High-Precision Registration of Grotto Point Clouds Using Multi-Source Data Fusion

Jinru Li [1], Miaole Hou[1, 2], Bo Ning [3], Yong He[3]

[1] School of Geomatics and Urban Spatial Informatics,Beijing University of Civil Engineering and Architecture - 102612 Beijing, China - jrli_2019@163.com
[2] School of Land Engineering, Chang’an University, - 710064 Xi'an ,China - houmiaole@bucea.edu.cn
[3] Yungang Researeh Institute - 037007 Datong, China- 40970156@qq.com, 18734686621@126.com

Keywords: Multi-source data fusion; Point cloud registration; Feature extraction; K-4PCS; Improved ICP; Cave digitization

Abstract

To address the challenges of large initial pose discrepancies in grotto point clouds acquired from multiple sources, complex local geometric structures, significant noise interference, and the tendency of traditional ICP algorithms to fall into local optima, a high-precision point cloud registration method is proposed by integrating feature extraction with the collaborative optimization of coarse and fine registration. This method first performs point cloud preprocessing through voxel downsampling and outlier removal; it then extracts stable feature regions based on normal vector estimation and curvature analysis, and constructs feature representations using FPFH descriptors; building on this, the K-4PCS algorithm is employed to perform coarse registration and obtain optimal initial transformation parameters, followed by fine registration using an improved ICP algorithm combined with KD-tree-based search optimization. The proposed method was validated using the STANFORD DRAGON dataset and the point cloud of the Buddha head statue from Cave 18 of the Yungang Grottoes. The results indicate that the proposed method effectively improves the convergence speed and accuracy of point cloud registration. It demonstrates good stability and applicability in complex cave heritage scenarios and can provide methodological support for the fusion of multi-source point clouds in the digital preservation of cultural heritage.

1 Introduction

3D point clouds can directly represent the spatial geometry of an object's surface and have been widely applied in fields such as remote sensing and mapping, autonomous driving, robotic perception, the digital preservation of cultural heritage, and 3D reconstruction. Since data collected from a single viewpoint often suffers from issues such as occlusion, sparsity, incompleteness, and inconsistent coordinate systems, it is necessary to unify point cloud data acquired from different viewpoints, platforms, or time periods into a single reference coordinate system. Consequently, point cloud registration has become a critical step in point cloud processing, and its accuracy and efficiency directly impact the performance of subsequent modeling, recognition, and analysis tasks(Wan and Wu, 2025). Current point cloud registration methods can be broadly categorized into three types: those based on iterative optimization, those based on feature matching, and those based on deep learning. Iterative optimization methods, such as ICP and its improved variants, offer advantages such as simplicity of implementation and strong local fitting capabilities. However, they generally suffer from issues such as sensitivity to initial pose, a tendency to get stuck in local optima, and insufficient robustness to noise, outliers, and changes in overlap ratio(Song et al., 2021; Li et al., 2024; Wang et al., 2022) . To improve the performance of traditional methods, related research has focused on enhancements to matching strategies, error models, and adaptation to specific application scenarios. Examples include improved methods for CAD models, curved blades, and multi-stage optimization registration(Yang et al., 2026; Yuan et al., 2026; Li et al., 2025), as well as NDT registration methods based on adaptive clustering features(Huang et al., 2025).

To address the strong dependence of traditional iterative methods on initial values, feature-matching approaches based on keypoints and local descriptors have been extensively studied. By extracting stable and discriminative geometric or structural features to establish correspondences, such methods can effectively improve coarse registration performance under conditions of large pose discrepancies. Previous research has extensively explored methods such as 3D-SIFT, normal vectors, curvature features, multi-feature key-points, and structural similarity constraints(Huang et al., 2026; Tu et al., 2025; Ji et al., 2025; Zhang et al., 2025) . Meanwhile, multi-scale feature modeling and multi-feature fusion, as key techniques for enhancing the descriptive capabilities of point clouds, have also been applied in point cloud registration(Zhao et al., 2026; Li et al., 2026). Furthermore, with the widespread adoption of multi-source sensors, the registration of heterogeneous point clouds and multimodal data has emerged as a new research focus. Related studies have combined texture, image features, architectural geometric constraints, and detection features with local geometric descriptions to improve the performance of cross-platform, cross-modal point cloud registration(Xu et al., 2026; Qian et al., 2026; Cao et al., 2026; Yi et al., 2025). In recent years, deep learning methods have opened up new avenues of research for point cloud registration. Through end-to-end feature learning and correspondence modeling, these methods have demonstrated strong robustness and adaptability in complex scenarios(Wang et al., 2026; Zhao et al., 2026; Zhou et al., 2026) . However, existing methods still commonly face challenges such as noise interference, uneven density, repetitive local structures, cross-modal differences, and real-time constraints. Overall, balancing robustness and computational efficiency while ensuring registration accuracy remains a key research direction in point cloud registration.

