

Article

Model-Free Path Planning for Complex Grooves on Spherical Workpieces Based on 3D Point Clouds

Zhongsheng Zhai, Aoxing Yi, Zhen Zeng * , Xikang Xiao and Ndifreke Offiong

Hubei Key Laboratory of Modern Manufacturing Quantity Engineering, School of Mechanical Engineering, Hubei University of Technology, Wuhan 430068, China; zs.zhai@hbut.edu.cn (Z.Z.); 102310039@hbut.edu.cn (A.Y.); 20211064@hbut.edu.cn (X.X.); offiongndifrekeanietie@hbut.edu.cn (N.O.)

* Correspondence: zengzhen@hbut.edu.cn

Abstract

To address the precision and motion-smoothing challenges in path planning for spherical workpieces without Computer-Aided Design (CAD) models, this paper proposes a robust point-cloud-driven framework. Conventional Principal Component Analysis (PCA) alignment suffers from centroid shift errors due to asymmetric data loss from light-absorbing surface features. To solve this, a RANSAC-compensated hybrid PCA algorithm is developed to decouple position and orientation estimation, ensuring stable coordinate alignment despite incomplete data. Furthermore, to resolve the geometric collapse and kinematic jitter caused by traditional planar slicing in high-curvature polar regions, a spherical latitudinal equiangular conical slicing algorithm is introduced. By aligning the slicing planes with the sphere's radial geometry, the method preserves topological accuracy while maintaining an optimal point density for smooth robotic execution. Experimental results on rubber ball groove processing demonstrate a repeat positioning accuracy of 0.09 mm and a feature coverage of 95.21%. This research provides a scientifically rigorous and computationally efficient solution for the automated processing of complex spherical surfaces.

Keywords: spherical workpieces; point cloud; path planning; RANSAC-compensated hybrid PCA; spherical conical slicing

1. Introduction

The manufacturing of rubber balls, as a representative of flexible spherical workpieces, has traditionally relied on labor-intensive manual inking [1], which suffers from low consistency and high rejection rates, and the ink application environment poses a potential threat to workers' health [2]. While conventional automated solutions rely on rigid Computer-Aided Design (CAD) models, they fail to accommodate the geometric uncertainties caused by workpiece deformation and material variability in real-world environments. Therefore, establishing a model-free path planning framework based on three-dimensional (3D) point cloud feedback is not only a practical necessity but also a scientific challenge involving high-fidelity geometric reconstruction and non-linear motion optimization.

With the rapid development of industrial automation and robotics technology [3], industrial robots have been extensively deployed in fields such as welding and spraying [4,5]. However, in rubber ball manufacturing, traditional “teach–playback” [6] modes and offline programming methods face considerable challenges, including the complex geometry of grooves and dynamic rotation of spherical workpieces. Requiring point-by-point teaching or intricate path programming, these approaches demand high operator expertise and are



Academic Editors: Yutaka Ishibashi and Paolo Renna

Received: 17 December 2025

Revised: 27 January 2026

Accepted: 3 February 2026

Published: 5 February 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

time-consuming, making direct adaptation impractical. Thus, developing a fully automated ink application system tailored to rubber ball surfaces is critical for improving product quality and green manufacturing.

Robot machining path planning is essential for automated processing. Researchers have conducted extensive investigations into robot path planning based on diverse processing scenarios and technical routes. Song et al. [7] addressed the limitations of complex curved surface grinding path planning for grinding robots, proposing a novel path planning and post-processing method based on NX 12.0 and Tecnomatix software. Zhou et al. [8] focused on the grinding of nuclear reactor coolant pump casings, introducing a multi-objective optimization-driven robotic disc grinding path planning approach. Lei et al. [9] developed a new 3D weld seam path extraction method utilizing a stereo structured light sensor, which can effectively accomplish 3D path teaching prior to welding operations. Guo et al. [10] proposed a skeleton contour segmentation-based hybrid path planning for robotic additive manufacturing, validated in industrial cases. Most of these methods rely on workpiece CAD models, which fail when models are missing or actual surfaces deviate from CAD data.

In recent years, 3D point cloud-driven path planning technology has provided new approaches for processing complex surfaces without CAD models [11]. Point cloud data can clearly reveal the surface features of target workpieces, prompting many researchers to adopt it for robot path planning [12–18]. Wang et al. [19] proposed an IEIPPP method based on 3D point cloud reconstruction, which can plan collision-free paths without prior knowledge of the path or obstacles. Xu et al. [20] introduced a welding path planning method that combines point cloud and deep learning, addressing issues with high redundancy and unevenly dense point clouds, thus enabling automatic welding.

However, when point cloud technology is directly applied to robotic path generation, coordinate calibration emerges as a critical bottleneck. Conventional PCA-based calibration [21] fundamentally assumes data integrity for accurate centroid calculation. In the presence of asymmetric data loss—such as light-absorbing black logos on rubber balls—the computed centroid undergoes a stochastic drift from the true geometric center, leading to systematic calibration failure. To resolve this, this study proposes a decoupled coordinate alignment framework. By leveraging the robustness of Random Sample Consensus (RANSAC) [22] to enforce a spherical geometric constraint, we decouple the origin localization from the orientation estimation. This hybrid approach effectively mitigates the adverse impact of surface incompleteness, transforming a statistically sensitive calculation into a geometry-constrained optimization.

For the extraction of target point clouds, conventional methods relying on local differential features (e.g., normal vectors or curvature) suffer from high computational complexity and noise sensitivity [23,24]. To address these issues, this study proposes a robust groove extraction methodology based on global spherical topology. The core innovation lies in the transition from local feature computation to radial distance mapping. By leveraging the Euclidean distance from points to the RANSAC-fitted sphere center, we establish a deterministic threshold for groove identification. This approach effectively circumvents the instability of local feature extraction in unstructured environments, significantly enhancing both computational efficiency and the algorithm's robustness against noise interference.

Existing slicing algorithms, such as those proposed by Raible et al. [25] and Liu et al. [26], have demonstrated efficacy in grinding and welding. However, these methods typically employ planar slicing, which faces inherent theoretical challenges when handling high-curvature spherical patterns. Specifically, as the slicing plane approaches the poles, the varying angle between the plane and the surface normal induces “geometric collapse” and topological distortion, resulting in kinematic jitter in robot motion. To this end, this

study introduces an innovative equiangular conical slicing algorithm tailored for spherical manifolds. By ensuring a consistent geometric relationship between the slicing planes and the spherical center, the algorithm transforms 3D point cloud data into ordered, executable motion paths while maintaining topological fidelity and kinematic smoothness.

In summary, the core contribution of this paper lies in proposing and validating a complete set of fully automatic ink application path planning solutions for groove features on rubber ball surfaces based on 3D point cloud processing.

Specifically, the key innovations of the proposed method are as follows:

- Improved PCA Calibration Method Incorporating RANSAC;
- A groove extraction method based on the spherical center distance is proposed;
- A spherical latitudinal equiangular conical slicing method is proposed.

Furthermore, to achieve high production efficiency of workpieces, this study implements a multi-DOF (degree-of-freedom) collaborative control architecture based on the EtherCAT (Ethernet for Control Automation Technology) bus. By synchronizing the robot's spatial trajectory planning with the continuous motion of a high-precision turntable, the framework realizes a dynamic processing mode. It provides key technical support for meeting the stringent cycle time requirements of industrial mass production.

2. Materials and Methods

Before elaborating on each algorithm module, this section first presents the overall technical framework of the proposed automatic path planning method in Figure 1. As illustrated in the figure, the framework is a systematic, end-to-end processing flow that starts with the acquisition of 3D point cloud data and ends with the generation of smooth trajectories executable by robots. Subsequent subsections will elaborate on the principles, algorithms, and implementations of each step in this sequence.

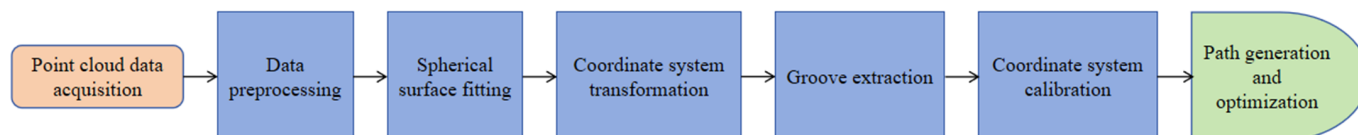


Figure 1. Overall flow chart of the automatic path planning method.

2.1. Point Cloud Data Processing Methods

This section describes the processing of rubber ball point clouds scanned by a 3D sensor to obtain coordinate information recognizable by the robotic arm. It aims to provide accurate geometric data support for the fully automated ink application path planning of rubber ball surface grooves, serving as the foundation for subsequent path planning. The algorithm process of point cloud data processing is shown in Figure 2.

