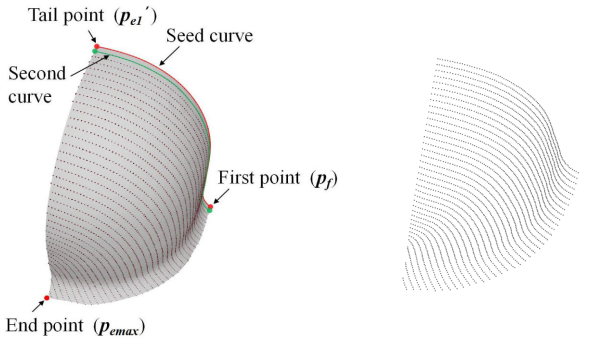
$$n = \frac{l_{t2} + L_e}{\lfloor d_s' \rfloor} \tag{17}$$

$$\begin{cases} t_{ri} = \frac{d_s' - L_e}{l_{tj}} & (i = 0, j = 0, 1, 2, \cdots, m) \\ t_{ri} = t_{r0} + \frac{id_s'}{l_{tj}} & (i = 1, 2, \cdots, n, j = 0, 1, 2, \cdots, m) \end{cases} \tag{18}$$

5. Repeat step 4 until the end of the whole curve division. If n was zero, the current curve segment was not divided, and recorded $L_e = (l_{tj} + L_e)$.

2) EQUIDISTANT OFFSET OF CURVE

The offset of the curve is usually operated by lines or points, but the equidistant offset between lines is complex and difficult to implement on complex surfaces. In this paper, the local point data of the triangular mesh were used to realize the equidistant generation of the trajectory. The results are shown in FIGURE 11(b).



(a) The curve offset process. (b) The overall trajectory on s_{f1}.

FIGURE 11. Schematic diagram of trajectory generation.

According to the vertex normal vector of the triangular mesh in the s_{f1}, the corresponding normal vector $\boldsymbol{n_z}$ of the current point $\boldsymbol{p}$ could be obtained by using the area coordinate. At the same time, the tangent vector $\boldsymbol{n_y}$ was calculated by using the adjacent point information, and the offset direction $\boldsymbol{n_x}$ could be obtained by solving the cross product of both. The linewidth d_w was set to further obtain the offset point $\boldsymbol{p}' = p + d_w n_x$. Since the local curvature of the helmet model is very small, this paper used an ideal offset $d_w n_x$ as an approximation. And the triangular mesh nearest point solution method based on octree was used to correct $\boldsymbol{p}$ to obtain $\boldsymbol{p}'$. When the distance between $\boldsymbol{p}'$ and s_{f1} was greater than 0.5mm, the point was removed to ensure that the contact point was always located on the surface of the model. Meanwhile, the curve formed by $\boldsymbol{p}'$ was homogenized by the method in Section III-C-I. The position of the second curve is shown in FIGURE 11(a).

It can be clearly seen from FIGURE 11(a) that with the offset of the trajectory, the tail points of some curves are not on the boundary of s_{f1}. This is because $\boldsymbol{n_x}$ of contact point at the end is not necessarily in the same direction as the boundary of s_{f1}. Considering that the distance between the point and the boundary is short, we used the nearest two-point information at the end to correct to solve this problem. Firstly, the two points $\boldsymbol{p_{e1}}$ and $\boldsymbol{p_{e2}}$ at the end of the curve were selected to calculate the vector $\boldsymbol{n_e} = p_{e1}\text{-}\boldsymbol{p_{e2}}$, and $\boldsymbol{n_e}$ was orthogonalized. Then, the minimum distance d_e between the end point and the boundary in the Y direction was calculated, and $\boldsymbol{p_{e1}}$ was corrected to obtain $\boldsymbol{p'_{e1}}$:

$$\boldsymbol{p_{e1}}' = \boldsymbol{p_{e1}} + \frac{|\boldsymbol{p_{e1}}(1) - \boldsymbol{p_{e2}}(1)|}{d_e}\boldsymbol{n_e} \tag{19}$$

Bring $\boldsymbol{p'_{e1}}$ into the original curve and repeat the trajectory offset. When the termination point $\boldsymbol{p_{emax}}$ satisfies:

$$(\boldsymbol{p_{emax}} - \boldsymbol{p_f}) \cdot (\boldsymbol{p_{emax}} - \boldsymbol{p_{e1}}') \leq 0.35 d_w \tag{20}$$

The iteration is terminated, where $\boldsymbol{p_f}$ is the first point of the curve. The complete migration trajectory is shown in FIGURE 11(b).