2 Related work

2.1 Data Preprocessing

https://doi.org/10.5194/isprs-archives-XLIX-B4-2026-687-2026

Before performing initial point cloud registration, some redundant points from the surrounding area are inevitably captured during data acquisition. these points can adversely affect the final registration results, leading to reduced efficiency and accuracy. Therefore, the raw point cloud data should be preprocessed prior to registration. By applying basic filtering to merge and remove outliers, a multi-scale filtering process is established. This effectively reduces the volume of the point cloud data while preserving structural boundaries and detailed sculptural features, thereby further improving registration efficiency.

In the preprocessing stage, a method combining global voxelization for scale unification with adaptive curvature-based sampling for detail enhancement is adopted. Global voxelization sampling reduces the overall number of points while preserving the overall geometric integrity of the grotto object with minimal deformation, thereby improving computational efficiency and maintaining data completeness in subsequent registration and reconstruction processes. Specifically, the point cloud is first partitioned into a three-dimensional voxel grid, where each voxel is regarded as a small cubic unit in 3D space. The centroid of all points within each voxel is then used as an approximate representative point, such that each voxel block is represented by a single center point. Compared with direct voxel-based downsampling, this strategy provides a clearer representation of the overall shape characteristics of the object and yields better performance in preserving and expressing local details.

The point cloud is first subjected to voxel downsampling to achieve data sparsification. Specifically, the centroid of each voxel grid is computed. If the centroid cannot be directly identified as an existing point, the sample point closest to the centroid is selected to replace the voxel centroid, thereby representing all points within that voxel. This downsampling process is performed for all voxel grids until completion. The centroid of a voxel grid is expressed as follows:

$$F_{ik} = \frac{1}{n}\sum_{i}^{k} p_i \tag{1}$$

where k denotes the number of points within each voxel grid, P_i represents an individual data point in the voxel, and F_{ik} denotes the centroid position calculated for that voxel grid.

2.2 Feature extraction

2.2.1 Normal vector estimation: Before conducting the experiment, we know that point cloud data inherently contains coordinate information. Due to variations in data acquisition methods, the specific attributes of point clouds can differ. Generally, before feature extraction, it is necessary to estimate the normal vectors and calculate the curvature of the point cloud data. Normal vector estimation and curvature calculation are among the most important geometric features of point cloud data; they are inherently invariant under rotation and translation, meaning they are subject to rigid transformations. Therefore, in this section of the experiment, we will use local surface fitting methods to calculate the normals of the point cloud.

For a given point $p_i = [x_i, y_i, z_i]^T$ in a point cloud $P = \{p_i \in R^3, i = 1,2 \ldots N\}$, fitting a least-squares plane to that point and the N neighboring points within its neighborhood yields a local subplane F. The formula is calculated as follows:

$$F(n,d) = \underset{(n,d)}{\arg\min} \sum_{i=1}^{N} (n \cdot p_i - d)^2 \tag{2}$$

In the equation, n represents the normal vector of plane F; d represents the distance from plane F to the origin.

Since the local plane F passes through the center of mass of point p_i and its N neighboring points, the formula for calculating it is as follows:

$$\overline{p} = \frac{1}{N+1}\sum_{i=1}^{N+1} p_i \tag{3}$$

The normal vector of the local plane F must satisfy $\|\boldsymbol{n}\|_2=1$,assuming $\boldsymbol{C} = \{ p_i \mid i = 1,2,\cdots N \}$, then construct the covariance matrix C using the following formula:

$$\boldsymbol{C} = \frac{1}{N}\sum_{i=1}^{N} (p_i - \overline{p})(p_i - \overline{p_i})^{\mathrm{T}} \tag{4}$$

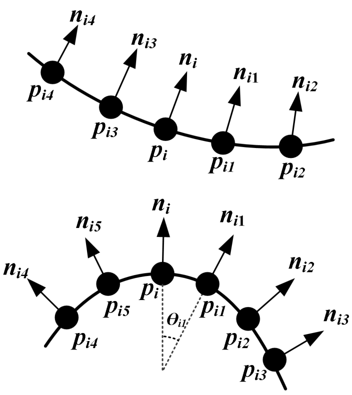
$$\boldsymbol{C}\boldsymbol{n}_i^j = \lambda_i^j \boldsymbol{n}_i^j, j = \{1,2,3\} \tag{5}$$

In the equation, let $\boldsymbol{n}_i^j$ denote the eigenvector corresponding to the eigenvalue λ_i^j of C. Eigenvalues obtained by solving the covariance matrix C λ_i^1 、 λ_i^2 、 λ_i^3 , recorded as $\lambda_i^1 < \lambda_i^2 < \lambda_i^3$, then the smallest eigenvalue λ_i^1 corresponding eigenvector $\boldsymbol{n}_i^1$ the normal vector of p_i.

