

# *The 3D Point Cloud Registration Algorithm Based on Harris-DLFS*

Zongwei Huang<sup>1</sup>, Juan Zhu<sup>1,\*</sup>, Chang Xiao<sup>1</sup>, Zeyuan Liu<sup>1</sup>, Zijian Cong<sup>2</sup>

<sup>1</sup>*School of Mechanical and Electrical Engineering, Changchun University of Technology, Changchun, China*

<sup>2</sup>*School of Electrical Information, Changchun Guanghua University, Changchun, China*

*\*Corresponding author*

**Keywords:** 3D point cloud; keypoints; feature description; feature matching; pairwise alignment

**Abstract:** Three-dimensional model reconstruction is a pivotal technology in the realm of computer vision. Point cloud registration serves as its integral step, which decisively impacts the efficiency and precision of the entire reconstruction process. However, existing point cloud registration algorithms often face issues. These include prolonged processing time, inadequate accuracy, and poor robustness. To address these problems, this paper proposes a novel point cloud registration algorithm based on corner detection (Harris) and partition-based local feature statistics (DLFS). The main steps are as follows: Firstly, the Harris corner detection algorithm is employed. This step is crucial for extracting key points and enhancing the efficiency of the registration process. Secondly, the DLFS method is used to describe the features of each key point, generating feature vectors. Subsequently, matching point pairs are filtered based on rigid distance constraints, and a coarse registration is performed using the Random Sample Consensus (RANSAC) algorithm. Finally, the Iterative Closest Point (ICP) algorithm is applied for fine registration. Experimental results demonstrated the effectiveness of this method. It significantly improved registration accuracy, robustness, and computational efficiency. Therefore, it holds substantial value for practical point cloud registration applications.

## 1. Introduction

3D point cloud registration is widely applied in various fields such as autonomous driving [1], industrial production [2], agriculture [3], and cultural heritage restoration [4]. In practical applications, due to factors such as object occlusion, surface layout, and measurement methods, it is often necessary to perform multiple measurements from different perspectives and integrate the point cloud data into a common coordinate system. This is achieved by calculating rotation and translation matrices to obtain a complete 3D point cloud dataset.

Point cloud registration can be primarily categorized into methods based on Iterative Closest Point (ICP) [5, 6], feature learning methods [7, 8], and end-to-end learning methods [9]. Compared to the latter two, ICP-based methods have a strict mathematical theory guaranteeing their convergence, and they do not require prior training for unknown datasets, thus possessing good

generalization capabilities [10]. Moreover, ICP methods have good robustness when handling certain degrees of initial errors and noise. However, these methods still face challenges such as sensitivity to initial alignment, slow convergence speed, and difficulty in finding global optimal solutions. To address these challenges, various optimization methods have been proposed. For instance, a point cloud automatic registration method based on normal vectors is proposed [11], which uses coarse registration to provide a good initial pose for fine registration, thereby avoiding local optima. However, due to the poor noise robustness of feature points based on normal vectors, this may lead to a decrease in registration accuracy. To overcome the impact of noise on registration, reference [12] adopted shape index key point detection and nearest neighbor distance ratio to enhance noise resistance. Reference [13] employed a point cloud registration method driven by multi-feature extraction and matching matrix, which reduces the impact of noise on registration by extracting robust keypoints and local geometric features. Reference [14] adopts a multi-normal neighborhood for feature point extraction, uses the Random Sample Consensus (RANSAC) algorithm and clustering selection method to optimize the descriptors for coarse registration, and finally applies the ICP algorithm for fine registration. This algorithm improves registration accuracy but suffers from reduced efficiency due to the computation-intensive multi-neighborhood normal calculation. To improve the success rate of point cloud registration under low overlap, reference [15] employs the 4-Point Congruent Sets (4PCS) algorithm for coarse registration and Sparse Iterative Closest Point (SICP) algorithm for fine registration. However, the 4PCS algorithm has high time complexity. To reduce the time consumption of the 4PCS algorithm, literature [16] used the Principal Component Analysis (PCA) algorithm to extract feature points, Super-4PCS to complete coarse registration, and then further used ICP for fine registration, which reduced the computational complexity during matching and improved registration efficiency. However, it still required manual setting of the overlap rate.