3) ELIMINATION OF ERROR POINT INFORMATION

According to the bending angle θ of the curve, there are four cases in the process of contact point offset, as shown in FIGURE 12. Since the first case does not appear in the trajectory generation of the helmet model, this study ignores its influence. For the 2-4 case, it can be seen that when $\theta = 90°$, the contact point shows a uniform offset. When $\theta > 90°$, the contact point appears corner aggregation and wrong point information. Among them, the corner aggregation problem can be solved by the above curve homogenization method. In the error point problem, the sorting sequence of the offset points does not change compared with the initial curve, but the spatial position of the wrong points changes. This will lead to incorrect fitting during the migration process, as shown in FIGURE 13.

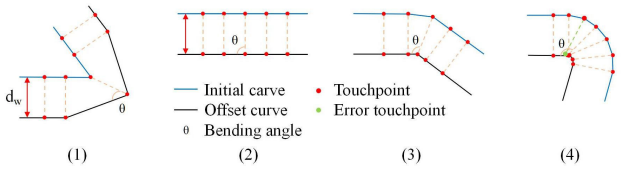
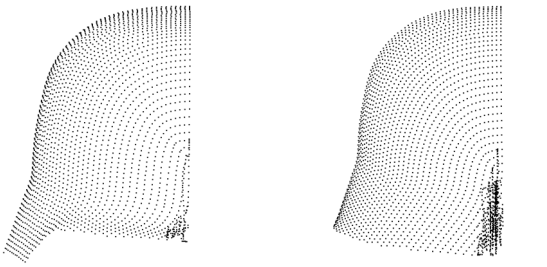


FIGURE 12. Schematic diagram in the contact points offset.



(a) Errors on the front of the helmet. (b) Errors on the back of the helmet.

FIGURE 13. Error fitting during offset.

To solve this problem, we constructed the search structure of the initial curve using the KD tree. When the offset curve was not a seed curve, a k-nearest neighbor search was performed on the contact points on the current curve. If the nearest distance d_{min} satisfied:

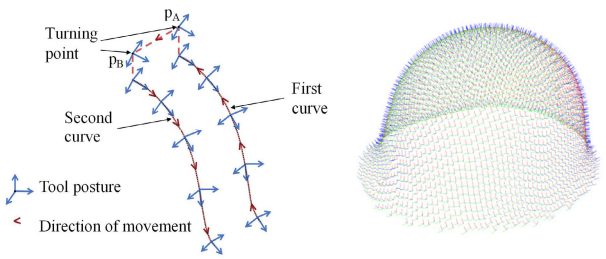
$$0.95 d_w \leq d_{\min} \leq 1.02 d_w \tag{21}$$

the point was retained. Then the current curve was fitted with the identified contact point to eliminate the influence of the wrong point position.

D. TOOL POSTURE OPTIMIZATION

For each contact point, the grinding force direction of the end tool of the manipulator should be ensured to be on the same line as the normal line of the model surface. The force sensor can control the force of the grinding tool to be constant. But the actual pressure on the model surface is related to the angle between the tool and the normal direction of the current contact point [27]. The normal vector estimation method based on triangular mesh can determine the normal information, and the adjacent point position information can determine the tangential information. The posture information can be obtained by solving the cross product of both.

To ensure the minimum posture change between the two adjacent trajectories, we set that the tangential vector of each contact point had the same projection direction on the Y axis. The transition points p_A and p_B were constructed between the two trajectories as shown in FIGURE 14(a). Their position information was obtained by moving the head and tail points of the first and second trajectories along the Z-axis by 5mm, respectively. The posture information was obtained by the tool posture transformation relationship $\theta(\alpha, \beta, \gamma)$ between p_A and p_B. The tool postures of p_A and p_B were $\boldsymbol{R_A}(\alpha/3, \beta/3, \gamma/3)$ and $\boldsymbol{R_B}(2\alpha/3,2\beta/3,2\gamma/3)$ respectively. The posture optimization between the two trajectories is shown in FIGURE 14(a). The overall optimization effect of the model is shown in FIGURE 14(b). To view the display effect more conveniently, the posture information of the transition point is not shown in FIGURE 14(b).



(a) The optimization between two adjacent trajectories.