Based on the distribution characteristics of point cloud normal vectors, the angle between the normal vector of a sample point and that of its neighboring points is defined to describe the curvature of the local surface. Feature regions of the point cloud are extracted, and the average of the dot products between the normal vectors of point pi and its neighboring points is calculated to represent the normal vector angle feature value f.

$$f = \frac{1}{N}\sum_{i=1}^{N} \left| n_i \bullet n_{ij} \right| = \left| n_{xi} \bullet n_{xij} + n_{yi} \bullet n_{yij} + n_{zi} \bullet n_{zij} \right| \bullet \cos(n_i, n_{ij}) \tag{6}$$

In the equation, $f \in [0,1]$, n_i denotes the normal vector of point p_i, and n_{ij} denotes the normal vector of the neighboring point. As shown in Figure 1(a), the angle between the normal vector at point pi and that at a neighboring point is small. By selecting an appropriate threshold ε, if f < ε, the surface curvature at point pi is small, and this region is classified as a non-featured region, with pi designated as a non-featured point; As shown in Figure 1(b), the angle between the normal vector at point π and those of neighboring points is large. If f > ε, the surface curvature at point π is significant, and this region is a feature region; point π is a feature point.



(a) Non-feature areas (b)Feature region

Figure 1: Schematic diagram showing the distribution of normal vectors in different regions

2.2.2 Curvature: The feature regions extracted based on the angles between the normal vectors are approximated using a quadratic surface to estimate the curvature characteristics at each point within the neighborhood. The method for calculating neighborhood curvature is as follows.

1) Select an arbitrary point p in the feature region and define a neighborhood around p. Let the equation of the quadratic surface be:

$$\boldsymbol{r}(u,v)=\sum_{i=0}^{2}\sum_{j=0}^{2}\boldsymbol{Q}_{ij}f_{i,2}\left(u\right)f_{j,2}\left(v\right) \tag{7}$$

In the equation, Q represents the coefficient matrix.

2) Calculate the Gaussian curvature K and mean curvature H at point p on the surface. The normal curvature of the surface in the direction is

$$\boldsymbol{k}_n=\frac{\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\eta}_2\boldsymbol{\mu}}{\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\eta}_1\boldsymbol{\mu}} \tag{8}$$

In the equation, $\boldsymbol{\mu}=\left[du,dv\right]^{\mathrm{T}}$, $\boldsymbol{\eta}_1$ and $\boldsymbol{\eta}_2$ denoting the first and second fundamental quantities.

Find the first and second partial derivatives at point p using the quadratic surface $\boldsymbol{r}(u,v)$

$$\boldsymbol{n}=\frac{\left(\boldsymbol{r}_u\bullet\boldsymbol{r}_v\right)}{\left(\left|\boldsymbol{r}_u\bullet\boldsymbol{r}_v\right|\right)} \tag{9}$$

$$\boldsymbol{r}_u=\frac{\partial r}{\partial u},\quad \boldsymbol{r}_v=\frac{\partial r}{\partial v} \tag{10}$$

$$\boldsymbol{r}_{uu}=\frac{\partial^2 r}{\partial u^2},\boldsymbol{r}_{uv}=\frac{\partial^2 r}{\partial u\partial v},\boldsymbol{r}_{vv}=\frac{\partial^2 r}{\partial v^2} \tag{11}$$

Derive the first fundamental quantity from Equation (10)

$$\boldsymbol{\eta}_1=\begin{bmatrix}\frac{\partial r}{\partial u}\bullet\frac{\partial r}{\partial u} & \frac{\partial r}{\partial u}\bullet\frac{\partial r}{\partial v}\\ \frac{\partial r}{\partial v}\bullet\frac{\partial r}{\partial u} & \frac{\partial r}{\partial v}\bullet\frac{\partial r}{\partial v}\end{bmatrix}=\begin{bmatrix}\boldsymbol{E} & \boldsymbol{F}\\ \boldsymbol{F} & \boldsymbol{G}\end{bmatrix} \tag{12}$$

