Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería

ARTICLE

A Point Cloud Registration Method for Ship Plates Based on Density Denoising and Anderson-Accelerated Iterative Closest Point

Zhonghao Wang[1,2], Xin Liu[1,2,*], Dongxu Liu[1,2], Yang Zhang[1,2], Pengfei Hou[1,2], Wenlong Sun[1,2], Yabin Li[3] and Haiwen Yuan[4]

[1]Department of Ship and Port Engineering, Shandong Jiaotong University, Weihai, China
[2]Department of Intelligent Shipping, Weihai Institute of Marine Information Science and Technology, Weihai, China
[3]Department of Engineering, Qingdao Institute of Shipping Development Innovation, Qingdao, China
[4]Department of Shipping, Wuhan University of Technology, Wuhan, China
*Corresponding Author: Xin Liu. Email: axinzaixian@163.com

Received: 08 January 2026; Accepted: 01 April 2026

ABSTRACT: Point cloud registration is essential for closed-loop digital forming of ship plates, where CAD-to-scan alignment is required for surface error evaluation and compensation. Industrial ship-plate point clouds acquired by structured-light or laser sensing are often corrupted by boundary-related structural outliers and are feature-sparse, causing classical ICP to be sensitive to mismatches and slow to converge. We propose a training-free, deployment-oriented pipeline that combines density-based outlier removal with an Anderson-accelerated ICP (A-ICP) formulated in the SE(3) Lie algebra. Experiments on three representative plate geometries show 37.7%–39.5% lower registration error and indicate improved convergence behavior relative to classical ICP variants. The method is further validated on an SKWB-2500 machine-in-the-loop workflow, achieving MAE of 0.68 mm (sail-shaped) and 0.38 mm (saddle-shaped), with corresponding RMSE of 0.7301 mm and 0.4141 mm. Learning-based baselines are not benchmarked due to the lack of a fair in-domain dataset and retraining protocol under proprietary sensing conditions.

KEYWORDS: Intelligent manufacturing; point cloud registration; Anderson acceleration; density-based outlier removal; shipbuilding

1 Introduction

Metallic three-dimensional curved panels are widely used in shipbuilding, aerospace, and automotive industries due to their high ductility and uniform stress distribution characteristics. Ship outer plates, as a typical example, are composed of numerous three-dimensional curved panels. Their forming quality and efficiency directly impact the hull structural strength and construction schedule. In the shipbuilding process, three-dimensional curved surfaces need to be unfolded into two-dimensional planes for nesting design. Subsequently, flat steel plates are formed into complex 3D shapes like “saddle” or “sail” types through plastic forming processes. Constrained by the small-batch, high-variety production nature of the shipbuilding industry, traditional integrated molds have limited application due to high mold development time and cost. The industry currently commonly uses a processing method combining roll bending and line heating. However, this method suffers from poor working conditions, high labor intensity, and low efficiency. Although extensive research has been conducted on the automation of line heating for 3D curved ship plates, this problem remains a major bottleneck in digital shipbuilding. Compared to the processing efficiency and intelligence level of processes like plate cutting, the forming of 3D curved ship plates has become a bottleneck hindering the development of digital shipbuilding. High-precision point cloud registration technology is the key to breaking this bottleneck.

Point cloud data registration is a crucial step in 3D scanning and modeling, primarily consisting of two stages: coarse registration and fine registration. Coarse registration aims to quickly establish an initial correspondence between point clouds. Common methods include the turntable method [1], the target-based method [2], and the surface feature-based method [3]. However, coarse registration often fails to meet accuracy requirements in many applications, and point clouds may have overlaps, necessitating subsequent fine registration.

For fine registration, the Iterative Closest Point (ICP) algorithm and its variants, as reviewed in [4] and improved in later studies such as [5], are among the most widely used methods for point cloud fine registration.

To address these issues, numerous improved algorithms have been proposed. Regarding feature utilization, Cheng et al. [6] combined improved SAC-IA with KD-tree-assisted ICP optimization for local point cloud alignment, Zhang et al. [7] investigated surface processing and repair for unorganized point clouds, and Dey et al. [8] investigated neighborhood-based geometric feature extraction for point cloud analysis. However, these methods are relatively sensitive to noise, and the accuracy of feature estimation can be easily affected. In terms of filtering and denoising, Arazm et al. [9] introduced an adaptive Wiener filter, Zeng et al. [10] proposed Laplacian-based denoising, and Zheng [11] and Ren et al. [12] investigated bilateral or multiscale filtering strategies for geometric data smoothing. However, the issue of volume shrinkage in feature regions remains difficult to completely avoid. Regarding computational efficiency, Cao [13] proposed a global-structure- and adaptive-weight-aware ICP variant. Eckart [14] and Sharp [15] investigated efficient registration strategies based on hierarchical modeling and invariant features, Vrochidis et al. [16] incorporated color information into ICP-style registration, and Lin [17] studied scale alignment for point clouds with different scales. These approaches have all improved efficiency to some extent.

Additional studies have also reported notable progress in point cloud registration, including deep image-to-point cloud registration [18] and robust global registration via fast descriptors and correspondence propagation [19]. In terms of robustness, Rousseeuw [20] introduced the Least Median of Squares (LMedS) method, Guan et al. [21] proposed an improved ICP-based registration strategy, and Vizzo et al. [22] incorporated adaptive thresholding into correspondence-based registration, thereby improving tolerance to outliers.

Additionally, other studies have achieved efficient alignment of multi-view data using Moving Least Squares (MLS) surfaces [23], improved coarse-to-fine registration by combining SAC-IA with improved ICP [24], optimized search efficiency by constructing multi-level index structures such as octrees and K-D trees [25], and balanced accuracy with computational efficiency by introducing Rodrigues matrix parameterization for rotation operations [26]. In recent years, Bai X et al. [27] proposed a novel deep neural network, Point DSC, which combines spatial consistency to optimize point cloud matching, thereby improving registration accuracy.

Notably, to improve the efficiency and anti-interference capability of data acquisition at the source, the shipbuilding industry has begun to adopt advanced sensors such as structured light depth cameras to capture ship plate point cloud data with high frame rates and high density. However, while the introduction of these new devices addresses long-standing issues, it also gives rise to new challenges.

Despite these advances, industrial ship-plate registration under structured-light sensing still faces several unresolved challenges. First, the acquired point clouds often suffer from substantial noise and structural outliers under complex working conditions. Second, existing registration algorithms still rely heavily on a suitable initial pose, and inaccurate initialization can easily lead to failure or poor local convergence. Third, a persistent trade-off remains between registration accuracy and the computational burden of processing large-scale point clouds in repeated forming cycles. Therefore, there is still a strong need for a solution that can address these issues in a unified and deployment-oriented manner.

In industrial digital forming, registration is not an isolated 3D vision task: its output directly determines the accuracy of surface error evaluation between the measured formed plate and the target CAD surface, and therefore affects subsequent compensation decisions. Hence, beyond standalone registration metrics, a convincing evaluation should also demonstrate that the registration module can run stably within a real forming-and-measurement loop under the target sensing setup.

Representative learning-based registration methods and deep correspondence networks have reported strong performance on public benchmarks. Nevertheless, directly transferring those benchmark settings to industrial ship-plate forming can be misleading, because ship-plate measurements are dominated by smooth and feature-sparse surfaces and are affected by boundary-related structural outliers that differ from the noise patterns in generic datasets. A fair comparison would require constructing an in-domain dataset collected under the same sensor and forming conditions and retraining or calibrating each model under a unified protocol, which is difficult when the raw measurements are proprietary (see Data Availability). Therefore, this work focuses on a training-free, geometry-driven pipeline that can run stably on standard CPU hardware and is validated both in offline experiments and in an on-machine closed-loop forming evaluation (Section 4.3), rather than reporting potentially misleading out-of-domain benchmark numbers.

To address these challenges, this study systematically explores key links in the ship-plate point cloud processing workflow during the plate forming process, including outlier generation mechanisms and removal, geometric feature extraction, coarse initialization, and fine registration. The main contributions are as follows:

(1) Sensor-oriented data purification: We propose a density evaluation-based outlier removal algorithm specifically targeting structural outliers produced by structured-light measurement near plate boundaries.

(2) Robust geometric cue construction for feature-sparse ship plates: We integrate SAM-based boundary cues, MLS-based curvature extrema, and surface normal consistency to construct stable geometric guidance for initialization under low-curvature and noisy industrial measurements.

(3) Initialization-robust coarse registration: We propose a deterministic coarse initialization based on curvature extrema and normal-consistency matching to provide a reliable initial pose for fine registration on feature-sparse ship-plate surfaces.