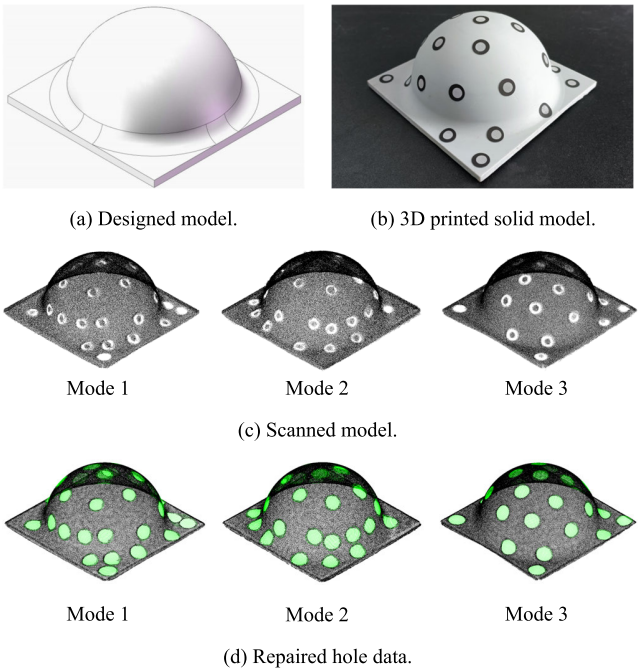
(b) The overall optimization effect.

FIGURE 14. The tool posture optimization effect of the grinding trajectory.

IV. RESULTS AND ANALYSES

A. VERIFICATION OF HOLE REPAIR ALGORITHM

Because the accurate model of the helmet is difficult to obtain, direct verification of its repair algorithm will lack the reference model. We used SolidWorks 2018 (Solidworks, Inc., Dassault Systemes, Waltham, MA, USA) to design a hemisphere model with a radius of 10 cm. Three solid models were printed by a photocuring 3D printer (HALOT-LITE, Creality, Guangdong, China), and then point cloud data were collected by a scanner, as shown in FIGURE 15. In this paper, the algorithm proposed in Section II was used to repair the hemisphere model to verify the accuracy of the algorithm.



(a) Designed model.

(b) 3D printed solid model.

(c) Scanned model.

(d) Repaired hole data.

FIGURE 15. Hemisphere model.

The experimental data are taken from the solid models of three hemispheres. Each model collects 10 sets of point cloud data in the state of pasting feature points, and a total of 30 sets of hemisphere models with hole defects. In the repair process, the point cloud data of the 1:1 hemisphere model were preprocessed firstly. Then, the edge position of each feature mark was determined according to the hole boundary method, and the hole repair was realized by B-spline surface. The green point cloud in FIGURE 15(d) is the repaired hole data. The total number of repair points for each model is 243684,226533 and 194400, respectively.

To ensure the accuracy of the comparison benchmark, the model designed by SolidWorks was used as the verification benchmark. The actual distance between each repaired feature point data and the benchmark was calculated. And the average calculation result of 10 sets of data for each model was used as the basis for evaluation. FIGURE 16 shows the deviation between the repaired hole data and the real model. Among them, the average errors of the three models are 0.093, 0.065 and 0.152 mm, respectively. It can be seen that some holes have a large deviation, which is due to the fact that the edge of the scanned hemisphere model has a non-manifold surface. And the B-spline surface used in this algorithm has a poor fitting effect for these. In addition, we also carried out hole repair experiments on three types of helmets, as shown in FIGURE 17(b). It can be seen that the algorithm proposed in this paper has good repair effect on different types of helmets.

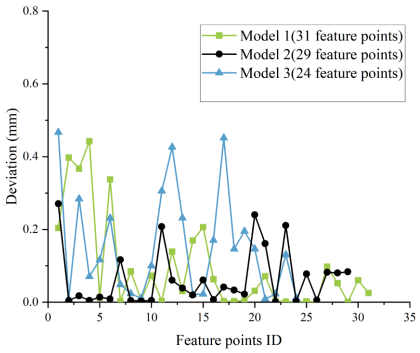


FIGURE 16. The deviation between the hole repaired by the algorithm and the actual model.