Derive the second fundamental quantity from Equations (9) and (11)

$$\boldsymbol{\eta}_2=\begin{bmatrix}\boldsymbol{n}\bullet\frac{\partial^2 r}{\partial u^2} & \boldsymbol{n}\bullet\frac{\partial^2 r}{\partial u\partial v}\\ \boldsymbol{n}\bullet\frac{\partial^2 r}{\partial u\partial v} & \boldsymbol{n}\bullet\frac{\partial^2 r}{\partial v^2}\end{bmatrix}=\begin{bmatrix}\boldsymbol{L} & \boldsymbol{M}\\ \boldsymbol{M} & \boldsymbol{N}\end{bmatrix} \tag{13}$$

Solve for K and H using Equations (11) and (12).

$$K=\frac{\boldsymbol{LN}-\boldsymbol{M}^2}{\boldsymbol{EG}-\boldsymbol{F}^2},H=\frac{\boldsymbol{EN}-2\boldsymbol{FM}+\boldsymbol{GL}}{2(\boldsymbol{EG}-\boldsymbol{F}^2)} \tag{14}$$

3) Calculate the principal curvatures k1 and k2 at point p on the surface.

Expand equation (8) to

$$\begin{cases}\left(\boldsymbol{L}-\boldsymbol{k}_n\boldsymbol{E}\right)du+\left(\boldsymbol{M}-\boldsymbol{k}_n\boldsymbol{F}\right)dv=0\\ \left(\boldsymbol{M}-\boldsymbol{k}_n\boldsymbol{F}\right)du+\left(\boldsymbol{N}-\boldsymbol{k}_n\boldsymbol{G}\right)dv=0\end{cases} \tag{15}$$

Eliminating du and dv yields

$$\left|\boldsymbol{\eta}_1\right|\boldsymbol{k}_n^2-\left(\boldsymbol{EN}-2\boldsymbol{FN}+\boldsymbol{GL}\right)\boldsymbol{k}_n+\left|\boldsymbol{\eta}_2\right|=0 \tag{16}$$

Substituting equation (14) into equation (16) yields

$$\left|\boldsymbol{\eta}_1\right|\left(\boldsymbol{k}_n^2+2H\boldsymbol{k}_n+K\right)=0 \tag{17}$$

Due to $\left|\boldsymbol{\eta}_1\right|\neq 0$,the principal curvatures k1 and k2 are obtained by solving equation (17).

$$\begin{cases}k_1=H+\sqrt{H^2-K}\\ k_2=H-\sqrt{H^2-K}\\ K=k_1\bullet k_2,H=\frac{(k_1+k_2)}{2}\end{cases} \tag{18}$$

The average curvature H of the neighborhood of point P is calculated based on the feature regions extracted using the angle between normal vectors, and this value is compared with the curvature threshold δ. If H < δ, points with curvature below the threshold are removed as non-feature regions; conversely, if H > δ, points with curvature above the threshold are retained as feature point sets Pω and Qω.

3 Initial registration

3.1 FPFH Feature Description

After performing calculations based on the normals and curvature of the point cloud, the feature points are described using FPFH. FPFH is a method that constructs and describes the network relationships of feature points primarily by dividing them into neighborhoods. By estimating the normals and calculating the curvature of the feature points, a feature histogram is constructed to quickly determine the correspondences between feature points.

Given a point cloud consisting of n points, the computational complexity of the PFH algorithm is $O(nk^2)$, where k represents the number of neighborhoods calculated for each point in the point cloud. Since using the PFH algorithm directly for point cloud registration does not guarantee real-time performance, and to reduce computational complexity, the FPFH algorithm is proposed. This approach reduces the computational complexity to $O(n\cdot k)$, while maintaining the recognition capability of the PFH algorithm and expanding the range of neighborhood features. The influence effect of the FPFH domain features of the feature points is shown in Figure 2.

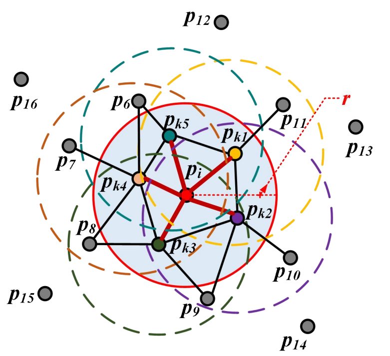


Figure 2. Influence regions of feature points and their neighboring points

The specific calculation process is as follows:

(1) Calculate the feature elements between a feature point and its neighboring points to construct a feature histogram;

(2) Determine the k-neighborhood of the feature point and use the SPFH value to calculate the FPFH feature. The calculation formula is as follows:

$$f_{FPFH}(P_q) = f_{SPFH}(P_q) + \frac{1}{k}\sum_{i=1}^{k}\frac{1}{\omega_k} \bullet f_{SPFH}(P_k) \tag{19}$$

Here, f_{FPFH} represents the fast point feature descriptor, represents the weight, and ω_k represents the distance between the feature point and its neighboring points.