2.1.1. Measurement and Preprocessing of Point Cloud Data on Rubber Ball Surfaces

The scanner used in this study is the AtlaScan developed by Wuhan Zhongguan Automation (a subsidiary of Hexagon Manufacturing Intelligence), which has strong adaptability to ambient light fluctuations and surface reflections, with key specifications: a maximum resolution of 0.01 mm, a measurement rate of 5.22 million measurements per second, a boundary scanning accuracy of 0.05 mm, and a maximum scanning area of $720 \times 640 \text{ mm}^2$. It provides crucial support for the high-precision and high-efficiency acquisition of point clouds on the surface of rubber balls. The original point cloud, obtained through high-density 3D sensor scanning, is vast, with nearly 500,000 points. Direct processing of such a large point cloud is inefficient and often results in noise points and outliers, as illustrated in Figure 3.

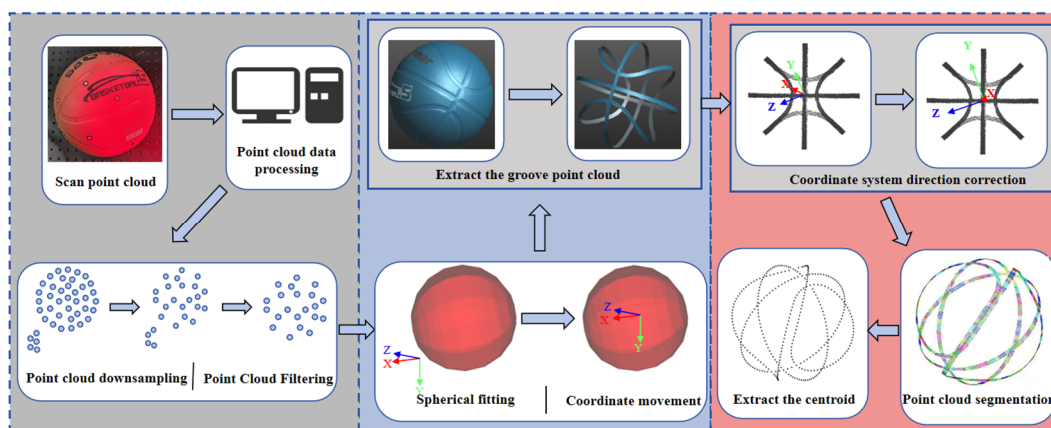


Figure 2. Flow chart of point cloud data processing algorithm.

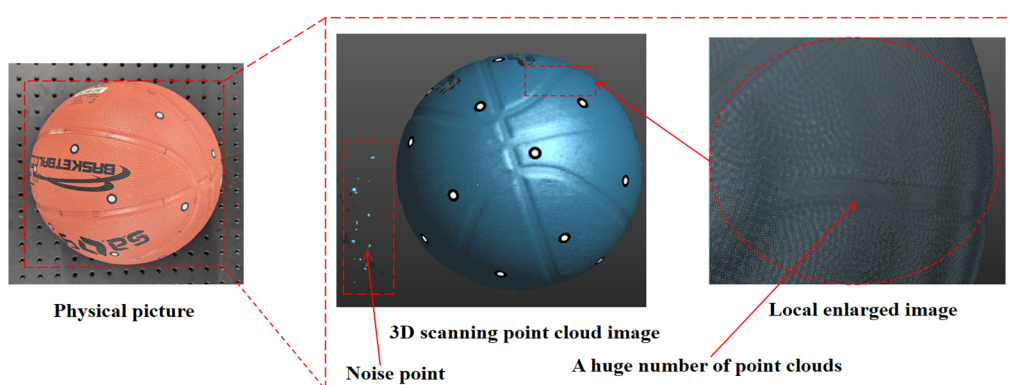


Figure 3. Extracted rubber ball point cloud.

Thus, point cloud downsampling is indispensable to reduce point density and enhance processing efficiency. This study employs voxel downsampling [27], which effectively reduces the number of data points while preserving point cloud details and statistical filtering to eliminate outliers in the point cloud [28].

First, the voxel leaf size is set. Then, each voxel is traversed, and the geometric centroid of all points inside it is calculated as a representative point, as shown in Appendix B Equations (A1) and (A2).

After downsampling, statistical filtering is applied to eliminate outliers. Based on the statistical characteristics of local point cloud regions, this method assumes that the neighborhood distance distribution of normal points follows a Gaussian distribution. It identifies and removes noise points deviating from the main distribution by calculating the distance statistics from each point to its neighboring points. Considering the point cloud inhomogeneity, a radius search is adopted herein: for each point p_i in the point cloud, all points within a spherical neighborhood with a radius of 5 mm (targeting 50 neighboring points) are searched, and the mean value μ_i and standard deviation σ_i of the distances are calculated:

$$\mu_i = \frac{1}{m} \sum_{j=1}^m \|p_i - p_j\| \quad (1)$$

$$\sigma_i = \sqrt{\frac{1}{m} \sum_{j=1}^m (\|p_i - p_j\| - \mu_i)^2} \quad (2)$$

The parameter m denotes the number of domain points, and the threshold condition $\mu_i + k \cdot \sigma_i > T$ has been established. Under the assumption of a normal distribution, this

study adopts $k = 2$ and $T = 5$ mm. Any point that satisfies the threshold condition is identified as an outlier and subsequently removed from the point cloud.

2.1.2. RANSAC-Based Sphere Fitting

In 3D scanned point cloud processing, the acquired workpiece coordinates are in the scanner's native coordinate system. This system deviates from the geometric center of the target object. To facilitate subsequent spherical point cloud processing and groove point cloud extraction, it is essential to transform the point cloud data from the original coordinates to the spherical center-based coordinate system.

For the point cloud data of rubber balls, the existing PCA algorithm directly calculates the point cloud centroid as the spherical center, but its error is significant when disturbed by surface defects; the least squares fitting method, on the other hand, is sensitive to outliers. The RANSAC iterative algorithm is not affected by surface defects [29] and can estimate optimal model parameters from data containing noise and outliers, making it suitable for scenarios with scanning noise, such as rubber ball point clouds. However, this algorithm has an obvious drawback: reasonable thresholds and the number of iterations need to be selected to achieve good fitting accuracy. To this end, a coordinate system migration method based on RANSAC spherical fitting is proposed.

First, the RANSAC algorithm robustly estimates the center of the sphere, incorporating the radius constraint of the rubber ball. According to International Basketball Federation (FIBA) regulations [30], the circumference of the No.5 ball is 685–700 mm. For details, refer to Table A1 in Appendix A. This narrows the parameter search space, moves the center of the sphere to (0, 0, 0), derives the translation matrix, and then performs decentering.

In the three-dimensional space, the spherical equation is shown in Appendix B Equation (A3). RANSAC performs the following steps iteratively:

(1) Random sampling is performed, four non-coplanar points $\{p_1, p_2, p_3, p_4\}$ are randomly selected. The candidate sphere parameters are obtained by solving the system of equations, and the sphere center coordinates and radius are calculated from these four points. An overdetermined system of equations is then constructed, as shown in Appendix B Equation (A4), to solve for the sphere center coordinates (x_0, y_0, z_0) and radius r .

(2) The inliers are screened out. For each point $p_i \in P$ in the point cloud, the distance from each point to the spherical surface is calculated:

$$d_i = \left| \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2} - r \right| \quad (3)$$

If $d_i \leq \text{threshold}$ and $r \in [r_{\min}, r_{\max}]$, it is determined as an interior point, and the proportion of interior points is calculated. The threshold is taken as 1.5 mm in this study. According to the regulations of FIBA, the radius constraint range of a size 5 ball is [109.07, 111.46] mm.

(3) Model evaluation is conducted. Iterations are performed on steps 2 and 3, with the sphere parameter exhibiting the highest in-point rate selected as the final solution. The radius of the rubber ball ranges from 109.07 mm to 111.46 mm. If the radius obtained in step 2 falls outside this range, step 1 should be repeated.

The detailed comparative experiments on the influence of RANSAC iteration numbers on fitting performance and the parameter optimization process are presented in Table A2.

After fitting the sphere and obtaining its center, the next step is coordinate system transformation. It is necessary to translate the original coordinate system, which deviates from the geometric center of the target object, to the sphere center, thus taking the sphere center as the new origin. For each point $P(x_i, y_i, z_i)$ in the original point cloud, its coordinate after transformation is denoted as $P' = T \cdot P = (x_i - x_0, y_i - y_0, z_i - z_0)$.

2.1.3. Groove Point Cloud Extraction Based on Sphere Center Distance

To extract the groove point cloud, this study proposes a groove point cloud extraction method based on the distance to the spherical center. By calculating the Euclidean distance from each point on the spherical surface to the center, points with a distance smaller than a specific threshold relative to the spherical radius are selected as the groove region. Compared with traditional segmentation methods based on normal vector direction or curvature, this approach avoids complex normal estimation and curvature calculation, relying solely on linear operations of geometric distances, which significantly improves computational efficiency.

To verify the geometric characteristics of the groove structure, a depth map of the rubber ball surface is further generated, mapping the difference in distances to the spherical center into color gradients. As shown in Figure 4, the left panel displays the overall depth map of the rubber ball: the convex surfaces are yellow, and the concave areas (grooves) are colored blue. The detailed sections, as illustrated in the cross-sectional view on the right side of the figure, show that the yellow part of the spherical surface is farthest from the center, with a maximum distance of 110.77 mm, while the blue area is closest to the center, with a minimum distance of 108.09 mm. This method provides a more intuitive perspective and a range of selectable thresholds for the subsequent selection of concave point cloud data.

