ISSN: 2710-0588

Extraction of Corresponding Point Cloud Features Based on Least Squares Deviation Analysis

Zhenhao Xing[1,*], Xiayu Zhao[1], Hongmei Qu[1]

[1]School of Surveying & Land Information Engineering, Henan Polytechnic University, Jiaozuo 454000, China

[*]Corresponding Author: Zhenhao Xing (Email: xzh123@home.hpu.edu.cn)

Abstract

Accurate extraction of corresponding feature points between adjacent frames is crucial for improving pose estimation accuracy in multi-line LiDAR frame-to-frame registration. This paper proposes a least squares deviation analysis method to optimize the matching process of pole-like object features. First, point cloud data is preprocessed, including denoising and grid projection, to enhance feature point stability. Then, candidate feature points are initially selected using point cloud clustering and bounding box methods. The least squares method is applied to analyze the deviation of features between adjacent frames, eliminating points with significant matching errors. Finally, optimized feature matching improves the accuracy of frame-to-frame registration. Experimental results demonstrate that the proposed method can efficiently and accurately extract corresponding pole-like object features between adjacent frames, enhancing registration stability.

Keywords

Multi-line LiDAR, frame-to-frame registration, least squares method, deviation analysis, feature matching.

1. Introduction

Accurate feature extraction between consecutive frames is crucial for enhancing the registration accuracy of multi-line LiDAR systems. In urban environments, dynamic objects, occlusions, and sensor noise introduce challenges in feature matching, leading to potential localization errors. To address these issues, robust feature extraction and outlier removal strategies are required to ensure reliable frame-to-frame alignment.

In recent years, various LiDAR-based registration methods have been proposed, leveraging point cloud characteristics to improve positioning accuracy. Traditional feature-based methods rely on geometric descriptors or handcrafted rules to identify corresponding points between frames. However, these approaches are sensitive to environmental variations and may result in mismatches, affecting the overall localization performance. To overcome these limitations, optimization-based techniques such as the least squares method have been widely applied for error minimization in feature matching.

This paper proposes an optimization method based on Euclidean clustering[1,2], centroid position fitting, and least squares deviation analysis to accurately extract corresponding point cloud features between adjacent frames. Taking pole-like objects, which are common in urban environments, as an example, the extracted pole-like objects in adjacent frames are first clustered using Euclidean clustering. The centroids of the clustered point clouds are then fitted to represent their positions in 3D space. By aligning the pole-like objects in a unified coordinate system, their deviations are analyzed using the least squares method[3,4] to identify and retain features with corresponding positions in adjacent frames. Experimental results show that the

ISSN: 2710-0588

proposed method effectively extracts corresponding features between adjacent frames, enabling accurate frame-to-frame registration while reducing interference from unmatched point clouds and improving registration accuracy.

2. Related Work

With the advancement of LiDAR technology, particularly multi-line LiDAR, it has been widely applied in autonomous driving, intelligent transportation, and robotic navigation. Feature extraction from LiDAR point cloud data is crucial for LiDAR odometry applications. Accurate extraction of corresponding feature points between adjacent frames is essential for precise frame-to-frame registration.

Feature extraction plays a crucial role in LiDAR data processing, as extracting stable point cloud features effectively is a key challenge in LiDAR odometry. Since multi-line LiDAR captures a large number of points in each frame, using all point clouds for registration is computationally expensive. Leveraging feature information for registration significantly improves computational efficiency. Traditional point cloud feature extraction methods include geometric features, statistical features, and feature descriptors.

Geometric features are derived from the shape and structure of the point cloud, describing the object's geometry. These include point cloud normal vectors, normal estimation, curvature and other geometric properties. Geometric features are intuitive and computationally simple, making them suitable for large-scale point clouds. However, they are sensitive to noise and inaccurate in describing complex shapes. Geometric features are suitable for applications such as simple object shape recognition, localization and terrain analysis. Weinmann et al.[5] examined the impact of geometric feature factors and proposed a deep learning-based classification framework. After downsampling the point cloud, neighborhood recovery was performed, followed by the extraction of a set of geometric features in different neighborhoods. Finally, various classifiers based on deep learning were applied to classify the point cloud and evaluate its correlation.