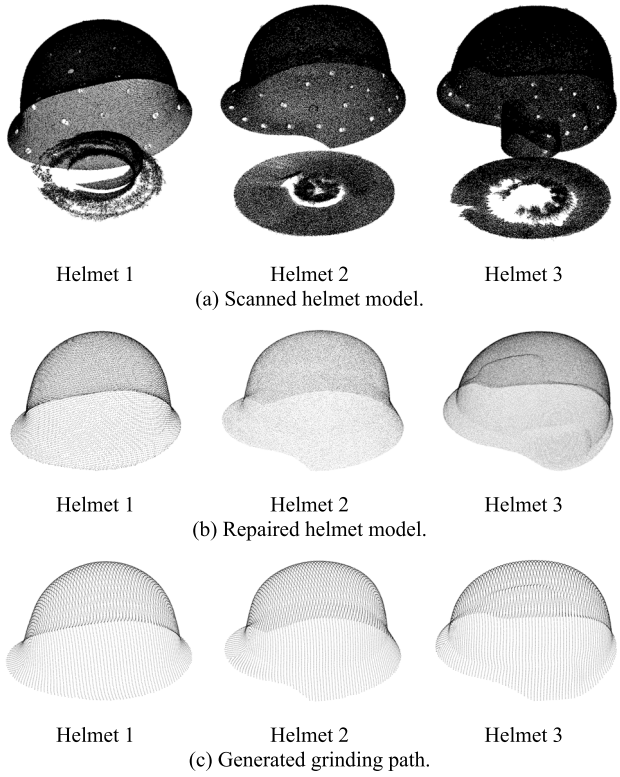


FIGURE 17. The planning effect of different styles of helmets.

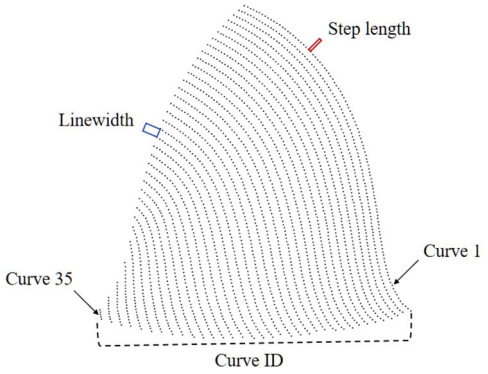
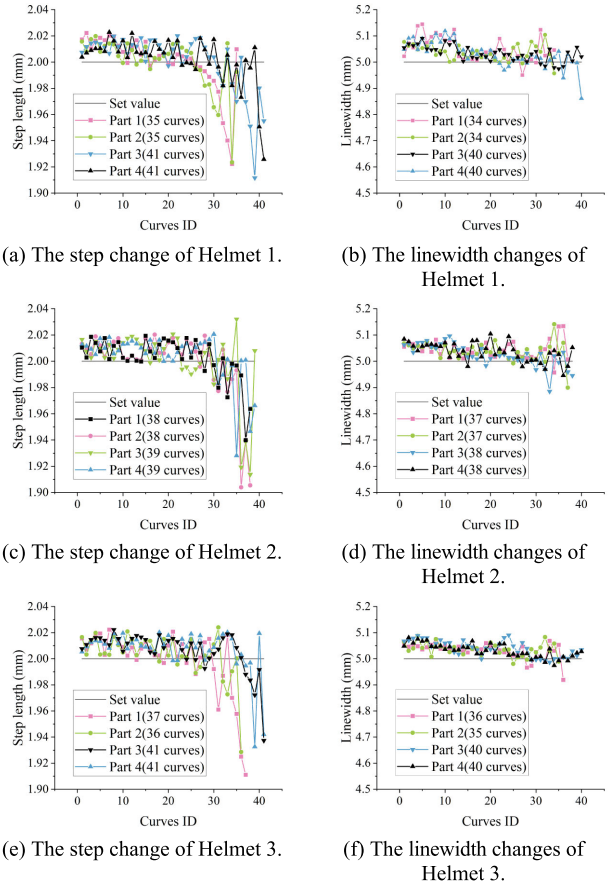


FIGURE 18. The schematic diagram of parameter setting.



(a) The step change of Helmet 1. (b) The linewidth changes of Helmet 1.
(c) The step change of Helmet 2. (d) The linewidth changes of Helmet 2.
(e) The step change of Helmet 3. (f) The linewidth changes of Helmet 3.

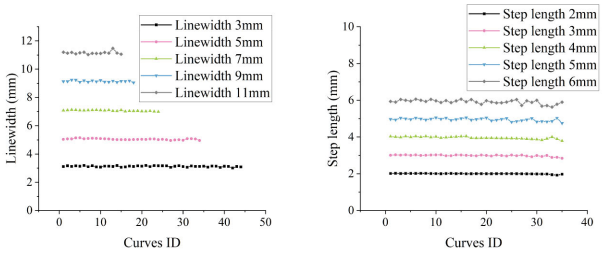
FIGURE 19. Step length and linewidth variations for different types of helmets.