3.2 Coarse Alignment Using the K-4PCS Algorithm

To register a source point cloud with a target point cloud, the initial step in the rough registration process involves performing keypoint detection on the raw data. The RANSAC algorithm can be used to randomly select three non-coplanar, distinct points as a coplanar set of four points in the source point cloud. By leveraging affine invariance, matching point sets are then sought in the target point cloud to form corresponding sets of four points; The extracted keypoints serve as input for the modified 4PCS correspondence, and the 4PCS algorithm generates a set of corresponding keypoints. Subsequently, coarse registration is performed based on the corresponding relationships.The detailed coarse registration algorithm process is shown in Figure 3.

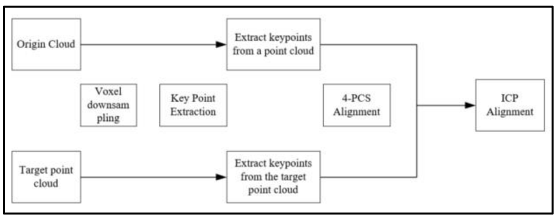


Figure 3 K-4PCS Algorithm Flowchart

3.3 Improved ICP Registration Algorithm

To further improve registration accuracy, we employ an improved ICP algorithm for precise registration and integrate K-dimensional tree search for optimization, thereby accelerating the identification of corresponding point pairs and effectively enhancing computational efficiency. The detailed algorithm is presented in the pseudocode.

The code demonstrates the overall registration process for the source point cloud and the target point cloud. It solves for the rotation matrix r and the translation matrix t without altering the shape, so that the source and target point clouds align. First, the raw data is preprocessed to reduce the computational load on the algorithm and enhance the stability of geometric constraints. Then, using the four-point primitives method, a coplanar region is identified within the target area to find structures that satisfy the geometric relationships. The optimal transformation is selected as T0 based on the overlap ratio. Upon entering the fine-alignment stage, nearest-neighbor correspondences are established, residual corresponding points are removed. Using SVD, incremental rotations and translations are estimated, the global optimized pose is updated, and the mean error is calculated. The process terminates when the change in error falls below a threshold, yielding the final transformation T*.

The pseudocode for the algorithm is shown below:

Require: Source point cloud P, target point cloud Q; number of K4PCS samples S, distance threshold τ_d, and angle threshold τ_θ; maximum number of ICP iterations K, convergence threshold ε, and outlier removal rate ρ

Ensure: Final Rigid Body Transformation $T^* = (R^*, t^*)$

Preprocessing： downsample P and Q and estimate the normal vectors;

$T_0 \leftarrow I$, $score_{best} \leftarrow -\infty$

for s=1 to S do

Randomly sample 4 nearly coplanar points in P to construct an ordered primitive B_P

Based on the side-to-side ratio and the angle invariant, search for candidate sets of 4-point primitives in Q BQ

for Each candidate primitive $B_P \in \beta_Q$ do

Estimate candidate transformations based on B_P ↔ B_Q $T_C = (R_c, t_c)$

Convert P to $P_c = R_c P + t_c$

Calculate the overlap percentage:

$$score = \frac{1}{|P_c|}\sum_{p\in P_c} 1\{\min_{q\in Q} \| p-q \|_2 < \tau_d\}$$

if score > scorebest then

$score_{best} \leftarrow score$, $T_0 \leftarrow T_c$

end if

end for

end for

Perform ICP fine-tuning with T0 as the initial value:

$(R, t) \leftarrow T_0$, $e_{prcv} \leftarrow +\infty$

$\tilde{P} \leftarrow RP + t$

for k=1 to K do

for $\tilde{p}_i \in \tilde{P}$ do

Search for the nearest neighbor in Q $q_{\pi(i)}$

$d_i \leftarrow \|\tilde{p}_i - q_{\pi(i)}\|2$

end for

Retain the set of interior points corresponding to the (1-ρ)N smallest di values C

Solving Incremental Transformations Using SVD in C (ΔR, Δt)