To address these issues, a three-dimensional point cloud registration algorithm based on Harris-DLFS was proposed. This algorithm aims to tackle the low registration accuracy and efficiency problems of traditional point cloud algorithms when dealing with noisy and occluded point clouds. The method was validated on the Stanford point cloud dataset and the U3OR dataset.

## 2. Harris-DLFS Algorithm Principle

In practical applications, large volumes of three-dimensional point cloud data are often encountered, and local neighborhood points may have similar features, leading to a certain degree of information redundancy. Concurrently, a considerable amount of point cloud data inevitably contains noise information. These factors can affect the efficiency and robustness of algorithms. To solve these problems, a voxel grid filter is employed to process the three-dimensional point cloud. Specifically, the three-dimensional point cloud is divided into multiple small cubes, or "voxels." Then, points from the original point cloud data that are closest to each voxel centroid are used to replace the voxel center, thereby simplifying the entire three-dimensional point cloud.

### 2.1. Harris Feature Point Extraction

To quickly extract a representative and stable set of feature points from the down-sampled point cloud, the Harris algorithm [17] is used for feature point extraction. The specific steps of the detection algorithm are as follows:

(1) For any point  $p_i$  in the point cloud, a spherical neighborhood is searched within a search radius  $R_k$ , and the neighboring points are denoted as a point set  $Q = \{q_1, q_2 \dots q_k\}$ . The covariance matrix of normals is constructed for point  $p$  and the neighborhood point set  $Q$ . The expression for the covariance matrix of normals is equation (1).

$$M_{3d} = \frac{1}{k} \sum_{x,y,z} \begin{bmatrix} n_x^2 & n_x n_y & n_x n_z \\ n_y n_x & n_y^2 & n_y n_z \\ n_z n_x & n_z n_y & n_z^2 \end{bmatrix}, \quad (1)$$

In the equation,  $k$  represents the number of neighboring points for point  $p_i$ , and  $n_x, n_y, n_z$  denote the coordinates of the normal vectors of the neighboring points.

(2) The response value for each point in the point cloud is calculated using the following equation (2).

$$R_{3d} = \det M_{3d} - 0.04(\text{trace} M_{3d})^2, \quad (2)$$

(3) The feature points in the point cloud are determined based on the comparison between the  $R_{3d}$  value of each point, calculated using a certain formula, and a threshold value  $R_t$ . If the  $R_{3d}$  value exceeds the threshold and the point is a local maximum, then the point  $p_i$  is considered a feature point and is included in the set of feature points  $p_{key}$ . Otherwise, the current point is not considered a feature point.

## 2.2. DLFS Feature Point Description

For the obtained set of Harris feature points  $p_{key}$ , the Distributed Local Feature Statistics (DLFS) descriptor [18] is used to describe the neighborhood information of the feature points. DLFS first constructs the Local Reference Axis (LRA) and Local Minimal Axis (LMA) using an efficient method. Then, based on the spatial and geometric information encoded by the neighborhood's LRA and LMA, DLFS exhibits robustness against noise and occlusion. The computation process of the DLFS descriptor is as follows:

(1) For any point  $p_{key-i}$  in the feature point set  $p_{key}$ , based on the search radius  $R_d$ , the neighboring point set  $Q' = \{q'_1, q'_2 \dots q'_k\}$  of  $p_{key-i}$  can be found. In order to achieve rotation and translation invariance of local neighborhood points,  $Q'$  is first transformed, as shown in Figure 1, so that  $p_{key-i}$  coincides with the coordinate origin, LRA aligns with the z-axis of the global coordinate system, and the transformed  $Q'$  is denoted as  $Q'_t$ .