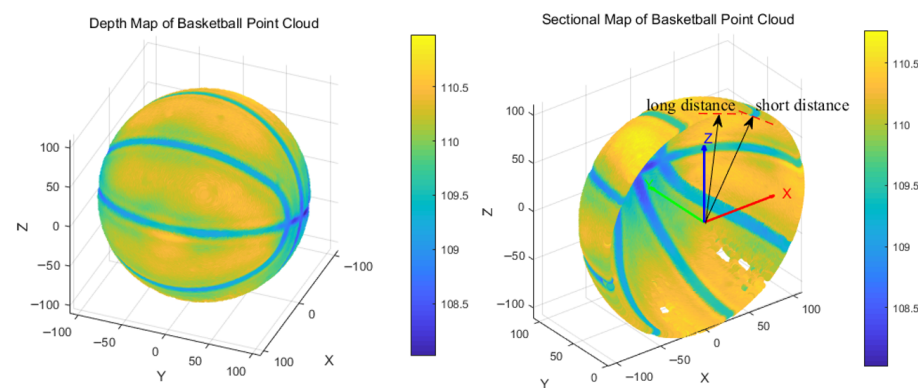


Figure 4. Depth map of the rubber ball surface.

The groove point cloud extraction method based on sphere center distance is derived from the fact that the distance d from points in the concave region to the sphere center is significantly smaller than the sphere radius R . By setting a reasonable threshold δ , points that satisfy the condition $R - d > \delta$ are selected as groove candidate points. First, for each point $P_i(x_i, y_i, z_i)$ in the point cloud that has been registered to the sphere-center coordinate system, the 3D Euclidean distance from the point to the sphere center is calculated point by point, as follows:

$$d_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2} \quad (4)$$

Second, a groove depth threshold is set based on the known sphere radius to screen groove candidate points. A threshold value δ (set to 0.5 mm herein) is defined, and points satisfying the condition $R - d_i > \delta$ are retained as the candidate point set $\rho_{groove} = \{p_i \in \rho | R - d_i > \delta\}$, thus forming a coarsely extracted groove candidate point set. Subsequently, outlier noise points are removed via Euclidean clustering segmentation, and the largest connected region is retained as the final groove point cloud. This method directly quantifies the surface concave features through geometric distance, avoiding the complex normal estimation and curvature calculation involved in traditional methods, and significantly improving the computational efficiency.

2.1.4. Sphere Coordinate System Calibration Algorithm Based on RANSAC-PCA

Following the point cloud preprocessing, a precise coordinate calibration between the scanning frame and the sphere's geometric frame is essential prior to groove segmentation. To address the inherent instability of traditional centroid-based alignment, this study develops a decoupled calibration strategy integrating RANSAC with PCA. Below are the specific steps of the algorithm for aligning the point cloud data.

Prior to this step, the sphere center coordinate $\bar{q}$ was obtained through sphere fitting using the RANSAC algorithm, and the decentralized point cloud dataset $P' = \{P'_1, P'_2, P'_3, P'_4, \dots, P'_n\}$ was generated accordingly. Therefore, the covariance matrix of dataset P' is calculated as follows:

$$C = \frac{1}{N} \sum_{i=1}^n P'_i P'^T_i \quad (5)$$

Through eigenvalue decomposition, the eigenvalues and eigenvectors of C 's covariance matrix can be expressed as:

$$C = V \Lambda V^T \quad (6)$$

Here, V is a 3×3 matrix composed of the eigenvectors of C . Λ is used to identify the main direction of the point cloud data, and it is a 3×3 diagonal matrix, and its diagonal elements are the eigenvalues of the covariance matrix, representing the data variance along each corresponding direction.

Subsequently, a new right-handed O-XYZ coordinate system is established. The centroid $\bar{q}$ is designated as the origin of the new system: the first principal component (corresponding to the largest eigenvalue) is selected as the X-axis, and the second principal component (corresponding to the second-largest eigenvalue) as the Y-axis, thereby forming the O-XY plane. The Z-axis is derived via the cross product of the X-axis and Y-axis vectors, completing the construction of the O-XYZ system. Utilizing this newly established coordinate system, a rotation matrix R_p can be computed to obtain the axis-aligned point set P'' :

$$R_p = V_p^T \quad (7)$$

$$P'' = P \times q \quad (8)$$

In this case, R_p is the rotation matrix.

2.2. Groove Point Cloud Slicing and Visualization

After repositioning the point cloud coordinate system to the origin, it is necessary to consider slicing the concave point cloud model. Point cloud slicing involves a series of parallel, equally spaced planes that are used to extract sampling points from the workpiece point cloud model, thereby determining the inking path [31].

Considering the complexity of performing horizontal slicing on the concave point cloud model of a rubber ball in 3D space, a spherical latitude equal-angle conical slicing method is proposed. This algorithm partitions the spherical point cloud into layers using cones passing through the sphere center. Leveraging the polar-angle-azimuth coordinate system of spherical geometry, the sphere is segmented into symmetrical layered regions. This technique can quickly and accurately segment the concave point cloud model of a rubber ball. The cutting diagram is shown in Figure 5, where $E_0 - E_6$ represents six conical slicing sections, with each adjacent section forming an interlayer.

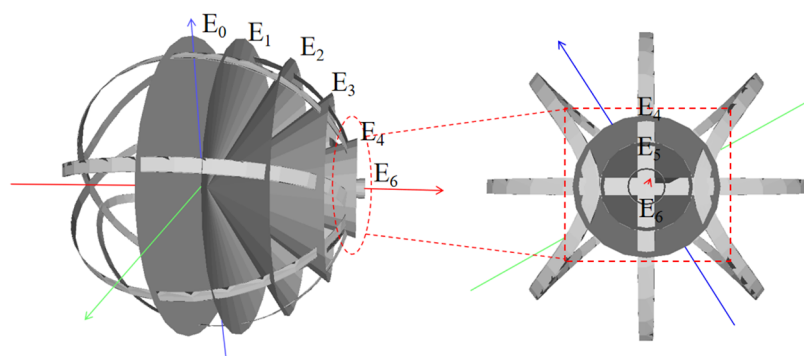


Figure 5. Schematic diagram of point cloud slicing.

Its core idea is to cut the spherical surface by constructing coaxial conical curved surfaces and disassemble the complex spherical point cloud into an ordered hierarchical structure, facilitating the subsequent extraction of texture trajectories. As shown in Figure 6.

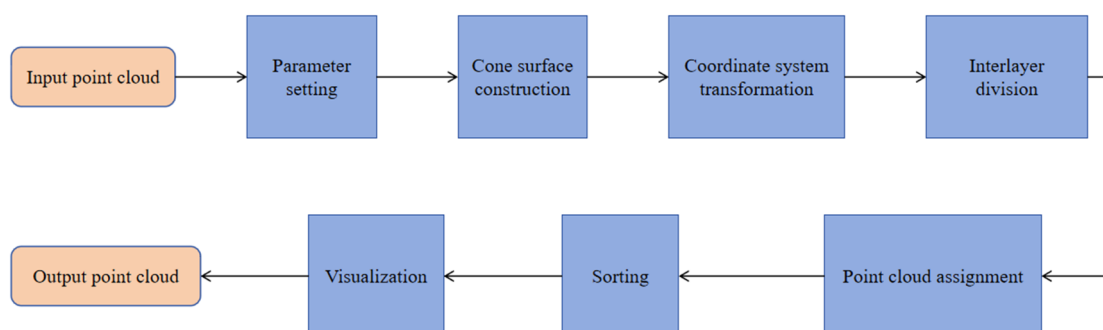


Figure 6. Slicing algorithm flow chart.

To achieve the equal-angle division of the spherical surface, an angular step size of $\Delta\theta = 3.6^\circ$ is set as the fixed increment of the half-apex angle of the conical surface. The selection of this value ensures the integrality of the slice number. Meanwhile, at the minimum slicing angle, the radius of the cross-section obtained by the intersection of the slicing plane and the spherical surface is close to the actual width of the grooves. This is an optimized result that balances the path generation accuracy and algorithm implementation efficiency.

Based on this, the range of the half-apex angle is determined to be from $\Delta\theta_{min} = 3.6^\circ$ to $\Delta\theta_{max} = 176.4^\circ$, where $\Delta\theta_{max} = 180^\circ - \Delta\theta$. Considering the symmetry of the sphere, the 49 conical surfaces of the entire sphere are symmetrically allocated: Each of the upper and lower hemispheres contains 24 conical surfaces, and the equatorial plane is a conical surface with a half-apex angle of 90° .

Thus, the half-apex angle of the k -th conical surface can be calculated using the formula $\Delta\theta = 3.6^\circ \times k (k = 1, 2, \dots, 49)$. Different values of k correspond to conical surfaces at specific positions. For each half-apex angle $\theta_k (k = 1, 2, \dots, 49)$, the analytical equation of the k -th cone is: $\ell_k : x^2 + y^2 = (z \cdot \tan\theta_k)^2$. The adjacent conical tangent planes ℓ_k and ℓ_{k+1} form an interlayer ℓ_k , where $\ell_k = \{P \in R^3 \mid \angle(OP, OZ) < \theta_{k+1}\}$. These 49 conical surfaces ultimately divide the entire sphere into 50 interlayers, including 2 polar caps and 48 intermediate interlayers, and an interlayer index $i = (1, 2, \dots, 50)$ is added. For any point on the groove, the latitude angle θ is calculated, and then the point is allocated to the corresponding interlayer based on the value of θ .