$\tilde{p}_i \leftarrow \Delta R \tilde{P}_i + \Delta t$

$(R, t) \leftarrow (\Delta RR, \Delta Rt + \Delta t)$

$$e \leftarrow \frac{1}{|C|}\sum_{(\tilde{p}_i, q_{\pi(i)})\in C} \| \tilde{p}_i - q_{\pi(i)} \|_2$$

if $|e_{prev} - e| < \varepsilon$ then

break

end if

$e_{prev} \leftarrow e$

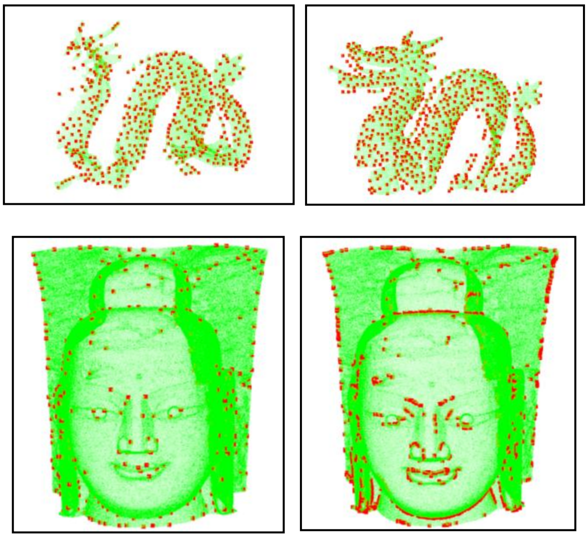
```
end for
T* ← (R, t)
return T*
```

4 Experimental Results and Analysis

To validate the effectiveness of the point cloud registration method described above, this experiment was conducted on an i5-4005U CPU@1.70GHz system running Windows 10 with 16 GB of RAM. The programming experiments were carried out using a C++ program integrated with PCL. The experimental data were obtained from the Dragon dataset in Stanford University's open 3D point cloud database, which includes views from various angles; the original point cloud contains 41,841 points. The other dataset consists of 3D-scanned data of a Buddha head statue from Cave 18 of the Yungang Grottoes; the original point cloud contains 260,431 points. These two distinct datasets were used for validation.

4.1 Feature extraction results

Due to the high density of the point cloud data used in this study, the original data was downsampled to voxels while preserving the features of the point cloud. At the same time, different neighborhood radii were used to estimate the normal vectors of the point cloud. By calculating the angles between the point clouds and normal vectors of the Dragon and Buddha Head statues, a normal vector angle threshold of 3 was set; points where the mean normal vector angle exceeded this threshold were identified as feature points, otherwise, they are classified as non-feature points. A comparison of the feature point extraction results between Reference 2(Song et al., 2021) and this study is shown in the figure below.



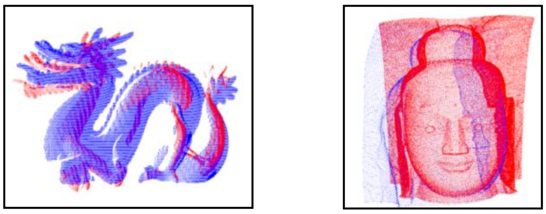
(a) The method in Reference [2] (b) Methodology of this paper

Figure 4 Illustration of feature point extraction results

As shown in Figure 4(a), the feature points extracted in Reference 2(Song et al., 2021) are too few and relatively scattered, which is not conducive to subsequent point cloud registration. However, in this study, downsampling effectively reduces the density of the point cloud, resulting in feature points that are more uniformly clustered while retaining the surface characteristics of the original point cloud.

4.2 Analysis of Point Cloud Registration Results

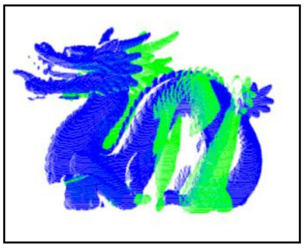
During the coarse registration process following feature point extraction, local point clouds within a neighborhood radius of 0.05 are used to describe features. Coarse registration is performed using the K-4PCS algorithm to derive the transformation matrix for global coordinates. The registration results are shown in Figure 5. As can be seen from the figure, the coarse registration yields good results; however, there are still some deviations at the edges and corners, which require further refinement.