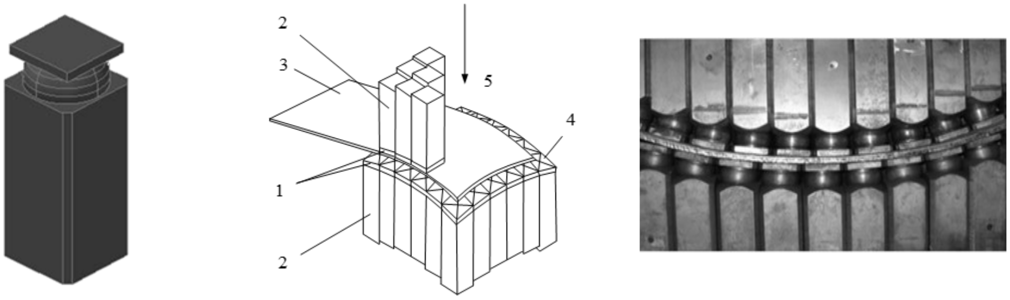
(4) Anderson-accelerated fine registration (A-ICP): Anderson acceleration is incorporated into ICP via SE(3) Lie algebra parameterization, providing a lightweight and training-free mechanism to improve convergence behavior and robustness in iterative rigid registration.

(5) Engineering deployment and closed-loop forming validation: The proposed A-ICP module is embedded into an upgraded SKWB-2500 bending machine and validated in a real forming-and-measurement loop, demonstrating reliable CAD-to-scan alignment for quantitative surface error evaluation under industrial sensing conditions.

2 Technical Background and Challenges in Ship Plate Forming

The digital cold bending forming of three-dimensional curved ship outer plates is a core aspect of intelligent manufacturing in shipbuilding. To better contextualize the registration challenges addressed in this study, it is necessary to first understand the industrial setup and technological constraints of modern plate bending systems.

In recent years, 3D CNC plate bending machines based on multi-indenter adjustable molds (such as the SKWB series) have become key equipment for realizing this process. This type of equipment utilizes upper and lower rows of independently controllable indenters to dynamically form a target curved surface mold, progressively press-forming the plate. The technical core of such bending machines lies in the adoption of the “Square Indenter Adjustable Flexible Mold” and “Asymmetric Press Forming Technology”, As shown in Fig. 1.



(a) (b) (c)

Figure 1: Structures and Technologies Related to Plate Bending Process. (**a**) Square Indenter Adjustable Flexible Indenter. (**b**) Square Indenter Adjustable Flexible Mold. (**c**) Asymmetric Press Forming Technology. Where 1-Square indenter; 2-Support body; 3-Workpiece plate; 4-Lower die surface; 5-Loading mechanism.

Compared to the hemispherical indenters used in traditional multi-point forming, the square indenters create surface contact with the plate. The design of staggered upper and lower indenters effectively disperses stress, significantly suppresses the generation of indentations and wrinkles, and greatly improves the forming quality of medium-thick plates. However, to achieve high-precision forming, this process heavily relies on a closed-loop feedback control system: after each pressing operation, the 3D point cloud data of the formed plate section must be rapidly acquired by an online measurement system (e.g., laser scanning or structured-light cameras). This data is then accurately registered with the design model to calculate the springback amount and guide the indenter stroke compensation for the next cycle. It is precisely within this core “measurement–registration–compensation” cycle that existing point cloud processing technologies face severe challenges: data quality issues with point clouds containing significant noise and outliers due to the harsh on-site environment of the bending machine, the initial registration dilemma where traditional fine registration algorithms are sensitive to initial pose and lack reliable initial pose estimation methods, and the challenge of balancing efficiency with accuracy when processing high-density point clouds over multiple “pressing–measurement–registration” cycles. These challenges collectively form the starting point of this research. Therefore, point cloud registration in the closed-loop forming system is constrained by three practical factors: boundary-induced structural outliers, weak geometric cues for initialization on smooth surfaces, and strict runtime requirements across repeated cycles. To address them in a unified manner, we first analyze the dominant outlier mechanism and remove structurally separated boundary points (Section 3.1), then construct stable geometric cues for coarse initialization (Sections 3.2 and 3.3), and finally accelerate fine registration via Lie-algebra-based Anderson-accelerated ICP (Section 3.4).

3 Method for Point Cloud Registration in Ship Outer Plate Forming

Accurate point cloud registration is fundamental to the forming precision of ship outer plates, yet practical measurements are often degraded by outliers and significant initial misalignment, which can lead to mismatched correspondences and suboptimal convergence. To address these challenges, this section presents an end-to-end point cloud registration pipeline for ship outer plate forming. The pipeline starts with point cloud acquisition and an analysis of outlier generation mechanisms, followed by preprocessing with density-based outlier removal to improve data quality. It then performs boundary extraction and segmentation to obtain stable geometric cues, computes a geometric-feature-guided coarse initialization to reduce sensitivity to the initial pose, and finally applies Anderson-accelerated ICP (A-ICP) for efficient and accurate fine registration. Together, these steps provide a reliable data foundation for subsequent error evaluation and closed-loop forming compensation, thereby improving the overall forming accuracy of plate bending machines.

3.1 Point Cloud Data Preprocessing

The prerequisite for high-precision registration is the acquisition of high-quality point cloud data. As mentioned above to enhance acquisition efficiency and anti-interference capability, structured-light depth cameras have been widely adopted to capture ship-plate point clouds. However, due to both the measurement mechanism and the on-site environment, the raw scans inevitably contain noise and outliers (bad points), which can severely degrade subsequent feature extraction and registration. Therefore, point cloud preprocessing is an indispensable first step in the registration pipeline. This section first analyzes the root causes of bad points and then proposes an efficient density-based outlier removal algorithm to provide a clean and reliable data foundation for subsequent processing. In our scenario, the dominant corruption is boundary-related structural outliers that are spatially separated from the true plate surface; thus, we start with a density-based preprocessing step to remove such bad points before feature extraction and registration. This preprocessing improves the geometric consistency of the remaining points, providing more reliable input for downstream feature extraction and registration on feature-sparse ship-plate surfaces.

3.1.1 Mechanism of Outlier Generation

To prevent displacement and wrinkling during processing, 3D CNC ship plate bending machines usually adopt a non-opposing pressure mode with cross-pressure applied by upper and lower pressure heads. In such equipment, the online measurement module acquires surface height information (Z-values) through depth sensing or scanning. A typical source of structural outliers is the misidentification of the target measurement surface: when the sensing position lies on the workpiece, the returned depth corresponds to the plate surface; however, when the sensing position moves close to the boundary or outside the plate, the sensor may capture the lower-die surface instead, producing points that belong to the die rather than the workpiece. These points manifest as abnormal spatial outliers (i.e., bad points) in the final point cloud, as shown in Fig. 2, and can severely interfere with subsequent feature extraction and registration. Therefore, identifying and removing such bad points is the first step of the proposed workflow.

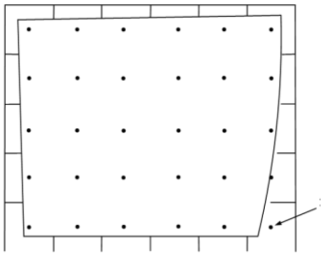


Figure 2: Generation of bad points.

3.1.2 Density-Based Outlier Removal Algorithm

To address the aforementioned issue of outliers introduced by specific measurement scenarios, this paper employs a density-based outlier removal algorithm. Its core premise is that the local neighborhood point density of outliers (bad points) is significantly lower than that of normal data points. By computing a local density estimate for each point and setting an appropriate threshold, these anomalous points deviating from the main point cloud can be effectively identified and removed. It should be emphasized that this density criterion is designed for structurally separated outliers (e.g., boundary-induced bad points) rather than serving as a general-purpose Gaussian denoiser. The key steps of this algorithm are as follows:

1. Distance calculation basis: Let the point cloud dataset be $X = \{x_a\}_{a=1}^{N}, x_a \in R^c$, calculate the Euclidean distance matrix P between any two points, defined as:

$$P(a,b) = \sqrt{\sum_{m=1}^{c} \left(x_a^m - x_b^m\right)^2}, a,b \in [1,2,\dots,N] \tag{1}$$

Among them, c is the point cloud dimension, here $c = 3$; x_a^m represents the coordinate of point x_a in the m-th dimension. To eliminate the dimension influence, normalize P to get P^*, so that its element value range is $[0,1]$.

2. Construction of similarity matrix: Define the similarity matrix SM as:

$$SM = I - P^* \tag{2}$$

where I is the identity matrix. The sum of the a-th row of the similarity matrix SM is taken as the initial density estimate for point x_a:

$$\gamma_a^0 = \sum_{b=1}^{N} SM_{ab} \tag{3}$$

3. Construction of the k-nearest neighbor graph: For each point x_a, select its k nearest neighbors with the highest similarity to form the set $W(x_a)$, and construct the adjacency matrix A:

The adjacency matrix A can be obtained.

$$A(a,b) = \begin{cases} 1, & x_b \in W(x_a) \\ 0, & other \end{cases} \tag{4}$$

4. Density Iterative Estimation: Define a diagonal matrix M, whose elements are $M = \mathrm{diag}(\sum_b A_{1b}, \sum_b A_{2b}, \cdots, \sum_b A_{Nb})$, i.e., the degree of each point. Construct the transition probability matrix $Y = M^{-1}A$.