Figure 1: Transformation Method of Point Set

(2) Alignment Based Local Neighborhood Point Set  $Q'_t$ , first divide the spatial area into  $N$  regions along the direction of the projection radius and calculate the local height information ( $lh$ ) of each neighboring point in each partition. The calculation formula is equation(3).

$$lh_k = R_d + q'_{tk}(z), \quad (3)$$

Among them, the range of  $lh_k$  is  $[0, 2R_d]$  and  $q'_{tk}(z)$  is the z-axis coordinate of the neighborhood point  $q'_{tk}$ .

(3) Calculate the three geometric properties ( $\alpha$ ,  $\beta$ ,  $\gamma$ ) of the local neighborhood point set  $Q'_t$  for point pkey-i, specifically the calculation of three angle attributes. The calculation formula is equation(4).

$$\begin{cases} \alpha_k = \arccos\left(\frac{(LRA \times pq'_{tk}) \cdot LMA_k}{\|LRA \times pq'_{tk}\| \|LMA_k\|}\right) \\ \beta_k = \arccos\left(\frac{LRA \cdot LMA'_k}{\|LMA'_k\|}\right) \\ \gamma_k = \arccos\left(\frac{pq'_{tk} \cdot LMA'_k}{\|pq'_{tk}\| \|LMA'_k\|}\right) \end{cases} \quad (4)$$

Where  $pq_k$  represents the vector from p to  $q_k$ ,  $LMA_k$  represents the LMA at  $q'_{tk}$ , and  $LMA'_k$  is calculated using the equation(5).

$$LMA'_k = LMA_k - \frac{(LRA \times pq'_{tk}) \cdot LMA_k}{\|LRA \times pq'_{tk}\|} (LRA \times pq'_{tk}). \quad (5)$$

The range of these three angle attributes is  $[0, \pi]$ .

(4) After calculating the four attributes ( $lh$ ,  $\alpha_k$ ,  $\beta_k$ ,  $\gamma_k$ ) in the subspace for point pkey-i with all points in  $Q'_t$ , the histograms ( $H_{lh}$ ,  $H_\alpha$ ,  $H_\beta$ ,  $H_\gamma$ ) of the four attributes are computed. The histograms are then normalized, and different weights ( $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ ) are assigned to each histogram. Finally, the four histograms are concatenated to form the DLFS descriptor  $f = \{\lambda_1 H_{lh}, \lambda_2 H_\alpha, \lambda_3 H_\beta, \lambda_4 H_\gamma\}$ .

### 3. Coarse and Fine Registration

Firstly, the feature point sets of the source and target point clouds are described using DLFS descriptors, resulting in feature vectors  $P_f$  and  $Q_f$ . Since the feature vectors for the same feature point are identical, based on this principle, the corresponding points in  $Q_f$  for each feature vector in  $P_f$  can be found by calculating the nearest Euclidean distance, thereby obtaining all initial corresponding point pairs.

Next, the RANSAC algorithm is used to compute the optimal transformation based on these initial corresponding point pairs. The specific steps of RANSAC are as follows:

(1) Randomly select 3 point pairs from the initial corresponding point pairs as samples.

(2) Calculate the rigid transformation matrix  $T$  based on the randomly selected 3 point pairs.

(3) Apply the transformation matrix  $T$  to the source point cloud and then evaluate the Euclidean distances between the remaining point pairs. If the Euclidean distance between the point pairs is less than a threshold value  $t$ , it is considered a correct match and classified as an inlier; otherwise, it is considered an incorrect match and classified as an outlier.

(4) Repeat the above steps until the predetermined number of iterations is reached. Finally, select the rigid transformation matrix with the highest number of inliers as the final transformation matrix  $T'$ .

After obtaining the final transformation matrix  $T'$ , it is applied to the source point cloud  $P$  to obtain  $P'$ , thus completing the coarse registration.