B. VERIFICATION OF TRAJECTORY PLANNING ALGORITHM

To better verify the accuracy of the distance between the contact points, we compared the actual step length (inter-point spacing) on each trajectory and the row spacing between adjacent trajectories. Since the curvature of the helmet model was very small, the former used the Euclidean distance between the two points as an approximation, and the latter used the shortest distance from the point to the curve as an approximation.

We planned the same model for 10 times, and took the average value of the results as the evaluation index, as shown in FIGURE 18. Among them, the average step length of the three types of helmets was 2.0008, 2.0014 and 2.0039 mm respectively, and the average linewidth was 5.0372, 5.0344 and 5.0343 mm respectively. It can be seen that for different types of helmets, the generated contact points are evenly distributed between points and lines. And the generated grinding trajectories have good accuracy.

It can be seen from FIGURE 19 that the linewidth error on the trajectory is one order of magnitude larger than the step length error. This is due to our approximate selection



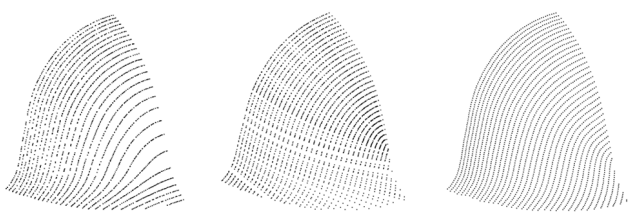
(a) Linewidth deviation results (step length 2mm). (b) Step length deviation results (linewidth 5mm).

FIGURE 20. Linewidth and step length deviation in different situations.

of the curve offset direction. When the setting value of the linewidth becomes larger, its error also becomes larger, and presents a poor deviation distribution outside 7mm, as shown in FIGURE 20(a). However, in the actual grinding, the range of linewidth is generally not more than 5mm. The planning algorithm proposed in this paper has good accuracy and robustness within 5mm. In addition, the step length has a large deviation in the second half of the Curve ID. This is because in order to ensure that the trajectory passes through the beginning and end points of each curve, the step length is individually corrected on each curve according to (14). And the margin is evenly distributed to each contact point. Because the total length and contact points of the curve with larger Curve ID are less, it leads to a larger step length deviation. FIGURE 20(b) shows the deviation distribution of different step length in the case of a constant linewidth of 5mm. It can be seen that there is a significant deviation between 5mm and 6mm. This is due to the increasing approximation error between the two-point Euclidean distance and the real value as the step length increases.

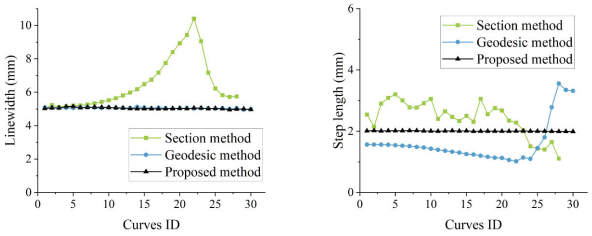
C. COMPARISON AND ANALYSIS

This paper uses the section method, the geodesic method and the proposed algorithm to carry out the trajectory planning experiment. The row width and step length were set to 5mm and 2mm respectively. FIGURE 21, FIGURE 22 and TABLE 2 show the experimental results of the three algorithms on Part 1. It can be seen that the trajectory planning result obtained by the section method is the worst, which has a large deviation in the step length and linewidth. The geodesic method has a small deviation in the linewidth, but the deviation in the step length is very large, and the running



(a) Section method. (b) Geodesic method. (c) Proposed method.

FIGURE 21. Trajectory planning results for different methods on Part 1.



(a) Linewidth comparison. (b) Step length comparison.

FIGURE 22. Linewidth and step length comparison for different methods on Part 1.

TABLE 2. Comparison of planning results for different methods.

Methods	Section method	Geodesic method	Proposed method
maximum linewidth deviation (mm)	5.404	0.164	0.121
maximum step length deviation (mm)	1.202	1.556	0.087
running Time (s)	0.183	1.229	0.141

time is very long. In contrast, the proposed algorithm has a small deviation in step length and linewidth, the contact points on the trajectory are evenly distributed, and the running time is the smallest, thus showcasing good robustness.

TABLE 3 shows the overall running time of the proposed algorithm. It can be seen that most time is spent on the repair of scanning data, the trajectory planning and preprocessing stage takes less time. This is due to the point cloud processing object has the disorder distribution and large quantity in the repair phase, resulting in more complex calculations. The overall running time of the algorithm is about 10.5 s, which can meet the actual needs of the project.