The density iterative update formula is:

$$\gamma_a^{t+1} = \alpha \sum p_{ab} \gamma_b^t + (1-\alpha)\gamma_a^0 \tag{5}$$

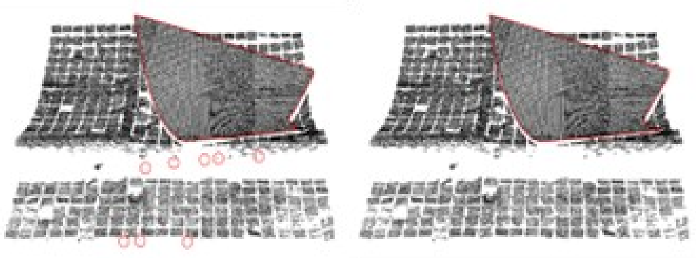
where γ_b^t is the density of node b at time t; $b \in W(x_a)$, p_{ab} is an element in the Markov transition probability matrix Y. After this iterative process converges, the density vector can be analytically expressed as:

$$Den = (I - \alpha Y)^{-1}(1-\alpha)\lambda^0 \tag{6}$$

where $\gamma^0 = (\gamma_1^0, \gamma_2^0, \cdots, \gamma_N^0)^T$.

5. Bad Point Identification and Removal: Set a density threshold; points with a converged density Den below this threshold are identified as bad points and are directly removed.

This algorithm is particularly effective for sparse and structurally separated outliers (e.g., bad points caused by measuring the lower-die surface near plate boundaries), whose local neighborhood density is significantly lower than that of inliers. Its time complexity primarily depends on the distance matrix calculation and the density iteration process. As shown in Fig. 3, after processing by this algorithm, edge noise points in the hull curved plate point cloud are successfully filtered out, providing better initial conditions for subsequent point cloud segmentation processing.



(a) (b)

Figure 3: Comparison before and after bad points removal. (**a**) Before filtering. (**b**) After filtering.

These results indicate that the density-based preprocessing is particularly effective for sparse and structurally separated outliers caused by structured-light measurement near plate boundaries, such as points mistakenly sampled from the lower-die surface. In this case, the outliers form sparse clusters with significantly lower neighborhood density than inliers, which matches the core assumption of the method. For purely random Gaussian perturbations around the true surface, the neighborhood density is usually similar to that of inliers, and the proposed density criterion is therefore not intended as a general Gaussian denoiser. In our pipeline, moderate random perturbations are instead alleviated by MLS-based smoothing, local plane fitting for normals, and the point-to-plane metric used in evaluation. The method can also remove isolated impulse-like points when they remain sparsely distributed, but its performance may degrade when outliers become dense and spatially entangled with the inlier surface, or when sampling becomes extremely sparse and local geometry is severely corrupted.

3.2 Geometric Feature Extraction

High-precision coarse registration relies on highly distinctive geometric features. However, the poor quality of point cloud data from novel structured-light cameras, combined with the gentle curvature of ship plates and the tendency of edge features to become fragmented or blurred in processing environments, leads to insufficient robustness when directly applying existing feature extraction methods (such as general SAM edge detection or standard curvature estimation). This makes it difficult to generate stable and reliable features for registration. Therefore, this section aims to address the aforementioned challenges by introducing a series of targeted optimizations to existing methods, in order to achieve robust extraction of edges, curvature, and normal vectors, thereby laying the foundation for subsequent coarse registration. These geometric cues are extracted not for visualization but to support a deterministic and stable coarse initialization on smooth ship-plate surfaces where generic keypoint–descriptor matching can be unreliable.

3.2.1 SAM-Based Edge Extraction

This paper proposes an optimized edge extraction algorithm based on SAM. The algorithm applies the Canny edge detection algorithm to convert the region mask of the hull curved plate into an edge mask. Then, utilizing corner extraction and edge segmentation sub-modules, it first extracts the corner points of the edge mask, and subsequently segments the edge mask into multiple curves based on the extracted corners. Afterwards, each segmented curve is individually fitted using polynomial fitting to generate a complete and smooth hull curved plate edge.

A key element of this edge-extraction procedure is the concentric-circle-based corner detector shown in Fig. 4. Its principle is that for an edge pixel near a corner point, as shown in Formula (7), the angles (θ_1, θ_2) defined by its two concentric circles with radii (r_1, r_2) are both less than the threshold θ_τ, thereby enabling the robust preliminary screening of candidate pixels.

$$\theta_1 < \theta_\tau \; and \; \theta_2 < \theta_\tau \tag{7}$$

Among the preliminarily screened candidate pixels, by locating the edge endpoints, calculating the perpendicular bisector of the line connecting these endpoints and intersecting it with the edge, the pixel with the smallest angle (θ_i, shown in Formula (8)) in the intersection set is ultimately identified as the precise corner point.

$$\theta_i = arccos \frac{\overrightarrow{P_iP_1} \cdot \overrightarrow{P_iP_2}}{\left\|\overrightarrow{P_iP_1}\right\| \cdot \left\|\overrightarrow{P_iP_2}\right\|} \tag{8}$$

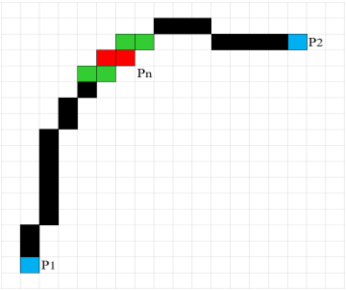


Figure 4: Schematic diagram of corner point extraction.

Another key step of the proposed algorithm is to automatically segment the edge mask into multiple independent curves based on the extracted corner points, to facilitate subsequent curve fitting. For each pixel point Pe on the edge, the distances from Pe to two adjacent corner points (e.g., C_1 and C_2) are first calculated. According to Formula (9), if $\left\|\overrightarrow{C_1P_e}\right\|$ is less than $\left\|\overrightarrow{C_2P_e}\right\|$, the angle θ_e between the two vectors $\overrightarrow{C_1C_2}$ and $\overrightarrow{C_1P_e}$ is calculated. Conversely, if $\left\|\overrightarrow{C_1P_e}\right\|$ is greater than or equal to $\left\|\overrightarrow{C_2P_e}\right\|$, the angle θ_e between the two vectors $\overrightarrow{C_2C_1}$ and $\overrightarrow{C_2P_e}$ is calculated. If Pe is located on the edge C_1C_2, the angle θ_e is relatively small; whereas if Pe is located on the other three edges, the angle θ_e is relatively large. Therefore, by setting an appropriate threshold θ_ϵ, different edges can be accurately segmented using the criterion $\theta_e < \theta_\epsilon$. In practice, θ_e can be set according to the expected corner sharpness and edge mask quality: a smaller θ_e enforces stricter edge separation (higher precision but risk of under-segmentation under noise), while a larger θ_e increases tolerance to noisy edges but may introduce edge mixing. In all experiments, θ_e was fixed once selected and was not retuned for individual plate instances, so the reported performance reflects a unified operating point rather than case-by-case parameter fitting.

The proposed edge segmentation assumes that a sufficient number of corner points can be detected from the edge mask. If corners are missing due to occlusion, incomplete scanning, or heavy boundary corruption, the segmentation may fail and thus degrade coarse initialization. In such cases, a practical fallback is to replace corner-based segmentation with robust boundary fitting (e.g., RANSAC line/curve fitting on boundary candidates or convex-hull-based segmentation). More resilient boundary extraction under missing-corner scenarios will be investigated in future work.

The edge extraction effect is shown in Fig. 5.

$$\theta_e = \begin{cases} arccos\dfrac{\overrightarrow{C_mP_e}\cdot\overrightarrow{C_mC_n}}{\left\|\overrightarrow{C_mP_e}\right\|\cdot\left\|\overrightarrow{C_mC_n}\right\|} & if\ \left\|\overrightarrow{C_mP_e}\right\| < \left\|\overrightarrow{C_nP_e}\right\| \\ arccos\dfrac{\overrightarrow{C_nP_e}\cdot\overrightarrow{C_nC_m}}{\left\|\overrightarrow{C_nP_e}\right\|\cdot\left\|\overrightarrow{C_nC_m}\right\|} & if\ \left\|\overrightarrow{C_mP_e}\right\| \geq \left\|\overrightarrow{C_nP_e}\right\| \end{cases} \tag{9}$$

where C_m and C_n are any two adjacent corner points.

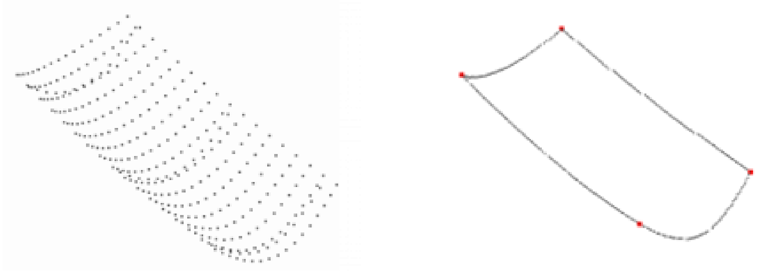

(a) (b)

Figure 5: Boundary extraction results. (**a**) Measured point cloud. (**b**) Boundary and corner points of measured point cloud.

3.2.2 Curvature Estimation via *Moving Least Squares*

Curvature is a key intrinsic feature describing the local bending degree of a surface, providing highly discriminative matching basis for the subsequent coarse registration based on curvature extreme points. Addressing the characteristics of gentle curvature in ship plates and noisy point clouds, directly applying standard curvature estimation algorithms easily leads to unstable features. Therefore, this paper implements and optimizes a robust principal curvature estimation algorithm based on Moving Least Squares (MLS). Its core process consists of three steps: normal curvature estimation, least squares fitting, and principal curvature solving.