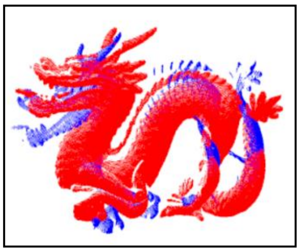
(a)Dragon Preliminary alignment results
(b) Preliminary alignment results for the head sculpture of the standing Buddha in Cave 18 at Yungang

Figure 5 Results of rough point cloud registration

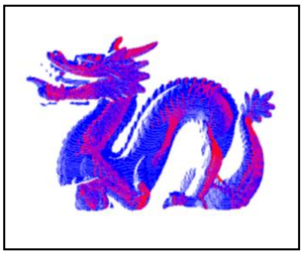
After coarse registration, fine registration is performed using the classical ICP registration algorithm, with the introduction of the KD value to accelerate the search for corresponding points. To verify the effectiveness of the method, the registration results of this algorithm were compared with those reported in Reference 2(Song et al., 2021), as shown in Figure 6 below.



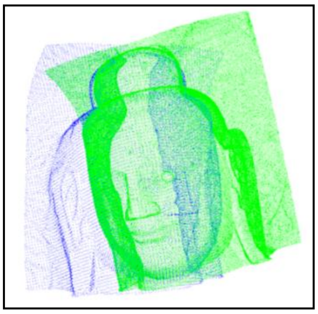
(a) Initial position of the point cloud



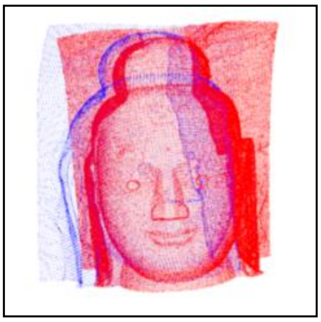
(b) Methods in Reference 2(Song et al., 2021)



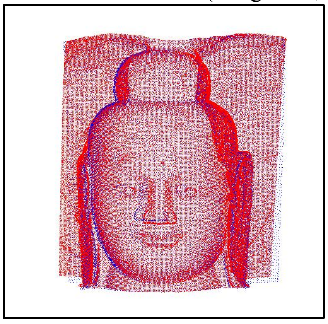
(c) Improvement Methods



(d) Original Location of the Buddha Statue



(e) Methods in Reference 2(Song et al., 2021)



(f) Improvement Methods

Figure 6 Point cloud registration results

Table 1: Registration results of the SAC-IA algorithm

Point cloud data	SAC-IA		ICP		Total time	RMSE
	Accuracy	Time	Accuracy	Time		
Dragon	1.87×10^{-4}	2.028	9.28×10^{-5}	32.872	34.898	1.73×10^{-3}
Buddha's head	1.3×10^{-3}	33.408	3.37×10^{-4}	58.663	91.961	0.78

Table 2: Comparison of the registration algorithm used in this paper with the traditional ICP registration algorithm

Point cloud data	Traditional ICP registration algorithms		The Algorithm in This Paper	
	Time /s	Accuracy /m	Time /s	Accuracy /m
Dragon	35.803	5.42×10^{-3}	18.345	3.53×10^{-4}
Buddha's head	14.582	5.44×10^{-2}	44.18	7.9×10^{-3}

Table 3: Comparison of the proposed algorithm with the method in Reference 2(Song et al., 2021)

Point cloud data	Methods in Reference 2		The Algorithm in This Paper	
	Time /s	Accuracy /m	Time /s	Accuracy /m
Dragon	25.804	1.455×10^{-7}	18.345	3.53×10^{-4}
Buddha's head	188.23	6.4×10^{-2}	44.18	7.9×10^{-3}

To verify the algorithm’s effectiveness in terms of convergence accuracy and registration time, we compared it with the SAC-IA algorithm, as well as traditional ICP and the method described in Reference 2(Song et al., 2021). The comparison results are shown in Tables 1, 2, and 3. As can be seen from the tables, compared with the traditional ICP algorithm, the proposed algorithm improves registration efficiency by 49.8%; compared to the method in Reference 2(Song et al., 2021), efficiency increased by 28.9%, improving convergence efficiency while maintaining registration accuracy. Experiments have demonstrated that, compared to the traditional ICP algorithm and the registration method in Reference 2(Song et al., 2021), the proposed algorithm achieves significant improvements in convergence speed and registration accuracy, thereby validating its reliability and effectiveness.

5 Conclusion

To address the issues with traditional ICP algorithms—namely, their high requirements for the initial point cloud’s position and their relatively low efficiency and speed during point cloud registration—this paper proposes a K-4PCS algorithm based on feature point matching for coarse registration. This approach enables the acquisition of a well-aligned initial pose, effectively resolving the problem of traditional ICP algorithms tending to get stuck in local optima during the registration process; During the registration process, feature points are further extracted by combining the normal vector features of the point clouds with FPFH descriptions of the feature points, thereby ensuring the correspondence between the point clouds. After coarse registration, the ICP algorithm is used to establish new point cloud correspondences, and a K-dimensional tree is employed to accelerate the search for corresponding points, further improving registration efficiency. Experimental results demonstrate that the method proposed in this paper can further enhance the accuracy and efficiency of the registration results.