TABLE 3. Overall running time of algorithm.

	Processing Steps	Running Time (s)
preprocessing	down-sampling (data compression)	0.165
	cluster segmentation	0.688
scan data repair	feature point removal	2.253
	smoothing	1.523
	hole repair	4.571
	down-sampling (uniform density)	0.012
trajectory planning	model posture tuning	0.017
	surface reconstruction and sharding	0.637
	trajectory generation	0.566
	tool posture optimization	0.091
total time		10.523

V. DISCUSSION

The surface of the helmet model is a typical three-dimensional manifold, and the repair of point cloud holes needs to consider its local surface features. By fitting the data around the hole with the B-spline surface, the local features of the hole domain can be effectively restored, and then the point

cloud data can be restored. As shown in FIGURE 16, when using the repair algorithm proposed in this paper to repair the hemisphere model, most of the repaired point clouds show low errors, and a few point clouds have large errors. This is due to the fact that some feature points are pasted on the edge of the model, which leads to the addition of local features of non-manifold regions in the fitting process, which affects the fitting ability of B-spline surfaces. Therefore, the point cloud hole repair algorithm proposed in this paper has a limitation: when the model hole is on a non-manifold surface, the B-spline surface fitting ability is weak and the repair ability is reduced.

In the process of planning the trajectory of the repaired model, the selection of linewidth will significantly affect the trajectory quality. When the linewidth in grinding is less than 7mm, the generated contact points show good uniformity in the spacing between rows and points. However, due to the approximate selection of the offset direction in the curve offset process, a large deviation will occur when the linewidth is greater than 7mm. And it will gradually increase with the increase of the linewidth, which is inevitable. Therefore, the trajectory generation algorithm proposed in this paper shows good accuracy and robustness only when the linewidth is less than 7mm. However, the linewidth of 7mm can meet the use of helmet grinding in this paper. For larger linewidth requirements, it is necessary to further study the trajectory generation algorithm.

VI. CONCLUSION

(1) According to the experimental results, the repair algorithm proposed in this paper has good repair ability for manifold surfaces. As far as the holes of artificial feature points are concerned in the helmet scanning model, the algorithm proposed in this paper can achieve good model repair.

(2) The trajectory generation algorithm proposed in this paper can achieve uniform distribution of grinding contact points in linewidth and step length. Compared with other trajectory planning algorithms, the proposed algorithm has smaller deviation in linewidth and step length and shorter running time. Through the generation of trajectories on different types of helmet models, the trajectories are compared and analyzed, which reflects the accuracy and robustness of the algorithm.

(3) The algorithm and operation process proposed in this paper can realize the generation of helmet grinding trajectory based on 3D scanner. However, the ability of repairing holes on non-manifold surfaces and generating trajectories with linewidth greater than 7mm is not so satisfactory, showing a large deviation. In the follow-up study, it is necessary to further study the trajectory generation of non-manifold surfaces and large linewidth. This is of great significance for the large-scale application of 3D visual processing technology in industry.

REFERENCES

[1] B. Zhang, S. Wu, D. Wang, S. Yang, F. Jiang, and C. Li, "A review of surface quality control technology for robotic abrasive belt grinding of aero-engine blades," *Measurement*, vol. 220, Oct. 2023, Art. no. 113381, doi: 10.1016/j.measurement.2023.113381.

[2] X. Liang, Z. Cai, C. Zeng, Z. Mu, Z. Li, F. Yang, T. Chen, S. Dong, C. Deng, and S. Niu, "A robotic polishing trajectory planning method for TBCs of aero-engine turbine blade using measured point cloud," *Ind. Robot: Int. J. Robot. Res. Appl.*, vol. 50, no. 2, pp. 275–286, Feb. 2023, doi: 10.1108/IR-05-2022-0141.

[3] H. Liang and J. Qiao, "Analysis of current situation, demand and development trend of casting grinding technology," *Micromachines*, vol. 13, no. 10, p. 1577, Sep. 2022, doi: 10.3390/mi13101577.

[4] J. Li, L. Zou, G. Luo, W. Wang, and C. Lv, "Enhancement and evaluation in path accuracy of industrial robot for complex surface grinding," *Robot. Comput.-Integr. Manuf.*, vol. 81, Jun. 2023, Art. no. 102521, doi: 10.1016/j.rcim.2022.102521.

[5] D. Zhu, X. Feng, X. Xu, Z. Yang, W. Li, S. Yan, and H. Ding, "Robotic grinding of complex components: A step towards efficient and intelligent machining—Challenges, solutions, and applications," *Robot. Comput.-Integr. Manuf.*, vol. 65, Oct. 2020, Art. no. 101908, doi: 10.1016/j.rcim.2019.101908.