Firstly, perform normal vector estimation. For a point p in the point cloud and one of its neighborhood points q_i, a normal curvature value k_n^i can be estimated based on their relative position and the relationship with the normal vector. As shown in Fig. 6, by establishing a local coordinate system $\{p, X, Y, N\}$ at point p (where N is the normal vector), q_i can be projected onto this reference plane. The normal curvature k_n^i of point p relative to q_i can be determined by the geometric relationship of their angles. The calculation formula is:

$$k_n^i = -\frac{sin\,\beta}{|\,pq_i\,|\,si\,n\,\alpha} \tag{10}$$

where α is the angle between vector N and vector pq_i, and β is the angle between vector N and the normal vector M_i at point q_i.

Secondly, perform principal curvature fitting. After obtaining the normal curvature values k_n^i corresponding to multiple neighborhood points, solving for the principal curvatures is transformed into an optimization problem. According to Euler's formula in differential geometry, the normal curvature k_n^i exhibits an inherent relationship with the principal curvatures k_1, k_2, and the principal direction θ:

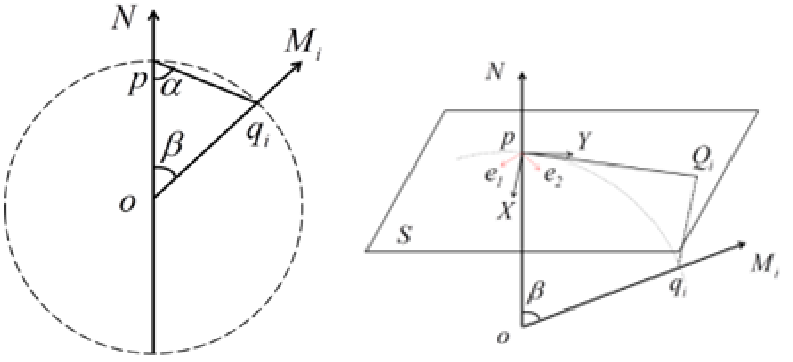
$$k_n^i = k_1\,cos^2\,(\theta_i + \theta) + k_2\,sin^2\,(\theta_i + \theta) \tag{11}$$

where θ_i is the angle between the projected point and the X-axis of the local coordinate system. This paper solves this equation by constructing a linear least squares problem $\underset{\mu}{min}||M\mu - R||^2$, where the design matrix M is composed of θ_i, and the observation vector R is composed of k_n^i.

Finally, perform solution via eigenvalue decomposition. Solving the aforementioned least squares problem yields the parameter vector $\mu = (A, B, C)^T$. Based on this, construct the matrix $W = \begin{bmatrix} A & B \\ B & C \end{bmatrix}$. The two eigenvalues of this matrix are the principal curvatures k_1 and k_2 at point p, and its eigenvectors, after a coordinate transformation, represent the principal directions. The calculation formulas are as follows:

$$\begin{bmatrix} A & B \\ B & C \end{bmatrix} = \begin{bmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \tag{12}$$

This study does not propose new curvature theories. Instead, it successfully applies and implements a complex curvature estimation pipeline tailored to the characteristics of ship plate point clouds. We focused on optimizing key parameters such as neighborhood search, normal projection, and fitting strategies, ensuring the algorithm's robustness and stability in industrial scenarios characterized by low curvature and high noise. This provides reliable feature assurance for subsequent coarse registration.



(a) (b)

Figure 6: Schematic diagram of curvature estimation. (**a**) Triangle defined by osculating circle, neighboring points, and normal vectors. (**b**) Local coordinate system L.

3.2.3 Normal Vector Calculation via *Local Plane Fitting*

The normal vector is a crucial geometric feature in point cloud data and can be used for subsequent point cloud registration. However, normal vectors cannot be obtained directly from measuring equipment and must be estimated from the initial point cloud.

The method based on local plane fitting is the most widely used normal vector calculation method. This method has a relatively simple algorithm and high efficiency. s_i is a data point in the 3D point cloud. A least squares local plane $\boldsymbol{S}$ is fitted to the $\boldsymbol{m}$ neighborhood points around s_i, satisfying the condition that the sum of the distances from s_i and its neighbors to the least squares local plane $\boldsymbol{S}$ is minimized. The fitted least squares local plane $\boldsymbol{S}$ can be represented as:

$$S(l,\mathrm{d}) = \arg min \sum_{i=1}^{k} (l \cdot s_i - d)^2 \tag{13}$$

where $\boldsymbol{l}$ is the normal vector of the least squares local plane $\boldsymbol{S}$; $\boldsymbol{d}$ is the distance from the coordinate origin to the fitted plane; The neighborhood covariance matrix $\boldsymbol{M}$ for the point can be calculated as:

$$M = \frac{1}{n}\sum_{j=1}^{k} \left(s_{ij} - \mathrm{G}_i\right) \cdot \left(s_{ij} - \mathrm{G}_i\right)^T \tag{14}$$

where $\boldsymbol{n}$ is the number of neighborhood points; G_i is the centroid of the least squares local plane $\boldsymbol{S}$;

$$M = [\mathrm{e}_1, \mathrm{e}_2, \mathrm{e}_3] \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \tag{15}$$

Since the covariance matrix $\boldsymbol{M}$ is a symmetric positive semi-definite matrix, its three eigenvalues are non-negative. λ_1, λ_2, λ_3 are the three eigenvalues of matrix $\boldsymbol{M}$. The eigenvector e_3 corresponding to the eigenvalue λ_3 can approximately represent the normal vector l_i at point s_i. The calculation of the surface normal vector is shown in Fig. 7.

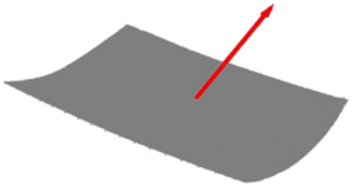


Figure 7: Surface normal vector calculation.

3.3 Coarse Registration Based on Geometric Features

Ship plates often exhibit large smooth areas with low curvature variation, where purely keypoint–descriptor–based global initialization may become unreliable due to insufficient distinctive correspondences under noise and partial observation. To improve stability on such feature-sparse surfaces, we initialize the alignment using global structural cues that remain stable under industrial disturbances. Specifically, we exploit curvature extreme points obtained from MLS-based curvature estimation and enforce a consistent orientation through surface normal matching. This "translate-then-rotate" strategy provides a deterministic and physically interpretable initial pose for subsequent fine registration. The overall workflow is shown in Fig. 8. Note that our coarse initialization is not based on SAC-IA-style random sampling of generic keypoints. Instead, it uses a deterministic "curvature-extrema + normal-consistency" cue that remains stable on smooth ship-plate surfaces under our sensing setup, thereby providing a reproducible initial pose for the subsequent ICP refinement.

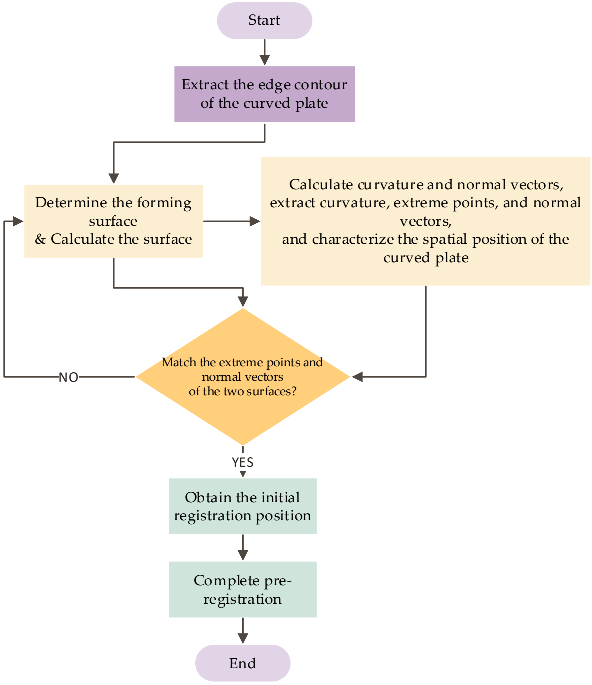


Figure 8: Workflow of point cloud coarse registration method.

Using the curvature calculation method based on Moving Least Squares (MLS), the curvature is computed for points in the two point clouds P and Q respectively, yielding the curvature extreme values K_{E_P} and K_{E_Q} for each point cloud.

$$K_{E_P} = \max_{p_i \in P} |K_{p_i}| \tag{16}$$

$$K_{E_Q} = \min_{p_i \in P} K_{q_i} \tag{17}$$

where E_P and E_Q are the curvature extremum points of point cloud P and point cloud Q, respectively; p_i is a point in point cloud P, $i = 1,2, \dots N_P$; q_i is a point in point cloud Q, $i = 1,2, \dots N_Q$.