Next, the Iterative Closest Point (ICP) algorithm is used to refine the coarse registration result. The basic steps of ICP are as follows: for each point in the source point cloud, search for the nearest point in the target point cloud as its corresponding match point. Then, based on these match point pairs, compute the transformation matrix between the source and target point clouds, which updates the coordinates of the source point cloud. Repeat this process iteratively, gradually approaching the spatial position of the target point cloud, until the iteration termination condition is met. In this way, the optimal alignment between the source and target point clouds is achieved.

## 4. Experimental Results

In order to validate the effectiveness of the proposed algorithm, comparative experiments were conducted with the K4PCS and ISS-FPFH algorithms in this study. The robustness of the algorithm was tested on the B3R and U3OR datasets, evaluating its performance on model point clouds, partial point clouds, and occluded point clouds. All experiments were conducted on an Ubuntu system with a 3.5GHz CPU (i5-13600Fk) and 32GB of memory, utilizing the pcl1.10 points cloud library. The evaluation criteria for the experiments included root mean square error (RMSE) and time. The formula for RMSE is the equation(6).

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N \|p_i - q_i\|^2}, \quad (6)$$

Where  $p_i$  and  $q_i$  represent the corresponding nearest neighbor points in the two registered point clouds, and  $N$  denotes the total number of corresponding points.

From Figure 2, it can be observed that all three methods successfully completed point cloud registration, but the ISS-FPFH method exhibited significant errors near the front paws of the rabbit, while the K4PCS method had noticeable errors around the rabbit's ears. In comparison, the coarse registration method proposed achieved a more uniformly accurate fusion compared to the ISS-FPFH and K4PCS methods. Combining the analysis from Table 1, it can be concluded that in the validation of common point clouds under the B3R dataset, the proposed method not only achieves good coarse registration results but also has shorter processing time. Compared to the ISS-FPFH and K4PCS methods, it demonstrates higher registration accuracy and efficiency.

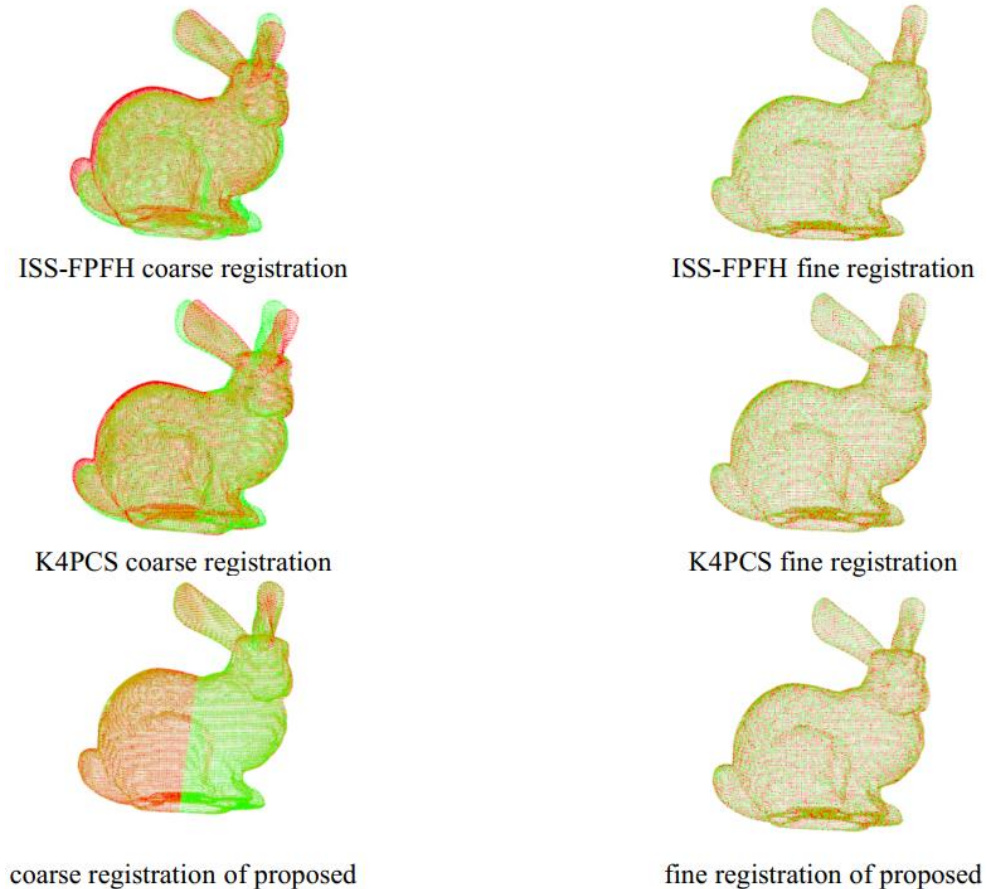
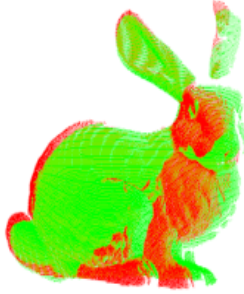