[6] Q. Yu, G. Wang, X. Hua, S. Zhang, L. Song, J. Zhang, and K. Chen, "Base position optimization for mobile painting robot manipulators with multiple constraints," *Robot. Comput.-Integr. Manuf.*, vol. 54, pp. 56–64, Dec. 2018, doi: 10.1016/j.rcim.2018.05.007.

[7] Z. Lu, J. Fan, Z. Hou, S. Deng, C. Zhou, and F. Jing, "Automatic 3D seam extraction method for welding robot based on monocular structured light," *IEEE Sensors J.*, vol. 21, no. 14, pp. 16359–16370, Jul. 2021, doi: 10.1109/JSEN.2021.3076341.

[8] H. Lyu, Y. Liu, J.-Y. Guo, H.-M. Zhang, and Z.-X. Li, "Tool-path generation for industrial robotic surface-based application," *Adv. Manuf.*, vol. 7, no. 1, pp. 64–72, Mar. 2019, doi: 10.1007/s40436-018-00246-x.

[9] H. Feng, X. Ren, L. Li, X. Zhang, H. Chen, Z. Chai, and X. Chen, "A novel feature-guided trajectory generation method based on point cloud for robotic grinding of freeform welds," *Int. J. Adv. Manuf. Technol.*, vol. 115, nos. 5–6, pp. 1763–1781, Jul. 2021, doi: 10.1007/s00170-021-07095-2.

[10] L. Lv, Z. Deng, T. Liu, Z. Li, and W. Liu, "Intelligent technology in grinding process driven by data: A review," *J. Manuf. Processes*, vol. 58, pp. 1039–1051, Oct. 2020, doi: 10.1016/j.jmapro.2020.09.018.

[11] D. Gleeson, S. Jakobsson, R. Salman, F. Ekstedt, N. Sandgren, F. Edelvik, J. S. Carlson, and B. Lennartson, "Generating optimized trajectories for robotic spray painting," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 3, pp. 1380–1391, Jul. 2022, doi: 10.1109/TASE.2022.3156803.

[12] G. Zhu, X. Zeng, Z. Gao, Z. Gong, W. Duangmu, Y. Zeng, and C. Lu, "Study on vibration stability of aircraft engine blades polished by robot controlled pneumatic grinding wheel," *J. Manuf. Processes*, vol. 99, pp. 636–651, Aug. 2023, doi: 10.1016/j.jmapro.2023.05.090.

[13] X. Wang, X. Zhang, X. Ren, L. Li, H. Feng, Y. He, H. Chen, and X. Chen, "Point cloud 3D parent surface reconstruction and weld seam feature extraction for robotic grinding path planning," *Int. J. Adv. Manuf. Technol.*, vol. 107, nos. 1–2, pp. 827–841, Mar. 2020, doi: 10.1007/s00170-020-04947-1.

[14] L. Yang, Y. Liu, J. Peng, and Z. Liang, "A novel system for off-line 3D seam extraction and path planning based on point cloud segmentation for arc welding robot," *Robot. Comput.-Integr. Manuf.*, vol. 64, Aug. 2020, Art. no. 101929, doi: 10.1016/j.rcim.2019.101929.

[15] W. Cai, J. Wang, P. Jiang, L. Cao, G. Mi, and Q. Zhou, "Application of sensing techniques and artificial intelligence-based methods to laser welding real-time monitoring: A critical review of recent literature," *J. Manuf. Syst.*, vol. 57, pp. 1–18, Oct. 2020, doi: 10.1016/j.jmsy.2020.07.021.

[16] W. Zhikun, Y. Jincheng, Y. Ling, Z. Sumin, C. Yehao, L. Caixing, and T. Xuhong, "Improved hole repairing algorithm for livestock point clouds based on cubic B-spline for region defining," *Measurement*, vol. 190, Feb. 2022, Art. no. 110668, doi: 10.1016/j.measurement.2021.110668.

[17] C. Zhang, H. Zhou, and J. Duan, "A method for identifying and repairing holes on the surface of unorganized point cloud," *Measurement*, vol. 210, Mar. 2023, Art. no. 112575, doi: 10.1016/j.measurement.2023.112575.

[18] P. Zhou, R. Peng, M. Xu, V. Wu, and D. Navarro-Alarcon, "Path planning with automatic seam extraction over point cloud models for robotic arc welding," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 5002–5009, Jul. 2021, doi: 10.1109/LRA.2021.3070828.