Statistical features are extracted by performing statistical analysis on local regions of the point cloud. Commonly used statistical features include the density and distance distribution of point clouds. Statistical features can adapt to noise and environmental changes, and are also applicable to sparse point clouds and dynamically transformed point clouds. However, statistical features rely on the overall distribution of local regions, potentially overlook fine geometric structures. Statistical features are suitable for analyzing the overall distribution of point clouds and point cloud clustering. Zhang et al.[6] used statistical model based on Poisson distribution and region-growing algorithm to extract feature points from the point cloud. They adaptively calculate different feature measurement thresholds for various local features and reconstruct feature lines based on the linkage information and the geometric shape of the feature point clusters.

Feature descriptors[7] are vectors or sets of eigenvalues that describe the characteristics of the region surrounding a feature point, and are divided into local and global descriptors. Feature descriptors capture local geometric and topological information of the point cloud, showing strong adaptability to shape changes. They can also introduce multiple features, such as normal direction and curvature through descriptors. However, computing feature descriptors is typically complex, requiring substantial computational resources for large-scale point clouds. They also have weak adaptability to sparse point clouds and noise. Feature descriptors are suitable for applications such as object recognition, matching and tracking. Rusu et al.[8,9] proposed the Point Feature Histograms (PFH) algorithm and the Fast Point Feature Histograms (FPFH) algorithm. The PFH algorithm describes local features by calculating the histogram of local geometric relationships in the point cloud. The FPFH algorithm optimizes PFH by reducing

ISSN: 2710-0588

computational complexity and the time required for calculations. Rusu et al.[10] also proposed the Viewpoint Feature Histogram (VFH) algorithm, which combines viewpoint information and local geometric information as a global descriptor. This algorithm is capable of handling complex object shapes but has a higher computational complexity and requires more computational resources. Charles et al.[11,12] proposed the PointNet model, which uses deep learning for feature extraction from point clouds. They later improved it to develop the PointNet++ model.

This study makes full use of the rod-shaped ground objects on both sides of urban roads. According to the geometric characteristics of these objects, initial feature extraction is performed in the vertical direction by analyzing the changes in depth information of neighboring points. Furthermore, the extracted point cloud clusters are assessed according to the overall geometric feature information of the rod-shaped ground objects.

3. Materials and Methods

After accurately extracting pole-like object point clouds from each frame of multi-line LiDAR data, it is necessary to further match corresponding pole-like objects between adjacent frames to ensure the accuracy of frame-to-frame registration. However, due to platform motion and dynamic environmental changes, there may be deviations in the extracted pole-like objects between adjacent frames. Some pole-like objects may not be successfully extracted in certain frames, or their shapes may change due to variations in perspective, occlusion, and other factors, increasing the difficulty of frame-to-frame matching. To address this issue, this study proposes a least squares deviation analysis method to identify corresponding pole-like object point cloud clusters between adjacent frames. The method calculates the positional deviation of pole-like objects between adjacent frames using the least squares method and selects pole-like object pairs with high matching confidence based on the statistical characteristics of the deviation. Specifically, the method first computes the geometric center coordinates of all candidate pole-like object point clouds in adjacent frames and analyzes their displacement trends using the least squares method. This process eliminates mismatches that may arise due to noise or local environmental changes, thereby ensuring the matching accuracy of pole-like objects between adjacent frames. The application of this method not only enhances the stable matching capability of pole-like objects between frames but also reduces mismatches caused by environmental changes or noise interference, providing more reliable feature point clouds for subsequent pose estimation and frame-to-frame registration. The specific steps are as follows:

(1) Fitting point Center point of cloud cluster. the Euclidean clustering algorithm based on KD-tree is used to obtain the point cloud clusters of rods in adjacent frames, and the three-dimensional coordinates of the center of each rod are fitted to represent the position of the rod. The centers of the fitted rod-shaped ground objects in the current frame and the previous frame are denoted as $M = \{m_1, m_2, m_3, \cdots, m_i\}$ and $N = \{n_1, n_2, n_3, \cdots, n_j\}$, where i and j represent the number of rod-shaped ground object point cloud clusters in the current and previous frames, respectively. The point cloud data from both frames of the multi-line LiDAR are extracted in the same coordinate system with the LiDAR as the origin. The points from M and N are combined and represented as $Q = \{q_1, q_2, \cdots, q_i, q_{i+1}, q_{i+2}, \cdots, q_{i+j}\}$.