Based on the relative positions of the curvature extreme points in the two point clouds, the formed point cloud P is translated so that the positions of the curvature extreme points from both clouds coincide. This yields the spatial coordinates of the translated point cloud P.

$$(\mathrm{x}_i, \mathrm{y}_i, \mathrm{z}_i) = (\mathrm{x}_i, \mathrm{y}_i, \mathrm{z}_i) + \left[\left(x_{E_Q}, y_{E_Q}, z_{E_Q}\right) - \left(x_{E_P}, y_{E_P}, z_{E_P}\right)\right] \tag{18}$$

where $i = 1,2, \dots N_P$.

Compute the normal vectors $\overrightarrow{\mathrm{l}_P}$ and $\overrightarrow{\mathrm{l}_Q}$ for point cloud P and point cloud Q, respectively. Calculate the dot product $\overrightarrow{\mathrm{l}_P} \cdot \overrightarrow{\mathrm{l}_Q}$ to determine the normal vector direction. If $\overrightarrow{\mathrm{l}_P} \cdot \overrightarrow{\mathrm{l}_Q} < 0$, reverse the direction of $\overrightarrow{\mathrm{l}_P}$ and $\overrightarrow{\mathrm{l}_Q}$; if $\overrightarrow{\mathrm{l}_P} \cdot \overrightarrow{\mathrm{l}_Q} > 0$, keep the current normal vector directions. After obtaining the corrected surface normals calculate the angle between the two normal vectors.

$$\theta = arcco\,s\left(\overrightarrow{\mathrm{l}_P} \cdot \overrightarrow{\mathrm{l}_Q}\right) \tag{19}$$

After the curvature extreme points coincide, rotate the formed surface point cloud around a certain rotation axis by an angle θ to align the normal vectors of the two point clouds. The rotation axis is defined as the unit vector starting from the curvature extreme point. The rotation vector is:

$$\overrightarrow{\mathrm{H}} = \frac{\overrightarrow{\mathrm{I}_P} \times \overrightarrow{\mathrm{I}_Q}}{\left|\overrightarrow{\mathrm{I}_P} \times \overrightarrow{\mathrm{I}_Q}\right|} \tag{20}$$

$$(\mathrm{x}_i, \mathrm{y}_i, \mathrm{z}_i) = (\mathrm{x}_{EP}, \mathrm{y}_{EP}, \mathrm{z}_{EP}) + [(\mathrm{x}_i, \mathrm{y}_i, \mathrm{z}_i) - (\mathrm{x}_{EP}, \mathrm{y}_{EP}, \mathrm{z}_{EP})] \cdot \overrightarrow{\mathrm{H}} \tag{21}$$

Here, $(\mathrm{x}_i, \mathrm{y}_i, \mathrm{z}_i)$ represents the coordinates of a node in point cloud P after the curvature extreme point coincidence process; $(\mathrm{x}_{EP}, \mathrm{y}_{EP}, \mathrm{z}_{EP})$ represents the coordinates of the curvature extreme point in point cloud P; $[(\mathrm{x}_i, \mathrm{y}_i, \mathrm{z}_i) - (\mathrm{x}_{EP}, \mathrm{y}_{EP}, \mathrm{z}_{EP})].\overrightarrow{\mathrm{H}}$ denotes the node coordinates after the vector difference $[(\mathrm{x}_\mathrm{i}, \mathrm{y}_\mathrm{i}, \mathrm{z}_\mathrm{i}) - (\mathrm{x}_\mathrm{EP}, \mathrm{y}_\mathrm{EP}, \mathrm{z}_\mathrm{EP})]$ is rotated by angle θ around the rotation axis H.

The core innovation of this study lies in proposing a fast coarse registration framework based on curvature extrema and surface normal vectors. This method reduces the dependency on manual initial pose estimation by providing a deterministic and physically interpretable initialization for subsequent fine registration. Nevertheless, extremely large misalignments, highly symmetric geometries, or severely incomplete measurements may lead to ambiguous curvature extrema or unreliable normals, which can degrade coarse initialization and require additional safeguards. Overall, the proposed coarse registration is computationally efficient and provides a sufficiently accurate initial transformation for fine registration, reducing the risk of ICP-style refinement getting trapped in poor local minima due to an improper initial pose.

The coarse alignment obtained above provides a feasible initial pose, but fine registration is still required to reach sub-millimeter accuracy for closed-loop compensation. In our data, classical ICP variants often converge slowly because correspondence updates form a fixed-point iteration that may require many steps even under a good initialization. Moreover, residual noise and imperfect correspondences can further slow convergence. Therefore, we introduce Anderson acceleration to the ICP iteration, implemented in the SE(3) parameter space to preserve rigid-motion validity.

3.4 Fine Registration with Anderson-Accelerated ICP

Coarse registration provides a good initial pose. However, to achieve the high precision required by the forming process, fine registration is essential. The Iterative Closest Point (ICP) algorithm and its variants are the mainstream solutions to this problem. However, their inherent drawbacks—sensitivity to initial values, tendency to fall into local optima, and slow convergence speed—limit their application in industrial scenarios. To break through these bottlenecks, this section first introduces engineering optimizations to the fundamental components of the traditional ICP algorithm to enhance its computational efficiency. Furthermore, it proposes the Anderson-Accelerated ICP (A-ICP) algorithm, which significantly improves convergence behavior and robustness for practical ship-plate registration.

3.4.1 Efficient Implementation of the Basic ICP Framework

The traditional ICP algorithm aligns point clouds by iteratively performing “Nearest Point Search (Correspondence)” and “Motion Estimation (Registration)”. Its core shortcomings lie in its high computational cost and dependence on initial values. To achieve efficient and accurate registration, this paper implements two key optimizations to the basic ICP framework:

Firstly, accelerate the nearest neighbor search using a K-D tree. In each ICP iteration, the algorithm needs to find the nearest neighbor point q_i in the target point cloud Q for every point p_i in the source point cloud P. This paper employs the K-D tree data structure to preprocess the target point cloud Q. The specific implementation workflow is as follows:

Preprocessing Stage: Before iterations begin, a K-D tree index is constructed once for the target point cloud Q. The time complexity of this step is $O(N\ log\ N)$.

Iterative Query Stage: In each iteration, for the M points in P, batch nearest neighbor queries are performed using the pre-built K-D tree index. This reduces the average complexity per query from $O(N)$ for brute-force search to $O(log\ N)$. Consequently, the total complexity of the correspondence search per iteration is optimized from $O(M \times N)$ to $O(M \times log\ N)$.

Secondly, solving for the optimal transformation based on SVD. After obtaining the point correspondences $\{(p_i,q_i)\}$, the optimal rotation matrix $\boldsymbol{R}$ and translation vector $\boldsymbol{t}$ need to be solved to minimize the mean square error. This paper employs a closed-form solution based on SVD, and its implementation steps are efficient and robust:

Compute Centroids: Calculate the centroids μ_p and μ_q of the source and target corresponding point sets, respectively.

Construct the Covariance Matrix: Compute the demeaned coordinates and construct the 3×3 covariance matrix $\mathrm{H} = \sum_i\left[(p_i - \mu_p)(q_i - \mu_q)^{\mathrm{T}}\right]$.

SVD Decomposition: Perform Singular Value Decomposition (SVD) on matrix H ($\mathrm{H} = \mathrm{U\Sigma V^T}$). Decomposing a fixed-size 3×3 matrix is an $O(1)$ constant-time operation, which is computationally highly efficient.

Solve the Transformation: The optimal rotation matrix $\boldsymbol{R}$ is given by $\mathrm{R} = \mathrm{VU^T}$, and the optimal translation vector t is given by $\mathrm{t} = \mu_q - \mathrm{R}\mu_p$.

We seamlessly integrated the K-D tree and the SVD solver into the ICP iterative loop, constructing a high-performance basic ICP framework. This framework resolves the efficiency bottleneck of correspondence search and the issues of accuracy and efficiency in optimal transformation estimation. It lays a solid foundation for the subsequent introduction of the core innovation—Anderson acceleration—ensuring the algorithm's feasibility for large-scale point clouds.

3.4.2 Anderson-Accelerated ICP (A-ICP) Algorithm

Although the aforementioned optimizations improve efficiency, they still fail to address the fundamental issues of the traditional ICP's slow convergence speed and tendency to fall into local optima. Inspired by the field of numerical analysis, this paper innovatively introduces the Anderson Acceleration mechanism into the ICP framework, proposing the A-ICP algorithm.

Anderson Acceleration is a technique used to accelerate the convergence of fixed-point iterations. Its core idea is to utilize historical information from the current iteration and the previous m iterations to predict a new iteration value closer to the final solution through extrapolation, thereby significantly reducing the number of iterations required.

However, directly applying Anderson Acceleration to the rotation matrix $\boldsymbol{R}$ and translation vector $\boldsymbol{t}$ of ICP is invalid because the linear combination of rotation matrices no longer satisfies the orthogonality constraint. The core technical innovation of this paper lies in the re-parameterization of the pose transformation via Lie Algebra:

Parameterization: The rigid transformation matrix from each iteration, $\mathrm{T^k} = [\mathrm{R^k}|\mathrm{t^k}]$, is mapped to a vector $\mathrm{X}^{(k)}$ in the Lie algebra $se(3)$ space. Linear operations within this vector space are safe.