After completing the point cloud distribution, it is necessary to segment the internal layers and connect the layers across. Since the original grooves consist of multiple curves, a clustering algorithm is used to segment the point clouds within each layer based on their

continuity. The point clouds belonging to different curves are separated and indexed, with the index sequence arranged in a circular pattern from layer 1 to 50 and then from 50 to 1.

Before visualization, the concave point cloud model was segmented into conical sections along the X-axis, with each slice stored in the set S . Each slice was indexed according to its cutting sequence and assigned a unique RGB (Red, Green, Blue) color to ensure color uniqueness and consistency.

Then, based on the number of independent point clouds in each layer, each segment of the point cloud was stored in the set E , and the cutting sequence index was assigned to each segment.

2.3. Robot Trajectory Planning

2.3.1. Coordinate Relationship Calibration Between Point Cloud and Ink Application Robot

The aligned point cloud coordinate system differs from the robotic arm's coordinate system, as shown in Figure 7.

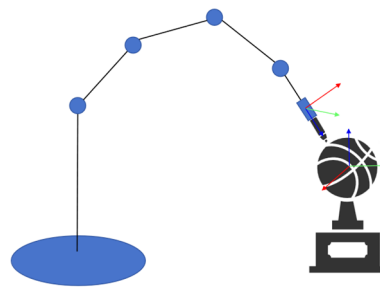


Figure 7. Position relationship of coordinates between rubber ball point cloud and ink application robot.

In robotic arm operation scenarios, a unified spatial reference frame must be established to enable the robotic arm to perform precise operations on the sphere based on its own coordinate system. In this study, a homogeneous transformation matrix is adopted to convert the point cloud coordinate system to the robotic arm's coordinate system, and the transformation matrix is defined as follows:

$$T_{4 \times 4} = \begin{bmatrix} R & e \\ 0 & 1 \end{bmatrix} \quad (9)$$

where R denotes a 3×3 rotation matrix, and e represents the translation vector.

2.3.2. Extraction of Surface Center Feature Points for Each Point Cloud Segment

Based on spherical geometric constraints and combined with the spatial distribution statistics of the point cloud, the central feature point of each sliced point cloud on the spherical surface is calculated. The essence lies in identifying the point that can represent the geometric center of the sliced point cloud set on the sphere, providing a reference benchmark for robotic arm path planning. Given the minimal curvature variation of the slice, all point clouds in this region are simplified into a single ink application point C , which is represented by the central point of the region, i.e., the centroid of the sliced point cloud. For each point cloud segment containing n points with coordinates denoted as $(x_i, y_i, z_i) (i = 1, 2, \dots, n)$, the centroid coordinate (C_x, C_y, C_z) of each segment is calculated as follows:

$$C = \frac{1}{N} \sum_{i=1}^n M_j \quad (10)$$

Among them, the set of the midpoint $\{M_1, M_2, M_3, \dots, M_j\}$ of the slice is stored in the point set G with the data of the inked point C .

2.3.3. Normal Vector Estimation of Ink Application Points

For each data point M_j in the execution point set G , its corresponding normal vector n can be regarded as the normal vector of the micro-tangent plane of the surface at that point. However, during the actual ink application process of the robotic arm, since the workpiece surface is spherical and the central axis of the ink application device must be perpendicular to the surface, the pen attitude needs to align with the normal vector direction (i.e., pointing toward the spherical center). Thus, the theoretical normal vector direction of each central point is the radial vector n pointing from the coordinate system origin to the point, defined as follows:

$$n = \frac{M_j - O}{\|M_j - O\|} \quad (11)$$

Calculating the normal vector of each central point using the aforementioned formula ensures that the robotic arm's end effector remains perpendicular to the spherical surface, enabling precise trajectory control.

2.3.4. Sorting of Slice Data

The point cloud segments obtained via the aforementioned slicing algorithm are disorganized and cannot be directly used as path points; thus, data sorting is required to enable effective application as path points. It is evident that the groove texture consists of two mutually perpendicular circular rings and a curve intersecting each ring twice. During ink application, the robotic arm must transition smoothly between different curves, making the selection of the ink application starting point and the rubber ball clamping position particularly critical. In this study, the clamping position is defined as the circular ring without an air nozzle rotated 22.5° clockwise around the X-axis. This configuration ensures the rotating gripper clamps on the non-groove area of the rubber ball, avoiding interference with the ink application process. The clamping position is illustrated in Figure 8 below.

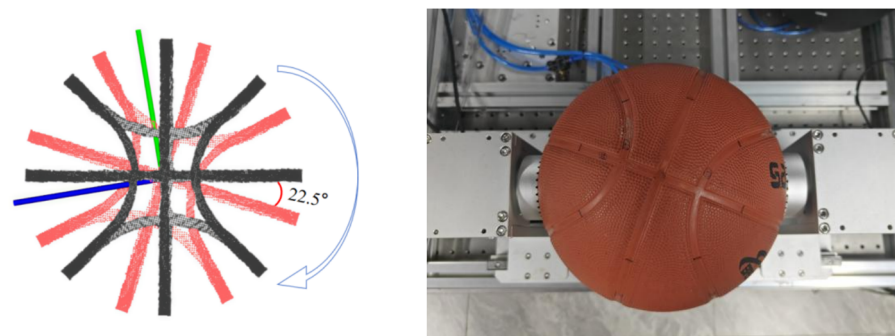


Figure 8. Rubber ball clamping position variation schematic diagram and physical clamping diagram.

The inking sequence is from the first positive circle to the second positive circle and then to the complex ($i = 1, 2, \dots, 50$) curve. For the layered cutting of the conical surface, the layered point set E is sorted, with the first and last layers each containing one element. The starting point for inking is at point $i = 1$. Next, the independent point clouds within layers 2 to 49 are separated and sorted. This process facilitates the separation and sorting of segments within layers, inter-layer topological connections, and global path planning.

2.3.5. Robot Arm Trajectory Generation via Point Fitting

The centroids of each point cloud segment obtained via earlier slicing form equally spaced path points. However, the micro-line segments connecting these discrete points are

not smooth. To ensure the robotic arm operates stably, smoothing fitting of these discrete points is necessary. As a derivative and development of Bezier curves, B-spline curves trace their theoretical foundation back to Schoenberg's [32] pioneering work in the 1940s. Subsequently, De Boer [33,34] and Cox [35] systematically constructed B-spline curves through recursive formulas, refining their theoretical framework. Inheriting the geometric invariance and convex hull property of Bezier curves, B-spline curves achieve significant breakthroughs in local control capability while exhibiting excellent characteristics such as reduced deviation. Leveraging these advantages, they have become a widely adopted fitting method in robotic path planning. Based on this, this study employs B-spline curves to smoothly fit the extracted execution points, thereby constructing a continuous and smooth ink application path. The mathematical representation of the k -th order B-spline curve is provided in Appendix B Equation (A5).

The basis functions of B-spline curves can be obtained via recursive calculation, and this recursive process is called the De Boor-Cox algorithm, namely:

$$\begin{cases} N_{i,0}(u) = \begin{cases} 1, u_i \leq u \leq u_{i+1} \\ 0, \text{others} \end{cases} \\ N_{i,k}(u) = \frac{u-u_i}{u_{i+k}-u_i} N_{i,k-1}(u) + \frac{u_{i+k+1}-u}{u_{i+k+1}-u_{i+1}} N_{i+1,k-1}(u) \end{cases} \quad (12)$$

In Equation (12), u represents the nodes, and k denotes the degree of the curve. On a B-spline curve, the first and last control points are typically utilized as the curve's start and end points, while the intermediate control points function as breakpoints or nodes. The knot vector is parameterized via the cumulative chord length method, with the first and last knots repeated $k + 1$ times to ensure the curve passes through the initial and final control points, defined as follows:

$$\begin{cases} u_0 = u_1 = u_2 = u_3 = 0 \\ u_{j+3} = u_{j+2} + \frac{\|G_j - G_{j-1}\|}{\sum_{m=1}^s \|G_m - G_{m-1}\|}, j = 1, 2, \dots, s-1 \\ u_{s+3} = u_{s+4} = u_{s+5} = u_{s+6} = 1 \end{cases} \quad (13)$$

The number of control points is $s + 1$, and the degree k of the spline curve and the number of knots $e + 1$ satisfy $e = s + 1 + k$. Once the degree k and the knot vector U are determined, the basis functions of the B-spline curve can be computed.

3. Results

3.1. System Construction and Experimental Setup

The software used in the experiment includes point cloud library (PCL 1.14.1) and MATLAB R2023b tools, and the hardware components of the control system include a SIASUN GCR5-910 (six-axis collaborative robot), a control cabinet and a computer. To achieve the collaborative control of the 6-axis robot and the rotary table, after comprehensive consideration of the adaptability of external axes and economy, the servo drive SV630NS1R6I and the servo motor MS1H4-20B30CB-T331R from Inovance (Shenzhen, China) were selected. Real-time communication and collaborative control with the robot are realized through the EtherCAT bus.