Figure 2: Model Point Cloud Registration Results

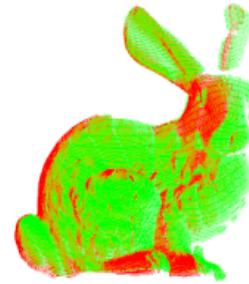
Table 1: RMSE and time consumption of model point cloud registration results

Registration method	RMSE	coarse time/s	fine time/s
ISS-FPFH	0.0021	4.07	1.136
K4PCS	<b>0.00207</b>	15.0	1.02
proposed	<b>0.00207</b>	<b>2.67</b>	<b>0.22</b>

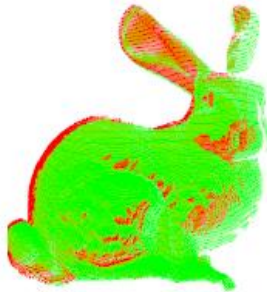
Based on the analysis results from Figure 3 and Table 2, it can be observed that the ISS-FPFH method exhibits rotational errors during coarse registration but has the shortest processing time. The K4PCS method shows translational errors and has the longest processing time. In contrast, the proposed method achieves substantial overlap during coarse registration, with a processing time comparable to that of the ISS-FPFH method. In the fine registration stage, the proposed method outperforms both the ISS-FPFH and K4PCS methods, exhibiting improved accuracy. Furthermore, it has the shortest processing time for fine registration, while the overall processing time remains like that of the ISS-FPFH method.



ISS-FPFH coarse registration



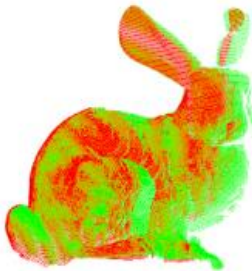
ISS-FPFH fine registration



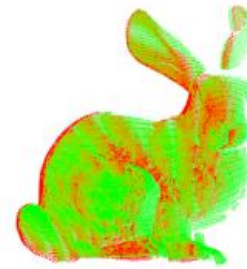
K4PCS coarse registration



K4PCS fine registration



coarse registration of proposed



fine registration of proposed

Figure 3: Partial Point Cloud Registration Results

Table 2: RMSE and time consumption of point cloud registration results for partial point cloud

Registration method	RMSE	coarse time/s	fine time/s
ISS-FPFH	0.513	<b>4.78</b>	1.42
K4PCS	0.515	23.2	0.69
article	<b>0.38</b>	6.2	<b>0.25</b>

In order to further validate the robustness of the proposed algorithm against occlusions, comparative experiments were conducted in the U3OR dataset. As shown in Figure 4 and Table 3, both the K4PCS and ISS-FPFH methods failed to achieve successful registration. In contrast, the method proposed in this study achieved accurate alignment during the coarse registration stage and achieved a perfect overlap with the source and target point clouds during the fine registration stage.