(2) The deviation from each central point to the nearest point was calculated. The minimum distance $d_{\min}$ to the nearest neighbor for all points in Q is calculated, as well as the average nearest neighbor distance $\overline{d}$. The deviation of the nearest neighbor distance for each point is denoted as δd .

ISSN: 2710-0588

(3) The corresponding rod-shaped object was determined by least square deviation analysis. The index of each point in Q is used as the independent variable x, and δd is taken as the dependent variable y. Least squares fitting is applied to analyze the data.

The least squares fitting line equation is:

$$y = ax + b \tag{1}$$

$$a = \frac{n\sum x_k y_k - \sum x_k \sum y_k}{n\sum x_k^2 - (\sum x_k)^2} \tag{2}$$

$$b = \overline{y} - a\overline{x} \tag{3}$$

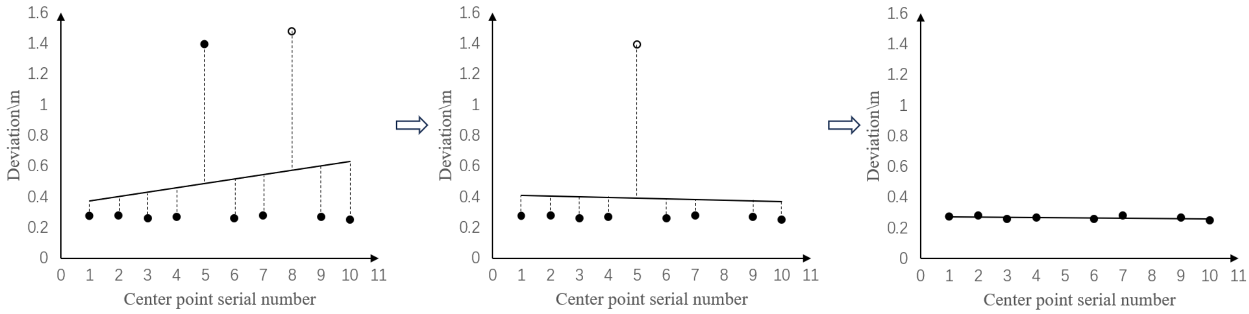

Figure 1. Least squares fitting deviation to remove point clouds with no corresponding position.

As shown in Figure 1, the X-axis represents the index of the center points of rod-shaped ground object point clouds, and the Y-axis represents the deviation of the distance between each center point and its nearest point. After each least squares fitting, the slope is checked to see if it meets the conditions. If the slope is not sufficiently small, the difference between the true value and the fitted value is calculated, as shown by the dashed line in the Figure 1. The point with the largest error (shown as the white point in the Figure 1) is removed, and least squares fitting is performed again until the slope meets the condition, or the error of all points is no greater than the set threshold, and the maximum number of iterations is reached.

Figure 2 illustrates the position of rod-shaped ground objects point clouds in adjacent frames. Since the dependent variable is the deviation of the distance to the nearest neighbor, and the position distance difference between corresponding points in adjacent frames is similar, the fitted line should be as parallel to the X-axis as possible, meaning the slope (a) should be minimized. The real and fitted values' deviations for each point are calculated, and points with large deviations are marked and removed. The least squares fitting process is repeated until the deviations between all points and their fitted values meet the threshold requirement (set to 0.2 in this study). The marked points corresponding to the point cloud clusters in M and N are then deleted, leaving only the points that have corresponding rod-shaped ground objects in both adjacent frames.