Anderson Acceleration: Within the $\boldsymbol{X}$ space, utilizing the historical iteration solutions $\mathrm{X}^{(k)}, \mathrm{X}^{(k-1)}, \ldots, \mathrm{X}^{(k-\mathrm{m})}$ and their corresponding residuals, a least squares problem is constructed to solve for a set of optimal coefficients θ_{j}^{*}. These coefficients are used to extrapolate a new accelerated value $\mathrm{x}_{\mathrm{AA}}^{(\mathrm{k}+1)}$.

Mapping Recovery: The accelerated parameter $\mathrm{X_{AA}}$ is recovered into a valid rigid transformation matrix $\mathrm{T_{AA}}$ via the exponential map, which is used to update the point cloud.

This process reformulates the ICP iteration into a safely accelerable format: $X^{(k+1)} = \overline{G}_{ICP}\left(X^{(k)}\right)$, and optimizes it via Anderson Acceleration. The algorithm workflow is shown in Fig. 9.

The A-ICP algorithm is the core innovation of this study. By parameterizing rigid motions in the SE(3) Lie algebra, Anderson acceleration can be applied safely in a linear vector space, and a valid rigid transformation is recovered via the exponential map. Practically, the method leverages a short history of residuals to extrapolate a better update direction, which typically improves convergence behavior and reduces the refinement effort required by ICP-style registration. In our ship-plate experiments, A-ICP consistently improves registration accuracy and convergence efficiency over classical ICP variants while maintaining a lightweight, training-free implementation suitable for industrial deployment. In addition, the SE(3) exponential/log mappings in our implementation operate on 6D vectors and 4 × 4 matrices with closed-form computations, so the added per-iteration overhead remains bounded and small relative to repeated correspondence search.

Taken together, the proposed method forms a progressive registration workflow tailored to industrial ship-plate forming. Density-based preprocessing first removes boundary-related structural outliers introduced by the sensing geometry. Deterministic coarse initialization then uses boundary, curvature, and normal cues to provide a stable initial pose on feature-sparse surfaces. Finally, A-ICP improves the efficiency and robustness of rigid refinement in the SE(3) parameter space. At the system level, the combined pipeline is designed to alleviate three practical bottlenecks in ship-plate registration—structural outliers, weak initialization cues, and iterative refinement efficiency. The present results support this overall design rationale and indicate the practical value of the integrated workflow for ship-plate registration under industrial sensing conditions.

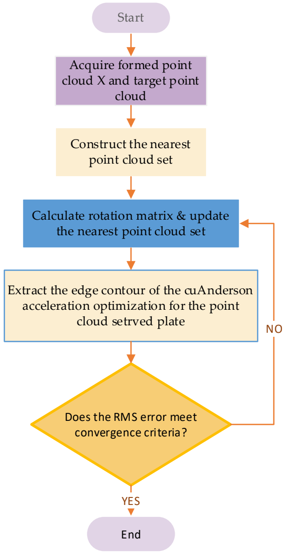


Figure 9: Flowchart of the A-ICP algorithm.

4 Experiments and Analysis

4.1 Registration Error Evaluation Criteria

To validate the effectiveness and robustness of the algorithm, experiments were conducted on a computer equipped with an Intel i5-7200U processor and 8GB RAM, under the software environment of Windows 10 and Matlab.

To objectively evaluate the registration performance, the "point-to-plane" distance was adopted as the evaluation metric for registration error. For each point in the target point cloud, a least-squares plane is fitted within its local neighborhood. The distance from the corresponding point in the source point cloud to this plane is calculated. The average distance of all points is taken as the overall registration error, calculated by the following formula:

$$e = \frac{1}{\mathrm{n}} \sum_{0}^{\mathrm{n}} \frac{\left| aX_{p_i} + bY_{p_i} - Z_{p_i} + c \right|}{\sqrt{a^2 + b^2 + 1}} \tag{22}$$

where (X, Y, Z) are the coordinates of a point in the source point cloud, and $ax + by + cz + d = 0$ is the equation of the plane fitted from the corresponding local neighborhood in the target point cloud. In our implementation, the local plane (and normal) is estimated by PCA-based fitting on the k-nearest neighbors (k = 30) of each point in the target point cloud, and the same neighborhood size is used for all compared methods.

Unless otherwise stated, the same parameter configuration was used across all plate types and all experiments, including the machine-in-the-loop validation, without per-instance tuning. This unified setting was adopted to reduce the risk of parameter overfitting and to ensure that the reported performance differences are not driven by case-specific adjustment. Under this setting, the experimental results are intended to reflect the behavior of the proposed registration pipeline under a consistent engineering configuration. In addition, under fixed point-cloud pairs, parameter settings, and implementation conditions, the proposed offline registration pipeline is deterministic; repeated execution of the same archived case reproduces the same result up to numerical precision.

4.2 Fine Registration Experiments and Analysis

To validate the effectiveness of the proposed pipeline across representative ship-plate geometries, we first visualize the coarse initialization results and then evaluate the fine registration performance on three types of surfaces: sail-shaped, saddle-shaped, and elongated plate segments. Figs. 10–12 show the coarse registration effects for these three plates. The red mesh represents the theoretical surface, while the green mesh represents the formed surface. In the overlapping images of the two surfaces, a greater presence of green areas indicates better coarse registration performance. Although the coarse registration effects vary slightly among the three plate types, overall, the theoretical surface (red mesh) and the formed surface (green mesh) achieve approximate alignment after coarse registration, providing a good initial pose for fine registration.

After fine registration using the A-ICP algorithm, the two surfaces show close overlap, indicating high registration accuracy. Across the three plate geometries, the qualitative results suggest that the proposed pipeline is adaptable to different curvature patterns commonly encountered in ship-plate forming.

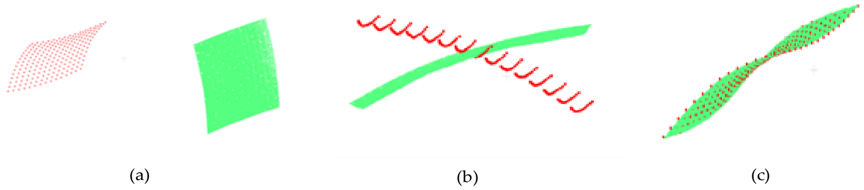

(a) (b) (c)

Figure 10: Coarse registration results of saddle-shaped plate. (**a**) Initial position of saddle-shaped plate registration. (**b**) Coarse registration effect of saddle-shaped plate. (**c**) Fine registration effect of saddle-shaped plate.

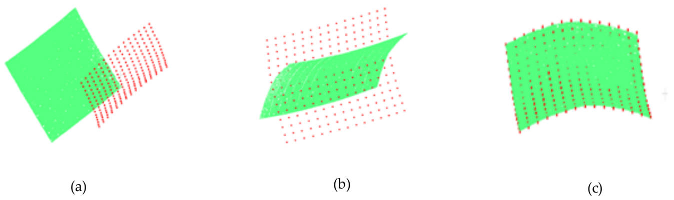


Figure 11: Coarse registration results of sail-shaped plate. (**a**) Initial position of sail-shaped plate registration. (**b**) Coarse registration effect of sail-shaped plate. (**c**) Fine registration effect of sail-shaped plate.

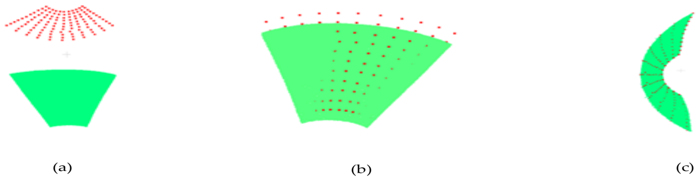


Figure 12: Coarse registration results of long-shaped plate. (**a**) Initial position of long-shaped plate registration. (**b**) Coarse registration effect of long-shaped plate. (**c**) Fine registration effect of long-shaped plate.

This paper compares the performance of the proposed registration algorithm with classical ICP and Tr-ICP. Tr-ICP is a widely recognized ICP variant that improves robustness by trimming unreliable correspondences in overlapping regions. Table 1 summarizes the average point-to-plane registration errors of the three methods. For fairness, all compared fine-registration methods (ICP, Tr-ICP, and A-ICP) were evaluated under the same preprocessing, coarse initialization, parameter configuration, and point-to-plane error setting. As shown in Table 1, both Tr-ICP and A-ICP achieve lower average point-to-plane registration errors than conventional ICP on all three plate types, and A-ICP yields the lowest error in each case. Relative to classical ICP, A-ICP reduces the average registration error from 4.66 to 2.88 mm (38.2%) on the sail-shaped plate, from 4.35 to 2.63 mm (39.5%) on the saddle-shaped plate, and from 4.78 to 2.98 mm (37.7%) on the long-shaped plate. Relative to Tr-ICP, the corresponding reductions are 10.3%, 31.5%, and 21.4%, respectively. These results provide quantitative evidence for the overall effectiveness of the proposed pipeline under the tested setting.

These comparisons are intended to evaluate the registration performance of the complete pipeline under representative ship-plate geometries. The observed improvements therefore support the practical effectiveness of the integrated workflow, including preprocessing, deterministic initialization, and A-ICP-based refinement, under representative ship-plate geometries.

Table 1: Comparison of average point-to-plane registration error (mm) among ICP variants.