3.2. Denoising and Simplification Experiment

To reduce the influence of noise on the subsequent point cloud processing, the point cloud downsampling and denoising algorithms described in Section 2.1.1 are employed to process the scanned point cloud data. The leaf size of the voxel is set to 5–10 times the average spacing of the point cloud. The average spacing of the rubber ball point cloud is

approximately 0.2 mm. In this study, 1.5 mm is selected to balance detail retention and efficiency. The synergistic effect of voxel downsampling and statistical filtering effectively suppresses the impacts of scanning resolution fluctuations and random noise, reducing the number of point cloud data from 497,475 to 85,183, and the processed point cloud shape is shown in Figure 9.

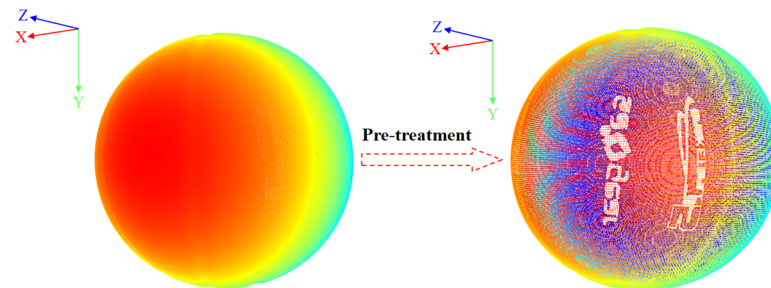


Figure 9. Rainbow mapping based on Z-axis height.

3.3. Sphere Fitting Experiment

After denoising and simplifying the point cloud, it is necessary to transfer the data from the scanning coordinate system to the sphere center coordinate system. Here, RANSAC is utilized to facilitate this process.

To balance the computational efficiency and fitting accuracy of the algorithm, comparative experiments were conducted on the RANSAC spherical fitting performance under different iteration numbers. As shown in Table A2 of Appendix A, the results indicate that when the number of iterations is set to 2, the internal point rate reaches 99.35% with a time consumption of only 15.44 s. Compared to higher iteration counts, this configuration significantly improves processing speed while maintaining fitting precision; thus, 2 iterations are selected as the optimal parameter in this study.

After processing by the PCL point cloud library algorithm, the fitted geometry of the RANSAC algorithm is shown in Figure 10. Its spherical center coordinates $\bar{q}$ are: (145.15, −11.8859, −326.682), with a radius of 110.034 mm. When the number of input point clouds is 85,183, the number of data points that meet the current model assumptions reaches 84,630, and the internal point rate is 99.35%. It provides crucial data for the subsequent extraction of groove point clouds.

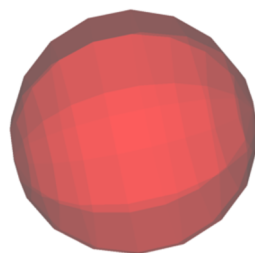


Figure 10. RANSAC algorithm fitting the geometry.

3.4. Experiments on Groove Point Cloud Extraction and Coordinate Calibration

3.4.1. Quantitative Comparative Analysis of Alignment Robustness

To quantitatively evaluate the superiority of the proposed algorithm, a comparative experiment was conducted between the RANSAC-compensated hybrid PCA and the conventional PCA using the same point cloud dataset. The experiment specifically evaluates the accuracy of coordinate origin determination under point cloud occlusions caused by laser scanning of light-absorbing black regions.

The visual comparison results are illustrated in Figure 11. As clearly observed in Figure 11b, the conventional PCA algorithm relies heavily on the arithmetic centroid of the point cloud. When encountering local “data holes” caused by laser scanning on black regions, the calculated origin undergoes a significant centroid drift, failing to align with the true geometric center of the sphere. Quantitative measurements indicate that the translational deviation of the coordinate axes using conventional PCA reaches 5.43 mm.

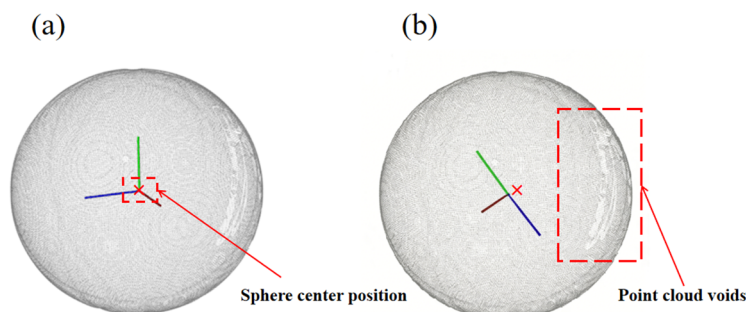


Figure 11. Comparison of coordinate system alignment performance between the proposed method and conventional PCA: (a) proposed method; (b) conventional PCA.

In contrast, Figure 11a displays the results of the proposed algorithm. Although the point cloud contains identical data gaps, the introduction of RANSAC sphere fitting as a geometric constraint allows the algorithm to bypass the dependency on uniform point distribution. Consequently, the coordinate origin is precisely localized at the true sphere center. The experimental data demonstrate that conventional PCA has inherent limitations when processing such incomplete spherical datasets, whereas the proposed method ensures high-precision alignment.

3.4.2. Implementation of Groove Extraction and Coordinate Calibration

Then, the algorithm described in Section 2.1.3 is employed to process the point cloud data obtained from the previous experimental steps. After obtaining the groove point cloud data, it is crucial to calibrate the scanning coordinate system with the sphere center coordinate system to ensure the accuracy of the slicing algorithm. As illustrated in Figure 12, through the coordinate transformation, the point clouds in the original scanning coordinate system are accurately converted to the aligned coordinate system. This calibration step eliminates scanning angle deviations and provides a unified reference for the subsequent spherical slicing and path planning.

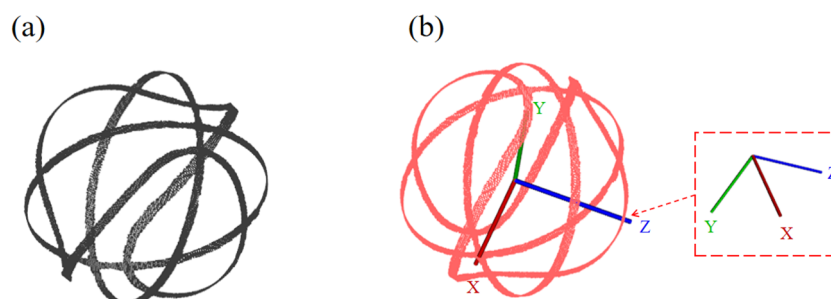


Figure 12. Groove extraction and coordinate calibration: (a) groove point cloud; (b) coordinate alignment diagram.

3.5. Point Cloud Slicing Experiment

3.5.1. Controlled Experiment

To verify the advantages of the proposed spherical equiangular conical slicing algorithm in complex feature extraction, a comparative study was conducted by projecting

and unfolding the sliced point clouds generated by both algorithms. The experiment specifically analyzed the distribution density and geometric fidelity of the feature points in high-curvature polar regions. The unfolded comparison results are illustrated in Figure 13.

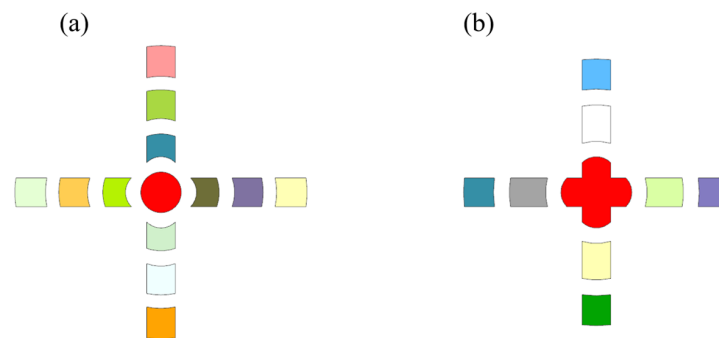


Figure 13. Point cloud slicing unfolding rendering: (a) conical surface segmentation diagram; (b) conventional segmentation diagram.

Figure 13a displays the unfolded point cloud generated by the proposed algorithm. It can be observed that the extracted points within the groove regions exhibit superior uniformity, maximizing the capture of intricate groove details.

In contrast, Figure 13b shows the results of the conventional planar slicing method. Compared to Figure 13a, the lengths of the point cloud segments extracted by the conventional method vary significantly. Given an identical number of slices, the conventional method yields slightly longer point cloud segments at the starting and ending slices. This discrepancy leads to geometric collapse during subsequent centroid calculations, causing the trajectory to shift towards the sphere's center.

Furthermore, comparative analysis reveals that to compensate for accuracy losses in polar regions, the conventional algorithm must significantly reduce the slicing step size. This results in an exponential increase in the number of path points, which easily triggers interpolation jitter during robotic execution. Conversely, the proposed method maintains a uniform and reasonable point density while ensuring high precision, effectively guaranteeing both the kinematic smoothness of the robot and processing efficiency.