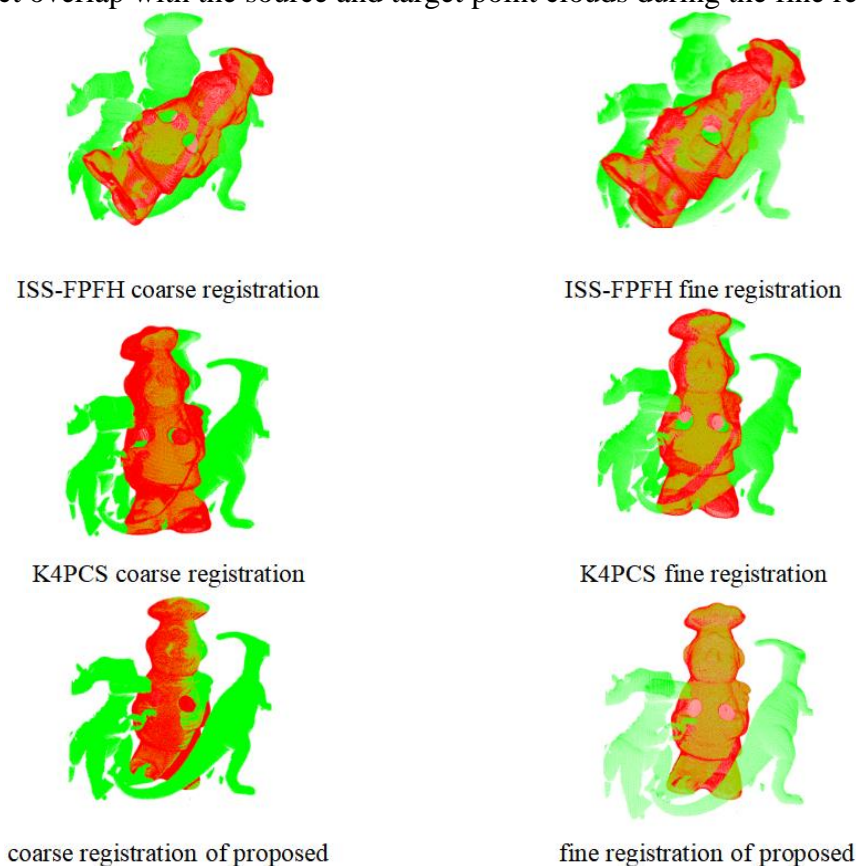


Figure 4: Occlusion Point Cloud Registration Results

Table 3: RMSE and time consumption of occluded point clouds in U3OR dataset

Registration method	RMSE	coarse time/s	fine time/s
ISS-FPFH	/	/	/
K4PCS	/	/	/
article	<b>751.003</b>	<b>10</b>	<b>79.39</b>

## 5. Conclusions

This research proposes a 3D point cloud registration algorithm based on Harris-DLFS. This algorithm exhibits robustness and can automatically register model point clouds, partial perspective point clouds, and scene-occluded point clouds. The algorithm first down-samples the original point cloud through voxel filtering, then extracts feature points through the Harris algorithm, and uses the

DLFS descriptor to describe the neighborhood information of the feature points, establishing a feature histogram. Subsequently, the initial registration point pairs are obtained through histogram distance constraints. Then, the optimal transformation matrix is obtained using the RANSAC algorithm. Finally, the ICP algorithm is used to complete the fine registration. Experimental results indicate that compared with ISS-FPFH and K4PCS, the method proposed in this research has higher registration accuracy, shorter time consumption, and stronger robustness, demonstrating practical value for achieving automatic point cloud registration.

## Acknowledgments

This work was supported by department of science and technology of Jilin Province (20220203091SF).