Plate Type	ICP (mm)	Tr-ICP (mm)	A-ICP (mm)
Sail-shaped plate	4.66	3.21	2.88
Saddle-shaped plate	4.35	3.84	2.63
Long-shaped plate	4.78	3.79	2.98

From the error-analysis perspective, the differences among the three plate types are also compatible with their geometric characteristics. The sail-shaped and saddle-shaped plates provide relatively clearer global curvature trends for deterministic initialization, whereas the long-shaped plate contains more elongated and locally less distinctive regions, making refinement more sensitive to residual boundary disturbance and local geometric ambiguity. Nevertheless, A-ICP still achieves the lowest average registration error in all three cases. This pattern is consistent with the overall empirical behavior of the proposed pipeline across different plate geometries.

From the efficiency perspective, the dominant computational burden in ICP-style registration mainly comes from repeated nearest-neighbor search and rigid transformation estimation. In Tr-ICP, trimming introduces only moderate additional overhead while improving robustness to mismatched correspondences. In contrast, the Anderson step in A-ICP only solves a small least-squares problem in a low-dimensional space, whereas the dominant nearest-neighbor search remains unchanged. Therefore, when A-ICP reduces the refinement effort needed for convergence, a corresponding efficiency advantage in refinement is expected.

While Table 1 establishes the offline accuracy advantage of A-ICP, industrial ship-plate forming ultimately requires the registration module to operate stably within a forming-and-measurement loop and to provide CAD-referenced surface deviation statistics for closed-loop evaluation. We therefore further validate the proposed method through an on-machine deployment on the SKWB-2500 platform, where A-ICP is used for CAD-to-scan alignment under real industrial sensing conditions.

*4.3 Engineering Deployment and Closed-Loop Forming Validation on SKWB-*2500

In closed-loop digital forming of ship plates, point cloud registration serves as the interface between 3D measurement and compensation update rather than as an isolated vision task. After each forming cycle, the scanned plate surface must be aligned to the target CAD model so that the surface deviation field can be computed and fed to the next compensation step. Therefore, the practical value of a registration method lies not only in offline accuracy, but also in whether it can provide stable CAD-to-scan alignment within a real forming-and-measurement loop under industrial sensing conditions. To verify this deployment-oriented requirement, we integrated the proposed A-ICP module into an upgraded SKWB-2500 3D CNC ship plate bending machine as the registration component of the measurement pipeline. Figs. 13 and 14 illustrate the SKWB-2500 platform and the practical forming-and-measurement procedure.

Figure 13: Model SKWB-2500 Plate Bending Machine.

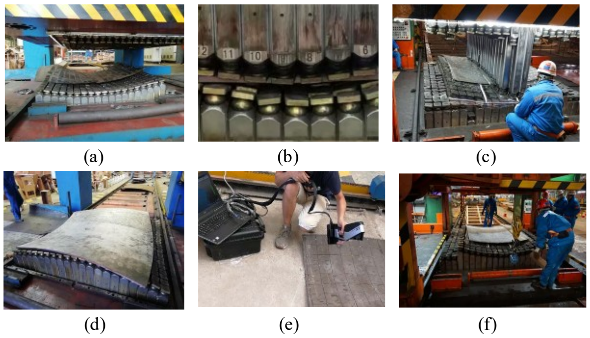


Figure 14: Plate Forming Experiment. (**a**) Lower Die Shape Adjustment for Saddle-shaped Plate. (**b**) Before Processing of Saddle-shaped Plate. (**c**) Processing Compensation for the Other Bending Direction During Saddle-shaped Plate Processing. (**d**) After Completion of Saddle-shaped Plate Processing. (**e**) Measurement of Saddle-shaped Plate with Laser Measuring Instrument. (**f**) Removal of Processed Plate with Lifting Tool.

Two representative double-curvature workpieces, namely sail-shaped and saddle-shaped Q235 plates (5 mm thickness, 840 mm × 840 mm), were used for validation. After each compensation cycle, the formed surface was scanned by a 3D laser measurement system and aligned to the theoretical CAD surface using the proposed A-ICP algorithm. The aligned result was then used to compute the CAD-referenced surface deviation required for closed-loop evaluation. Three representative closed-loop operating conditions were considered, corresponding to a conventional iterative compensation mode and two data-driven compensation modes in the upgraded machine workflow. These conditions were introduced to examine whether the same registration module can stably support CAD-to-scan alignment and CAD-referenced surface-deviation evaluation under different practical closed-loop forming conditions. Accordingly, the three modes should be understood as different operating conditions of the forming system rather than different variants of the registration algorithm itself.

Figs. 15 and 16 show the formed surfaces obtained under the three representative closed-loop operating conditions, where the target CAD surface is shown in blue and the measured results are shown in different colors. Table 2 reports the MAE and RMSE of the final formed surfaces after CAD-to-scan alignment using the same A-ICP registration module. Since these CAD-referenced surface-error statistics are defined only after alignment, the purpose of Table 2 in the present study is not to compare the three operating modes in terms of better or worse numerical performance, but to verify whether the same registration stage can remain operationally effective when embedded in representative industrial closed-loop conditions. In other words, Table 2 is used to validate the practical performance of the proposed registration module within the external machine-side workflow, namely, whether it can stably support CAD-to-scan alignment and make downstream CAD-referenced surface-deviation evaluation feasible under real industrial deployment settings.

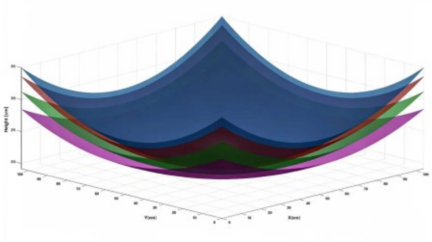


Figure 15: Comparison of formed surfaces for the sail-shaped plate.

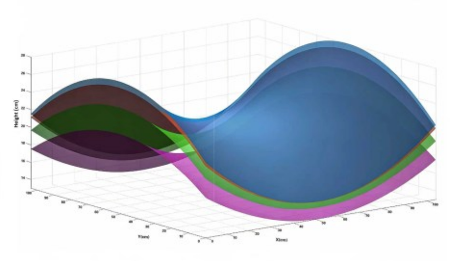


Figure 16: Comparison of formed surfaces for the saddle-shaped plate.

Because the same A-ICP registration module is used in all three representative closed-loop operating conditions, Table 2 further shows that the proposed registration stage is not restricted to a single workflow setting, but can remain effective throughout the current industrial closed-loop framework. Numerically, meaningful MAE/RMSE values are obtained for both the sail-shaped and saddle-shaped plates under all three representative operating conditions, indicating that CAD-referenced surface-deviation evaluation remains feasible across different practical workflow settings when supported by the same registration module. Meanwhile, the progressively improved system-level outcomes observed under more refined external workflow configurations are consistent with the practical compatibility of the proposed registration module with the existing forming-and-compensation process. This observation should not be understood as implying that Table 2 is intended to rank the three operating modes themselves, nor that the observed differences are attributed solely to the registration stage. Rather, for the purposes of the present study, the main implication is that the same registration module can be reliably deployed across representative operating conditions and can consistently provide the alignment basis required for downstream CAD-referenced evaluation in a real industrial workflow. Therefore, the numerical differences among the three modes should be interpreted as condition-specific system outcomes under different workflow settings, while Table 2 itself serves primarily as deployment-oriented validation of the registration stage, complementing the offline registration comparison reported in Table 1.

Table 2: CAD-referenced surface error statistics (MAE/RMSE) obtained after alignment by the same A-ICP registration module under three representative closed-loop operating conditions of the SKWB-2500 system.

Workpiece	Metric	Mode I	Mode II	Mode III
Sail-shaped plate	RMSE (mm)	4.1161	2.0835	0.7301
Sail-shaped plate	MAE (mm)	4.00	1.96	0.68
Saddle-shaped plate	RMSE (mm)	3.4938	1.9323	0.4141
Saddle-shaped plate	MAE (mm)	3.26	1.85	0.38

Mode I, Mode II, and Mode III denote three representative closed-loop operating conditions of the SKWB-2500 forming system, corresponding to stepwise approximation, WOA-CNN-GRU-Attention-based compensation, and dynamic intelligent compensation, respectively. In all three cases, CAD-to-scan alignment is performed using the same A-ICP registration module.

5 Discussion and Conclusions

This work targets point cloud registration in closed-loop ship-plate forming, where boundary-induced structural outliers, feature-sparse geometry, and strict runtime constraints jointly limit practical deployment. To address these challenges in an integrated manner, we combine density-based outlier removal for stabilizing correspondence formation, deterministic geometric-cue-based coarse initialization for reducing initial-pose sensitivity, and Anderson-accelerated ICP in the SE(3) space to improve convergence while preserving rigid-motion validity.

The experimental results support the effectiveness of the proposed design from two complementary perspectives. First, offline experiments on three representative ship-plate geometries show that the proposed pipeline consistently improves registration accuracy over classical ICP variants while also indicating improved convergence behavior. In particular, relative to classical ICP, A-ICP reduces the average registration error by 38.2%, 39.5%, and 37.7% on the sail-shaped, saddle-shaped, and long-shaped plates, respectively, while also outperforming Tr-ICP on all three cases. These observations are consistent with the overall design rationale of the proposed workflow and support its practical effectiveness under the tested conditions.