3.5.2. Sectioning Experiment

By applying the aforementioned spherical latitudinal equiangular conical slicing algorithm, precise extraction and layering of the grooves on the rubber ball surface are achieved. Figure 14a displays the distribution of groove point clouds segmented by conical surfaces, and Figure 14b shows the rendering effect of its segmentation results.

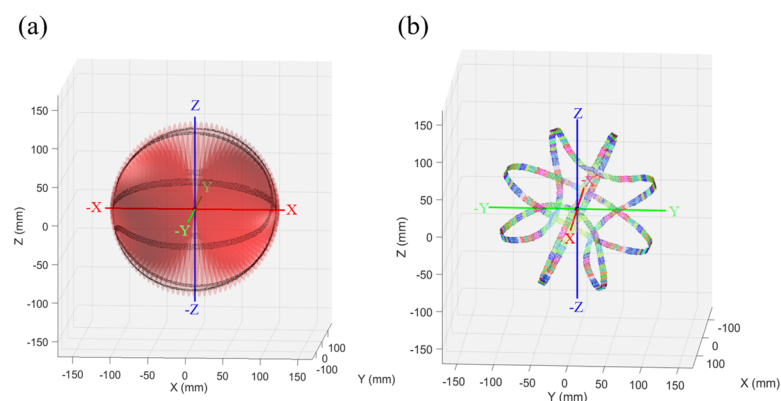


Figure 14. Groove point cloud segmentation: (a) conical surface segmentation diagram; (b) segmentation result rendering diagram.

As observed, the point clouds in each layer are orderly arranged with clear boundaries, demonstrating the superior performance of the algorithm in processing high-curvature spherical textures.

3.6. Point Cloud Centroid Extraction Experiment

To obtain the points readable by the robot arm, it is necessary to calculate the centroids of the point cloud segments obtained from the previous slicing process. The detailed calculation procedure is provided in Section 2.3.2, and the extracted centroid positions for each point cloud slice are illustrated in Figure 15.

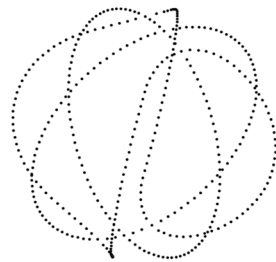


Figure 15. Centroid position of each extracted point cloud slice.

3.7. Point Cloud Calibration in Robot Coordinate System

Prior to the robotic arm performing ink application trajectory planning, a calibration operation between the point cloud and the robotic arm must be executed to convert the coordinate system of the rubber ball point cloud to that of the robotic arm. To address this issue, point selection calibration between the point cloud and the robotic arm is conducted. To improve accuracy, 10 sets of data are collected to calculate the homogeneous transformation matrix from the point cloud to the robotic arm, and the resulting transformation matrix is as follows:

$$M_{4 \times 4} = \begin{bmatrix} 0 & 0.5363 & -0.8435 & -414.322 \\ 0 & 0.8435 & 0.5363 & -1.308 \\ 1 & 0 & 0 & -252.656 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (14)$$

3.8. Transition Planning Between Curves

Through B-spline curve fitting, the execution points have been successfully converted into a continuous path plan. To ensure smooth transitions between different curves of these paths and guarantee the stability and efficiency of the ink application process, trajectory sequence planning is required. Starting from the origin O as the ink application starting point, the first full circle of ink application is completed via path ①② and upon returning to point O, the second full circle of ink application is completed via path ③④, as illustrated in Figure 16.

A transition point J is inserted between the endpoint O of the second positive circle and the starting point I of the next complex curve, as shown in Figure 17. By adjusting the control points of the B-spline curve, a consistent first derivative can be achieved between these three points. This optimization of the connection path ensures smoothness, as illustrated by Equation (15):

$$P'(u_O) = P'(u_I) = P'(u_J) \quad (15)$$

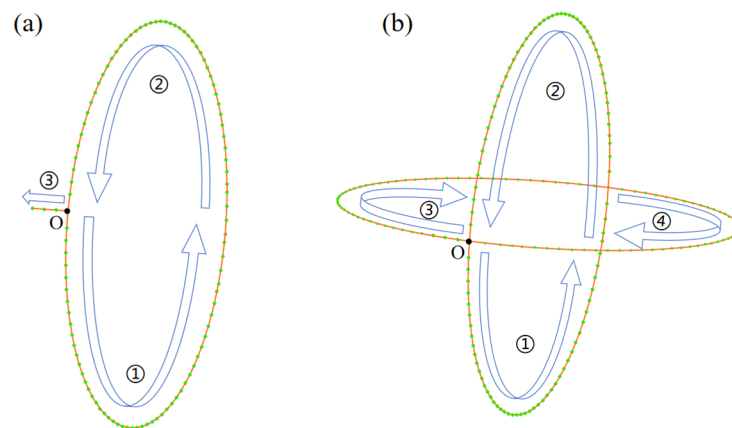


Figure 16. Schematic diagram of the ink application trajectory sequence for the first half: (a) ink application sequence for the first perfect circle; (b) ink application sequence for the second perfect circle.

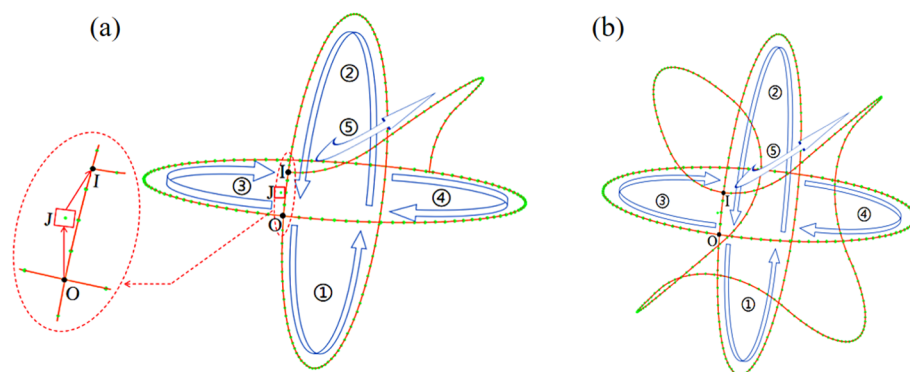


Figure 17. Schematic diagram of the ink application trajectory sequence for the second half: (a) magnified view of the transition point; (b) schematic diagram of ink application sequence for complex curves.

When the first-order derivatives at the connection points of two curves at different levels are equal, it indicates that these points are continuous in the tangent direction. This method helps avoid sudden changes in the angle at path junctions, ensuring smoother and more natural movement of the robotic arm.

When the robotic arm transitions between different layers, its movement speed must be appropriately reduced to avoid vibrations and attitude deviations caused by rapid movement. Additionally, the robotic arm's attitude needs to be adjusted to maintain a proper angle between the ink application tip and the workpiece surface. This helps reduce mechanical wear and the potential risk of malfunctions resulting from improper movement.

3.9. Path Repeatability Precision Validation

To evaluate the performance of this system, we conducted a verification of the path repeatability accuracy. Since this method aims to achieve closed-loop production without external devices, the accuracy verification is entirely based on the feedback data of the robot system itself. We instructed the robot to repeat the planned path 10 times. After each execution, we recorded the actual 3D coordinates fed back by its servo system at each preset path point, including the start point, end point, and path midpoint. The repeatability accuracy of the system is quantitatively evaluated by analyzing the distribution of these coordinate points in multiple executions.

The data collected from 10 repeated runs were statistically analyzed to calculate the coordinate standard deviations σ of each critical path point in the X, Y, and Z directions, and based on this, the spatial comprehensive repeatability accuracy 3σ was calculated. The

results are shown in Table 1. The spatial repeatability accuracy of all key points is better than 0.1 mm, with an average of 0.09 mm. This result fully demonstrates that the robot's execution of the path planned in this paper has extremely high stability and consistency, and can meet the process requirements of precise inking for rubber ball grooves.

Table 1. Statistical results of positional accuracy at monitoring points.

Monitoring Points	X-Direction (mm)	Y-Direction (mm)	Z-Direction (mm)	Spatial Composite (mm)
Start Point	0.05	0.04	0.03	0.07
Midpoint	0.07	0.06	0.05	0.10
End Point	0.06	0.05	0.04	0.09
Average Value	0.06	0.05	0.04	0.09

3.10. Ink Coating Quality Verification Based on Image Analysis

To verify the effectiveness and practicality of the proposed algorithm, we conducted a preliminary evaluation of the ink application quality on the rubber ball samples after ink application based on image analysis. This evaluation aimed to quantify two key industrial indicators: groove coverage rate and coating overflow rate, thereby verifying the effectiveness of the planned path from the perspective of the final process performance.

This experiment aimed to achieve quantitative evaluation of ink application quality based on the YOLOv8 instance segmentation model, by analyzing pairs of high-definition images of the same rubber ball collected at the same precision workstation before and after ink application. The experimental image acquisition system consisted of a fixed fixture and a Hikvision industrial camera (Model: MV-CA050-21UM, Resolution: 2592×1944), which ensured that the position and lighting conditions were strictly consistent between the two captures to eliminate registration errors.