## References

- [1] Gu L. P., Sun S. Y., Liu X. H., & Li X. (2021). Research on 3D multi-target tracking algorithm based on laser point cloud coordinate system. *Laser and Infrared*, 51(10), 1307-1313.
- [2] Xue S., Lü N. F., Shen Y. Y., Liu Z. Y., & Guo J. B. (2019). Recognition method for mechanical parts based on laser 3D point cloud. *Infrared and Laser Engineering*, 48(4), 169-176.
- [3] Ma X. L., Xue H. R., & Zhou Y. Q. (2023). Point cloud registration method for sheep body based on feature matching. *Journal of China Agricultural University*, 28(4), 129-138.
- [4] Yang P. C., Yang C., Meng J., Xiao Y., & Cui J. B. (2023). Point cloud boundary registration method for cultural relics based on normal vectors and planar index features. *Chinese Optics*, 16(3), 654-662.
- [5] Zhu J. H., Zhu L., Li Z. Y., Li C., & Cui J. R. (2016). Automatic multi-view registration of unordered range scans without feature extraction. *Neurocomputing*, 171, 1444-1453.
- [6] Yang J. Q., Xiao Y., & Cao Z. G. (2019). Aligning 2. 5D Scene Fragments With Distinctive Local Geometric Features and Voting-Based Correspondences. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(3), 714-729.
- [7] Gojcic Z., Zhou C., Wegner J. D., & Wieser A. (2019). The Perfect Match: 3D Point Cloud Matching With Smoothed Densities. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5540-5549.
- [8] Zeng A., Song S., Nießner M., Fisher M., Xiao J., & Funkhouser T. (2017). 3DMatch: Learning Local Geometric Descriptors from RGB-D Reconstructions. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 199-208.
- [9] Yang Z. P., Pan J. Z., Luo L. J., Zhou X. W., & Huang Q. X. (2018). Extreme Relative Pose Estimation for RGB-D Scans via Scene Completion. *2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4531-4540.
- [10] Guo Y. L., Bennamoun M., Sohel F., Lu M., Wan J., & Kwok N. M. (2016). A Comprehensive Performance Evaluation of 3D Local Feature Descriptors. *International Journal of Computer Vision*, 116(1), 66-89.
- [11] Tao H. J., & Da F. P. (2013). Automatic Registration Algorithm for the Point Clouds Based on the Normal Vector. *Chinese Journal of Lasers*, 40(8), 0809001.
- [12] Zhang L., Xiao J., Cheng X. L., & Wang Y. (2023). Three-dimensional point cloud registration based on matching of multiple types of geometric primitives. *Journal of University of Chinese Academy of Sciences*, 40(2), 258-267.
- [13] Liu L., Xiong F. G., Yin Y. H., Guo R., Xue H. X., & Han X. (2023). Point cloud registration driven by multi-feature extraction and matching matrix. *Computer Engineering and Design*, 44(5), 1419-1426.
- [14] Lu J., Peng Z. T., & Xia G. H. (2015). Point cloud multi-normal neighborhood feature registration algorithm. *Optoelectronics Laser*, 26(4), 780-787.
- [15] Fu Y., Chen, P., Guo G. S., & Liu X. Y. (2022). Application of point cloud registration method based on 4PCS and SICP in rail wear calculation. *Journal of Electronic Measurement and Instrumentation*, 36(12), 210-218.
- [16] Zeng W., Yang T., & Yu Y. (2022). Super-4PCS point cloud registration method combining improved FPFH. *Modern Radar*, 1-8.
- [17] Aiger D., Mitra N. J., & Cohen-Or D. (2008). 4-points congruent sets for robust pairwise surface registration. *Acm Transactions on Graphics*, 27(3), 1-10.
- [18] Zhao B., & Xi J. T. (2020). Efficient and accurate 3D modeling based on a novel local feature descriptor. *Information Sciences*, 512, 295-314.