Second, beyond offline benchmarks, the proposed A-ICP module is validated in an SKWB-2500 machine-in-the-loop workflow, where CAD-to-scan alignment is directly used for closed-loop surface evaluation. This deployment result is important because the primary contribution of this paper is to provide a training-free and geometry-driven registration module that can operate reliably in ship-plate digital forming under proprietary industrial sensing conditions, rather than to present a broad cross-domain registration benchmark.

The scope of the present study should nevertheless be noted. This work is focused on the practical effectiveness of a training-free, geometry-driven registration workflow under representative industrial ship-plate conditions. Learning-based baselines are not benchmarked here because a fair comparison would require an in-domain dataset together with a unified retraining or calibration protocol under the same sensing setup. To avoid per-instance tuning, a single fixed parameter configuration was used throughout the experiments. Accordingly, the present results are intended to support the effectiveness of the complete workflow under representative industrial conditions. Future work will extend the validation to additional industrial parts and sensing scenarios, and will investigate fair in-domain comparisons with learning-based registration models.

Overall, the proposed method provides a practical and deployment-oriented registration solution for CAD-to-scan alignment in ship-plate digital forming, bridging offline point cloud processing and real industrial closed-loop evaluation. Within the above limitations, the current evidence supports the practical effectiveness of the complete pipeline under the tested setting.

Acknowledgement: We sincerely thank the Weihai Institute of Marine Information Science and Technology for their essential support in point cloud data acquisition and experimental validation. We are also grateful to the anonymous reviewers for their insightful comments and constructive suggestions, which significantly improved the quality of this manuscript. Furthermore, we acknowledge the pioneering work of D. G. Anderson in the field of acceleration techniques, which provided fundamental inspiration for the methodological innovations presented in this study.

Funding Statement: The authors gratefully acknowledge support from the Shandong Provincial Key Research and Development Program (Grant No. 2024SFGC0201) and the Central Government-Guided Local Fund Project (Grant No. YDZX2025123).

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Zhonghao Wang and Xin Liu; data collection: Zhonghao Wang, Xin Liu, Yang Zhang, Pengfei Hou, Yabin Li and Haiwen Yuan; analysis and interpretation of results: Yabin Li, Wenlong Sun ,Dongxu Liu and Haiwen Yuan; draft manuscript preparation: Zhonghao Wang and Xin Liu. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The raw point cloud data are confidential and not publicly available due to proprietary industrial constraints. Derived data supporting the findings are available from the corresponding author upon reasonable request.

Ethics Approval: Not applicable.

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

References

1. Ye Y, Song Z. An accurate 3D point cloud registration approach for the turntable-based 3D scanning system. In: Proceedings of the 2015 IEEE International Conference on Information and Automation; 2015 Aug 8–10; Lijiang, China. p. 982–6. doi:10.1109/icinfa.2015.7279429.
2. Wang Y, Shi H, Zhang Y, Zhang D. Automatic registration of laser point cloud using precisely located sphere targets. J Appl Remote Sens. 2014;8(1):083588. doi:10.1117/1.jrs.8.083588.
3. Johnson AE, Hebert M. Using spin images for efficient object recognition in cluttered 3D scenes. IEEE Trans Pattern Anal Machine Intell. 1999;21(5):433–49. doi:10.1109/34.765655.
4. Wang F, Zhao Z. A survey of iterative closest point algorithm. In: Proceedings of the 2017 Chinese Automation Congress (CAC); 2017 Oct 20–22; Jinan, China. p. 4395–9. doi:10.1109/cac.2017.8243553.

5. Saleh AR, Momeni HR. An improved iterative closest point algorithm based on the particle filter and K-means clustering for fine model matching. Vis Comput. 2024;40(11):7589–607. doi:10.1007/s00371-023-03195-0.
6. Cheng Y, Chu H, Li Y, Tang Y, Luo Z, Li S. A hybrid improved SAC-IA with a KD-ICP algorithm for local point cloud alignment optimization. Photonics. 2024;11(7):635. doi:10.3390/photonics11070635.
7. Zhang C, Zhou H, Duan JA. A method for identifying and repairing holes on the surface of unorganized point cloud. Measurement. 2023;210:112575. doi:10.1016/j.measurement.2023.112575.
8. Dey EK, Tarsha Kurdi F, Awrangjeb M, Stantic B. Effective selection of variable point neighbourhood for feature point extraction from aerial building point cloud data. Remote Sens. 2021;13(8):1520. doi:10.3390/rs13081520.
9. Arazm N, Sahab A, Kazemi MF. Noise reduction of SEM images using adaptive Wiener filter. In: Proceedings of the 2017 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom); 2017 Nov 20–22; Phuket, Thailand. p. 50–5. doi:10.1109/cyberneticscom.2017.8311683.
10. Zeng J, Cheung G, Ng M, Pang J, Yang C. 3D point cloud denoising using graph Laplacian regularization of a low dimensional manifold model. IEEE Trans Image Process. 2020;29:3474–89. doi:10.1109/tip.2019.2961429.
11. Zheng Y, Fu H, Au OK, Tai CL. Bilateral normal filtering for mesh denoising. IEEE Trans Visual Comput Graph. 2011;17(10):1521–30. doi:10.1109/tvcg.2010.264.
12. Ren Y, Li T, Xu J, Hong W, Zheng Y, Fu B. Overall filtering algorithm for multiscale noise removal from point cloud data. IEEE Access. 2021;9:110723–34. doi:10.1109/access.2021.3097185.
13. Cao L, Zhuang S, Tian S, Zhao Z, Fu C, Guo Y, et al. A global structure and adaptive weight aware ICP algorithm for image registration. Remote Sens. 2023;15(12):3185. doi:10.3390/rs15123185.
14. Eckart B, Kim K, Kautz J. HGMR: Hierarchical Gaussian mixtures for adaptive 3D registration. In: Computer Vision—ECCV 2018. Cham, Switzerland: Springer International Publishing; 2018. p. 730–46. doi:10.1007/978-3-030-01267-0_43.
15. Sharp GC, Lee SW, Wehe DK. ICP registration using invariant features. IEEE Trans Pattern Anal Machine Intell. 2002;24(1):90–102. doi:10.1109/34.982886.
16. Vrochidis A, Krinidis S, Tzovaras D. Colored ICP: Advancing 2D to 3D shape registration for enhanced accuracy and visualization. In: Proceedings of the 7th International Conference on Algorithms, Computing and Systems; 2023 Oct 19–21; Larissa, Greece. p. 105–10. doi:10.1145/3631908.3631923.
17. Lin B, Tamaki T, Zhao F, Raytchev B, Kaneda K, Ichii K. Scale alignment of 3D point clouds with different scales. Mach Vis Appl. 2014;25(8):1989–2002. doi:10.1007/s00138-014-0633-2.
18. Ren S, Zeng Y, Hou J, Chen X. CorrI2P: Deep image-to-point cloud registration via dense correspondence. IEEE Trans Circuits Syst Video Technol. 2023;33(3):1198–208. doi:10.1109/tcsvt.2022.3208859.
19. Lei H, Jiang G, Quan L. Fast descriptors and correspondence propagation for robust global point cloud registration. IEEE Trans Image Process. 2017;26(8):3614–23. doi:10.1109/tip.2017.2700727.
20. Rousseeuw PJ. Least Median of squares regression. J Am Stat Assoc. 1984;79(388):871–80. doi:10.1080/01621459.1984.10477105.
21. Guan W, Li W, Ren Y. Point cloud registration based on improved ICP algorithm. In: Proceedings of the 2018 Chinese Control and Decision Conference (CCDC); 2018 Jun 9–11; Shenyang, China. p. 1461–5. doi:10.1109/ccdc.2018.8407357.
22. Vizzo I, Guadagnino T, Mersch B, Wiesmann L, Behley J, Stachniss C. KISS-ICP: In defense of point-to-point ICP–simple, accurate, and robust registration if done the right way. IEEE Robot Autom Lett. 2023;8(2):1029–36. doi:10.1109/lra.2023.3236571.
23. Huang YB, Tan ZH, Wang XH, Wang QF, Zhang J, Li JJ. ICP alignment method for multi-view 3D point cloud data based on moving least squares surface. Eng J Wuhan Univ. 2011;44(02):249–53. (In Chinese).
24. Chen XW, Zhu YL, Wu T, Zhang XW. Point cloud registration technology based on SAC-IA and improved ICP algorithm. J Xi’an Polytech Univ. 2017;31(03):395–401. (In Chinese).
25. Wang YJ, Lian TF, Wu MM, Wang GL. Point cloud registration method based on octree and KD-tree index. Surv Mapp Eng. 2017;26(08):35–40. (In Chinese).
26. Li F. Research on improved Rodrigues matrix-ICP point cloud registration algorithm. Mine Surv. 2017;45(02):5–9. (In Chinese).
27. Bai X, Luo Z, Zhou L, Chen H, Li L, Hu Z, et al. PointDSC: Robust point cloud registration using deep spatial consistency. In: Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2021 Jun 20–25; Nashville, TN, USA. p. 443–59. doi:10.1109/cvpr46437.2021.01560.