The core analysis process is divided into three stages: First, the YOLOv8-seg model fine-tuned on a self-built rubber ball groove dataset is used to process the non-ink-coated images. Its core task is to identify and segment all grooves on the rubber ball surface and extract their pixel-level masks. The total number of pixels contained in these masks, denoted as N_{groove} is regarded as the reference total area of the groove regions.

Subsequently, the ink-coated images of the same rubber ball are input into the same model, where the task objective of the model is converted to segment the regions filled with black ink. Since the geometric positions of the grooves remain unchanged in the two precisely registered images, the pixel-by-pixel logical comparison between the ink region masks segmented from the ink-coated images and the reference groove masks obtained from the non-ink-coated images enables the accurate distinction between the number of ink pixels correctly filled in the grooves N_C and the number of ink pixels overflowing outside the grooves N_o .

Finally, the quantitative indicators of ink application quality are calculated using the following formulas. The ink coverage rate is expressed as:

$$CR = \frac{N_C}{N_{groove}} \quad (16)$$

Coating overflow rate:

$$OR = \frac{N_o}{N_i + N_o} \quad (17)$$

Fifteen standard rubber balls of the same model were selected for the experiment and divided into five sample groups, with three rubber balls per group. The experimental results were averaged for each sample group, as shown in Table 2.

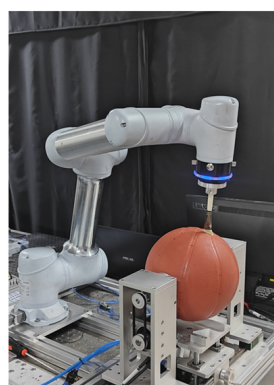
Table 2. Pixel statistics.

Sample Number	Number of Groove Pixels	Number of Ink Pixels in Grooves	Number of Overflowed Ink Pixels	Coverage Rate	Overflow Rate
1	191,250	182,720	4543	95.54%	2.43%
2	201,057	190,199	5438	94.60%	2.78%
3	190,544	181,989	4510	95.51%	2.42%
4	193,540	184,695	4944	95.43%	2.61%
5	195,072	185,279	5103	94.98%	2.68%

Analysis of the five sample groups indicates that the proposed method achieves an average coverage rate of 95.21% and an average overflow rate of 2.58%. It should be noted that this evaluation is based on two-dimensional (2D) image analysis, and its accuracy is limited by the lighting conditions of the shooting environment. The reflective properties of the rubber ball grooves may cause local overexposure of the ink marks, thus leading to a slight underestimation of the coverage rate.

These quantitative metrics provide direct evidence of the methodological improvements discussed in previous sections: First, the exceptionally low overflow rate (2.58%) is a direct consequence of correcting the 5.43 mm coordinate shift using the RANSAC-compensated alignment (Section 2.1.4). Without this correction, the positional drift would cause the ink line to deviate entirely from the groove. Second, in the high-curvature polar regions, the coverage remains above 95% due to the equiangular conical slicing proposed in Section 2.2. This effectively resolves the “under-inking” issue caused by the “geometric collapse” inherent in traditional planar slicing. By correlating physical process metrics with algorithmic precision, we demonstrate the scientific robustness and superior geometric fidelity of our model-free path planning framework.

On the basis of verifying the repeatability accuracy, this study preliminarily validates the accuracy of the proposed algorithm. Finally, we recorded the total task execution time for the robot to complete ink coating on all grooves of a single rubber ball. The timing starts when the robot initiates movement and ends when it finishes the operation and returns to the safe position, with an average time consumption of only 24.93 s. In comparison, manual teaching requires tens of minutes, and the ink application time alone ranges from 50 s to 90 s (This data is derived from on-site investigations at basketball manufacturing plants, with the average value obtained from multiple observations of workers with varying levels of proficiency). Therefore, the efficiency of the proposed method is fully compatible with the requirements of industrial production. Figure 18 visually presents the physical scene of the robot performing automated inking operations along the planned path.

**Figure 18.** Physical diagram of the robotic arm applying ink along the preset path.

4. Conclusions and Discussion

This study proposes a robust, model-free framework for autonomous path planning on spherical surfaces using 3D point clouds. The proposed method circumvents the reliance on CAD models and achieves full-process automation. Key conclusions and methodological impacts include:

- **Geometric Accuracy:** By addressing the 5.43 mm centroid drift through RANSAC-compensated alignment and resolving polar geometric collapse via conical slicing, the system ensures high-fidelity trajectory extraction.
- **Process Efficiency:** Experimental results demonstrate a significant 60% efficiency improvement over manual operations, with a robot path repeatability of 0.09 mm and a rapid processing time of 24.93 s per unit.
- **Quality Validation:** Quantitative assessment confirms a 95.21% coverage rate and a 2.58% overflow rate, providing a physical closed-loop validation of the algorithm's precision.

Critical Discussion on Limitations: Despite its effectiveness, the framework has limitations.

- **Data Sensitivity:** The coordinate alignment robustness is coupled with 3D sensor quality; extreme point cloud loss or intense ambient noise may degrade the RANSAC-based sphere fitting.
- **Geometric Generality:** The current mathematical model is specifically optimized for spherical topology. Extending this to free-form or non-spherical surfaces requires reconfiguring the topological mapping between slicing planes and surface normals.
- **Industrial Scalability:** Future work will focus on transitioning from semi-automated to fully autonomous feature extraction to better handle high-variability industrial patterns.

Author Contributions: Conceptualization, Z.Z. (Zhen Zeng) and A.Y.; methodology, A.Y.; software, A.Y.; validation, Z.Z. (Zhen Zeng), X.X. and N.O.; formal analysis, X.X.; investigation, X.X. and N.O.; resources, Z.Z. (Zhongsheng Zhai); data curation, A.Y.; writing—original draft preparation, A.Y.; writing—review and editing, A.Y.; visualization, A.Y.; supervision, Z.Z. (Zhen Zeng); project administration, Z.Z. (Zhongsheng Zhai); funding acquisition, Z.Z. (Zhongsheng Zhai). All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Hubei Natural Science Foundation (No. 2022CFA006).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

RANSAC	Random Sample Consensus
PCL	Point Cloud Library
RANSAC-PCA	Improved PCA Calibration Method Incorporating RANSAC
PCA	Principal Component Analysis
EtherCAT	Ethernet for Control Automation Technology
FIBA	International Basketball Federation
RGB	Red, Green, Blue
CAD	Computer-Aided Design

2D	Two-dimensional
3D	Three-dimensional
SV630NS1R6I	Servo Drive Model (Inovance)
MS1H4-20B30CB-T331R	Servo Motor Model (Inovance)
MV-CA050-21UM	Industrial Camera Model (Hikvision)
YOLOv8	You Only Look Once Version 8

Appendix A

Table A1. Basketball circumference and weight tolerances.

Ball Size	7	6	5	Lightweight 5
Circumference	750–770 mm	715–730 mm	685–700 mm	685–700 mm
Weight	580–620 g	510–550 g	465–495 g	360–390 g

Table A2. Comparison table of RANSAC spherical fitting performance under different iteration numbers.

Number of Iterations	1	2	50	100
Time	12.55 s	15.44 s	16.32 s	17.27 s
Internal Points	84,601	84,630	84,630	84,630
Internal Point Rate	99.31%	99.35%	99.35%	99.35%

Appendix B

$$S_{Leaf} = 7.5 \times \frac{1}{N} \sum_{i=1}^N \min(\|p_i - p_j\|) \quad (A1)$$

$$\bar{p} = \left(\frac{1}{n} \sum_{k=1}^n x_k, \frac{1}{n} \sum_{k=1}^n y_k, \frac{1}{n} \sum_{k=1}^n z_k \right) \quad (A2)$$

In Equation (A2), n denotes the total number of points within the voxel, and (x_k, y_k, z_k) represents the 3D coordinates of the k -th point inside it.

$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = r^2 \quad (A3)$$

The first (x_0, y_0, z_0) is the ball center r coordinate, which is the radius of the sphere.

$$\begin{cases} (x_1 - x_0)^2 + (y_1 - y_0)^2 + (z_1 - z_0)^2 = r^2 \text{ (Point } p_1) \\ (x_2 - x_0)^2 + (y_2 - y_0)^2 + (z_2 - z_0)^2 = r^2 \text{ (Point } p_2) \\ (x_3 - x_0)^2 + (y_3 - y_0)^2 + (z_3 - z_0)^2 = r^2 \text{ (Point } p_3) \\ (x_4 - x_0)^2 + (y_4 - y_0)^2 + (z_4 - z_0)^2 = r^2 \text{ (Point } p_4) \end{cases} \quad (A4)$$

$$P(u) = \sum_{i=0}^s d_i N_{i,k}(u) \quad (A5)$$

In this formula, $\{d_i\}$ is the vertex sequence of the control polygon, and $N_{i,k}(u)$ is the basis function defined by the node vector $U = \{u_1, u_2, \dots, u_e\}$.