[19] L. Li, C. Li, Y. Tang, and Y. Du, "An integrated approach of reverse engineering aided remanufacturing process for Worn components," *Robot. Comput.-Integr. Manuf.*, vol. 48, pp. 39–50, Dec. 2017, doi: 10.1016/j.rcim.2017.02.004.
[20] X.-F. Han, J. S. Jin, M.-J. Wang, W. Jiang, L. Gao, and L. Xiao, "A review of algorithms for filtering the 3D point cloud," *Signal Process., Image Commun.*, vol. 57, pp. 103–112, Sep. 2017, doi: 10.1016/j.image.2017.05.009.
[21] A. Agathos, P. Azariadis, and S. Kyratzi, "Elliptic Gabriel taubin smoothing of point clouds," *Comput. Graph.*, vol. 106, pp. 20–32, Aug. 2022, doi: 10.1016/j.cag.2022.05.009.
[22] W.-L. Li, H. Xie, G. Zhang, Q.-D. Li, and Z.-P. Yin, "Adaptive bilateral smoothing for a point-sampled blade surface," *IEEE/ASME Trans. Mechatronics*, vol. 21, no. 6, pp. 2805–2816, Dec. 2016, doi: 10.1109/TMECH.2016.2581808.
[23] Y. Gan, Z. Sun, Z. Chen, X. Zhang, and Y. Liu, "Enhancement of the material point method using B-spline basis functions," *Int. J. Numer. Methods Eng.*, vol. 113, no. 3, pp. 411–431, Jan. 2018, doi: 10.1002/nme.5620.
[24] Y. Geng, M. Lai, X. Tian, X. Xu, Y. Jiang, and Y. Zhang, "A novel seam extraction and path planning method for robotic welding of medium-thickness plate structural parts based on 3D vision," *Robot. Comput.-Integr. Manuf.*, vol. 79, Feb. 2023, Art. no. 102433, doi: 10.1016/j.rcim.2022.102433.
[25] G. Fu, B. Sheng, Y. Huang, R. Luo, G. Chen, and G. Sheng, "An adaptive sampling method for STL free-form surfaces based on the quasi-gauss curvature grid," *Int. J. Comput. Integr. Manuf.*, vol. 36, no. 11, pp. 1716–1733, Nov. 2023, doi: 10.1080/0951192X.2023.2204470.
[26] A. Majeed, A. R. Mt Piah, Z. R. Yahya, J. Y. Abdullah, and M. Rafique, "Construction of occipital bone fracture using B-spline curves," *Comput. Appl. Math.*, vol. 37, no. 3, pp. 2877–2896, Jul. 2018, doi: 10.1007/s40314-017-0487-0.
[27] B. Ma, X. Yao, T. An, B. Dong, and Y. Li, "Model free position-force control of environmental constrained reconfigurable manipulators based on adaptive dynamic programming," *Artif. Intell. Rev.*, vol. 56, no. 3, pp. 3143–3171, Oct. 2023, doi: 10.1007/s10462-023-10600-6.

YUXIANG MENG is currently pursuing the master's degree with the Laboratory of Special Precision Processing and Measurement and Control Instruments, Dalian Polytechnic University, under the supervision of Dr. Tong Qiang. His research interests include robot control and 3D vision.

YU JIANG is currently pursuing the master's degree with the Laboratory of Special Precision Processing and Measurement and Control Instruments, Dalian Polytechnic University, under the supervision of Dr. Tong Qiang. His research interests include additive manufacturing and laser selective processing.

YI LI is currently pursuing the master's degree with the Laboratory of Special Precision Processing and Measurement and Control Instruments, Dalian Polytechnic University, under the supervision of Dr. Tong Qiang. His research interests include finite element simulation and mechanical structure design.

GUIBING PANG received the Ph.D. degree in machinery manufacturing and automation from the Dalian University of Technology, in 2004. He is currently a Professor with the Institute of Special Precision Machining and Measurement and Control Instruments, Dalian Polytechnic University. His research interests include special precision machining, intelligent manufacturing technology, and precision instrumentation technology.

QIANG TONG received the Ph.D. degree in material processing engineering from the Dalian University of Technology, in 2021. He is currently a Lecturer with the Institute of Special Precision Machining and Measurement and Control Instruments, Dalian Polytechnic University. At present, he is engaged in advanced manufacturing technology and equipment, polymer 3D/4D printing technology, industrial robot processing technology, and other research.