Acknowledgements

We acknowledge grants from the National Natural Science Foundation of China (No.42171444).

References

Cao, H, Zhu, Q, 2026. A point cloud registration network integrating attention mechanism and selective state space model. Computer Science, 1-18. Accessed 23 March 2026.

Huang, J, Li, D, Yan, W, 2025. An NDT registration method based on adaptive clustering features. Journal of Navigation and Positioning 13(06), 41-47. doi.org/10.16547/j.cnki.10-1096.20250605.

Huang, S, Zhou, Z, Ren, P, 2026. A point cloud registration algorithm based on octree and 3D-SIFT feature extraction. Journal of Applied Optics 47(01), 147-155.

Ji, Z, Ai, H, Han, Y, 2025. Point cloud registration based on normal vectors and curvature key features. Optics and Precision Engineering 33(13), 2136-2152.

Li, J, Wang, J., Guo, S, 2024. Point cloud matching based on adaptive local neighborhood conditions. Optics and Precision Engineering 32(10), 1606-1621.

Li, W., Huang, M, Chen, M., 2026. Registration of aerial images and airborne LiDAR point clouds under building geometric constraints. Geomatics and Information Science of Wuhan University, 1-18. Accessed 23 March 2026.

Li, X, Li, G, Wang, L., 2025. A multi-stage optimized CUBE-ICP algorithm for point cloud registration. Bulletin of Surveying and Mapping (12), 77-81+120. doi.org/10.13474/j.cnki.11-2246.2025.1213.

Qian, J, Xu, H, Zeng, K, 2026. A multimodal 3D point cloud registration method based on improved RT-DETR and FPFH features. Journal of Optoelectronics·Laser, 1-13. Accessed 23 March 2026.

Song, C, Li, J, Liu, G, 2021. Point cloud registration method using feature point sample consensus to improve the ICP algorithm. Beijing Surveying and Mapping 35(03), 317-322. doi.org/10.19580/j.cnki.1007-3000.2021.03.007.

Tu, H, Wu, J, Wang, Y., 2025. Point cloud registration based on multi-feature keypoint extraction and similar-triangle-optimized point pairs. National Remote Sensing Bulletin 29(12), 3706-3722.

Wan, H., Wu, X, 2025. A review of registration methods for three-dimensional point clouds. Journal of Image and Graphics 30(10), 3255-3274.

Wang, W, Tian, M, Yu, J, 2022. An improved iterative closest point method for point cloud registration. Laser & Optoelectronics Progress 59(02), 390-399.

Wang, Y., Zhang, W., Xie, L, 2026. A point cloud registration algorithm using multi-scale feature fusion. Journal of Northeastern University (Natural Science), 1-10. Accessed 23 March 2026.

Xu, M, Zhong, R, Zhong, X, 2026. Registration method for vehicle-borne and backpack heterogeneous point clouds based on image feature matching. Bulletin of Surveying and Mapping (01), 32-38. doi.org/10.13474/j.cnki.11-2246.2026.0106.

Yang, J., Du, C, Fang, D, 2026. An improved ICP-based registration method for point clouds and CAD models. Machinery Design & Manufacture, 1-5. Accessed 23 March 2026.

Yi, J, Xiong, W, Peng, X, 2025. An unsupervised point cloud registration algorithm integrating feature interaction and enhanced point matching. Optics and Precision Engineering 33(20), 3281-3298.

Yuan, C., Wu, L, Zhang, Q, 2026. Research on surface blade point cloud registration based on improved GICP. Modular Machine Tool & Automatic Manufacturing Technique, 1-8. Accessed 23 March 2026.

Zhang, P, Zhao, H., 2025. A point cloud registration method based on an improved 3DSIFT algorithm. Laser & Infrared 55(02), 296-303.

Zhao, F, Li, Z, Yu, J, 2026. A three-dimensional point cloud registration algorithm based on color texture and geometric features. Science Technology and Engineering 26(05), 2051-2059.

Zhao, F, Zhang, Y, Yu, J, 2026. A point cloud registration algorithm based on dimension-reduced multi-scale features and iterative closest point. Laser & Optoelectronics Progress, 1-19. Accessed 23 March 2026.

Zhou, Y, Li, M, Tan, D, 2026. A point cloud registration method based on quaternion spectral features. Laser & Optoelectronics Progress, 1-17. Accessed 23 March 2026.