References

1. How It's Made: Rubber Balls. Available online: <https://www.youtube.com/watch?v=K1VfdXmqjN8> (accessed on 17 January 2026).
2. Othman, N.; Lerk, L.L.; Chelliapan, S.; Mohammad, R. Comparative Content of Harmful Substances Contained in the Raw Material for Various Types of Printing Ink. *Int. J. Res. Sci.* **2017**, *3*, 7–12. [CrossRef]
3. Liu, Y.; Liu, J.; Tian, X. An approach to the path planning of intersecting pipes weld seam with the welding robot based on non-ideal models. *Robot. Comput. Integr. Manuf.* **2019**, *55*, 96–108. [CrossRef]

4. Arents, J.; Greitans, M. Smart Industrial Robot Control Trends, Challenges and Opportunities within Manufacturing. *Appl. Sci.* **2022**, *12*, 937. [\[CrossRef\]](#)
5. Gleeson, D.; Jakobsson, S.; Salman, R.; Ekstedt, F.; Sandgren, N.; Edelvik, F.; Carlson, J.S.; Lennartson, B. Generating Optimized Trajectories for Robotic Spray Painting. *IEEE Trans. Autom. Sci. Eng.* **2022**, *19*, 1380–1391. [\[CrossRef\]](#)
6. Ren, T.; Dong, Y.; Wu, D.; Chen, K. Design of Direct Teaching Behavior of Collaborative Robot Based on Force Interaction. *J. Intell. Robot. Syst.* **2019**, *96*, 83–93. [\[CrossRef\]](#)
7. Song, H.; Ye, R.; Xie, M. Research on Complex Surface Grinding Path Planning of Grinding Robot Based on NX and Tecnomatix. *Int. J. Adv. Manuf. Technol.* **2024**, *133*, 1961–1971. [\[CrossRef\]](#)
8. Zhou, B.; Tian, T. Robotic disc grinding path planning method based on multi-objective optimization for nuclear reactor coolant pump casing. *J. Manuf. Syst.* **2024**, *77*, 810–833. [\[CrossRef\]](#)
9. Yang, L.; Li, E.; Long, T.; Fan, J.; Liang, Z. A Novel 3-D Path Extraction Method for Arc Welding Robot Based on Stereo Structured Light Sensor. *IEEE Sens. J.* **2019**, *19*, 763–773. [\[CrossRef\]](#)
10. Zhao, T.; Yan, Z.; Wang, L.; Pan, R.; Wang, X.; Liu, K.; Guo, K.; Hu, Q.; Chen, S. Hybrid Path Planning Method Based on Skeleton Contour Partitioning for Robotic Additive Manufacturing. *Robot. Comput. Integr. Manuf.* **2024**, *85*, 102633. [\[CrossRef\]](#)
11. Zhou, P.; Peng, R.; Xu, M.; Wu, V.; Navarro-Alarcon, D. Path Planning with Automatic Seam Extraction Over Point Cloud Models for Robotic Arc Welding. In *IEEE Robotics and Automation Letters*; IEEE: New York, NY, USA, 2021. [\[CrossRef\]](#)
12. Yang, B.; Cheng, J.; Xue, B.; Jiao, J.; Liu, M. Efficient Global Navigational Planning in 3D Structures Based on Point Cloud Tomography. *IEEE/ASME Trans. Mechatron.* **2024**, *30*, 321–332. [\[CrossRef\]](#)
13. Yan, L.; Shuai, Y.; Xin, C.Q.T. A Novel Path Planning Method of Robotic Grinding for Free-Form Weld Seam Based on 3D Point Cloud. *Int. J. Adv. Manuf. Technol.* **2024**, *131*, 5155–5176. [\[CrossRef\]](#)
14. Zhang, Q.; Li, R.; Sun, J.; Wei, L.; Huang, J.; Tan, Y. Dynamic 3D Point-Cloud-Driven Autonomous Hierarchical Path Planning for Quadruped Robots. *Biomimetics* **2024**, *9*, 259. [\[CrossRef\]](#)
15. Zhang, Y.; Geng, Y.; Tian, X.; Zhou, L. Feature Extraction and Robot Path Planning Method in 3D Vision-Guided Welding for Multi-Blade Wheel Structures. *Opt. Lasers Eng.* **2024**, *176*, 108066. [\[CrossRef\]](#)
16. Hu, Y.; Xie, F.; Yang, J.; Zhao, J.; Mao, Q.; Zhao, F.; Liu, X. Efficient Path Planning Algorithm Based on Laser SLAM and an Optimized Visibility Graph for Robots. *Remote Sens.* **2024**, *16*, 2938. [\[CrossRef\]](#)
17. Guo, W.; Huang, X.; Qi, B.; Ren, X.; Chen, H.; Chen, X. A Novel Versatile Method for Generating Machining Path Directly from Point Cloud. *J. Manuf. Process.* **2024**, *120*, 15–27. [\[CrossRef\]](#)
18. Ding, X.; Qiao, J.; Liu, N.; Yang, Z.; Zhang, R. Robotic Grinding Based on Point Cloud Data: Developments, Applications, Challenges, and Key Technologies. *Int. J. Adv. Manuf. Technol.* **2024**, *131*, 3351–3371. [\[CrossRef\]](#)
19. Wang, H.; Li, Y. Limited Environmental Information Path Planning Based on 3D Point Cloud Reconstruction. *J. Supercomput.* **2024**, *80*, 10931–10958. [\[CrossRef\]](#)
20. Xu, C.; Wang, J.; Zhang, J.; Lu, C. A New Welding Path Planning Method Based on Point Cloud and Deep Learning. In *Proceedings of the 2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, Hong Kong, China, 20–21 August 2021; IEEE: New York, NY, USA, 2020. [\[CrossRef\]](#)
21. Marukatat, S. Tutorial on PCA and Approximate PCA and Approximate Kernel PCA. *Artif. Intell. Rev.* **2022**, *56*, 5445–5477. [\[CrossRef\]](#)
22. Qian, X.; Ye, C. NCC-RANSAC: A Fast Plane Extraction Method for 3-D Range Data Segmentation. *IEEE Trans. Cybern.* **2014**, *44*, 2771–2783. [\[CrossRef\]](#) [\[PubMed\]](#)
23. Ji, Z.; Ai, H.X.; Han, Y.; Yao, J.; Wang, Y.; Zheng, F.; Sun, Z.; Wenjie, W. Research on Key Technologies of Spatial Non-Cooperative Target Pose Measurement. *IEEE Access* **2025**, *13*, 175733–175745. [\[CrossRef\]](#)
24. Chu, H.; Cheng, Y.; Li, Y.; Fan, J.; Chen, N.; Wang, Y. Fine Extraction of Geometric Contour Feature Based on Spherical Thinning. *Opt. Laser Technol.* **2025**, *192*, 113784. [\[CrossRef\]](#)
25. Raible, J.; Braun, C.; Huber, M. Automatic Path Planning for Robotic Grinding and Polishing Tasks Based on Point Cloud Slicing. In *Proceedings of the Europe ISR 2023—International Symposium on Robotics*, Stuttgart, Germany, 26–27 September 2023; pp. 382–389.
26. Liu, K.; Sheng, Z.; Zhang, W.; Zhong, Z.; Yang, X.; Chen, H.; Xiao, R. A Novel Robotic Multi-Layer Multi-Pass Welding Adaptive Path Generation Method Based on Point Cloud Slicing and Transformer for Saddle-Shaped Weld Seams. *J. Manuf. Process.* **2025**, *141*, 1578–1594. [\[CrossRef\]](#)
27. Miknis, M.; Davies, R.; Plassmann, P.; Ware, A. Efficient Point Cloud Pre-Processing Using the Point Cloud Library. *Int. J. Image Process.* **2016**, *10*, 63–72.
28. Gordon, W.J.; Riesenfeld, R.F. B-Spline Curves and Surfaces. *Comput. Aided Geom. Des.* **1974**, *23*, 95–126. [\[CrossRef\]](#)
29. Pi, L. Target Fitting Method for Spherical Point Clouds Based on Projection Filtering and K-Means Clustered Voxelization. *Sensors* **2024**, *24*, 5762. [\[CrossRef\]](#) [\[PubMed\]](#)

30. International Basketball Federation (FIBA). Official Basketball Rules 2024: Basketball Equipment. 2024. Available online: <https://fiba.basketball/docs/official-basketball-rules-2024-equipment> (accessed on 17 January 2026).
31. Chen, W.; Li, X.; Ge, H.; Wang, L.; Zhang, Y. Trajectory Planning for Spray Painting Robot Based on Point Cloud Slicing Technique. *Electronics* **2020**, *9*, 908. [[CrossRef](#)]
32. Schoenberg, I.J. Contributions to the Problem of Approximation of Equidistant Data by Analytic Functions. In *I.J. Schoenberg Selected Papers*; Birkhäuser: Boston, MA, USA, 1988. [[CrossRef](#)]
33. Boor, C.D. On Calculating with B-Splines. *J. Approx. Theory* **1972**, *6*, 50–62. [[CrossRef](#)]
34. Boor, C.D. A Practical Guide to Spline. *Math. Comput.* **1980**, *34*, 325. [[CrossRef](#)]
35. Cox, M.G. The Numerical Evaluation of B-Splines. *IMA J. Appl. Math.* **1972**, *10*, 134–149. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.