ISSN: 2710-0588

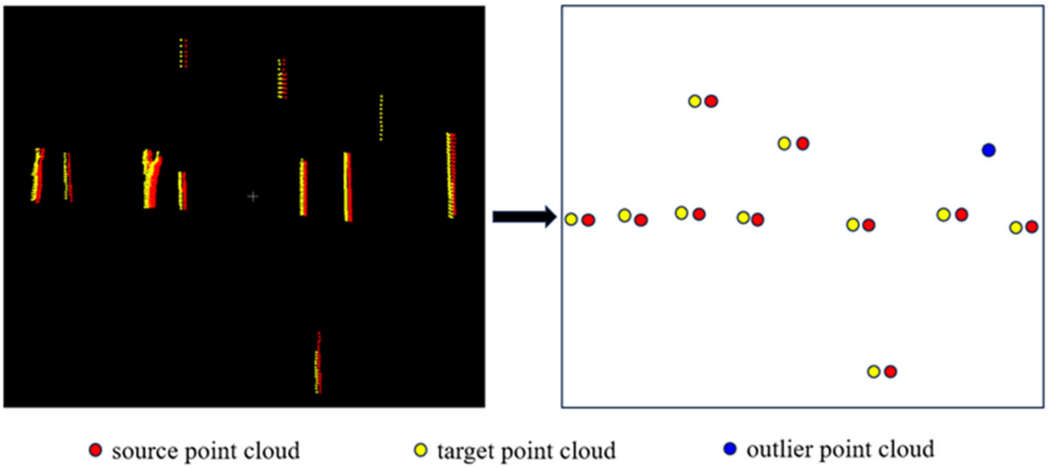

Figure 2. Position diagram of point cloud of adjacent frame rod

4. Experimental Results

The experimental system runs on Windows 11 with Visual Studio 2017 and utilizes the Point Cloud Library (PCL) 1.13.1. The experimental data was collected using a Hesai 32-line multi-line LiDAR (XT32M2X) configured in dual-echo mode, with a point cloud acquisition rate of 1,280,000 points per second, a scanning frame rate of 20 Hz, a horizontal angular resolution of 0.36°, and a vertical angular resolution of 1.3°. The data details are shown in Figure 3.

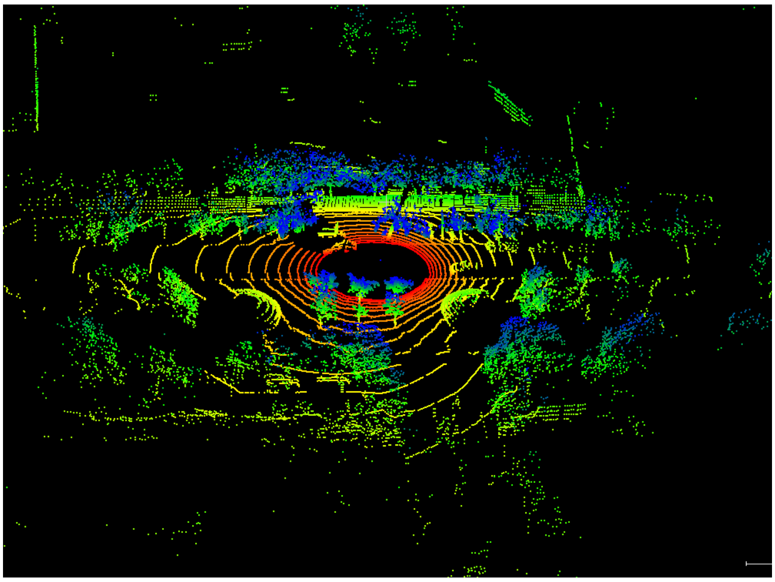

Figure 3. Multi-line LiDAR point cloud data

The first two subfigures in Figure 4 show the results of the processed multi-line LiDAR point cloud data, where the extracted vertical point clouds have been clustered using Euclidean clustering. The corresponding pole-like object point clouds between adjacent frames, retained using the least squares method, are also displayed. In these figures, the red and yellow point clouds represent the pole-like objects in the current and previous frames, respectively. As shown, the pole-like object features retained through the least squares method are clearly defined and have corresponding positions in adjacent frames, facilitating subsequent feature registration for pose transformation estimation. The last two subfigures in Figure 4 illustrate the results of applying Iterative Closest Point (ICP) to align the pole-like objects in adjacent frames within the same coordinate system. In these figures, the yellow point cloud represents

ISSN: 2710-0588

the target, while the red point cloud represents the source. It can be observed that after registration, the yellow and red point clouds are nearly perfectly aligned visually.

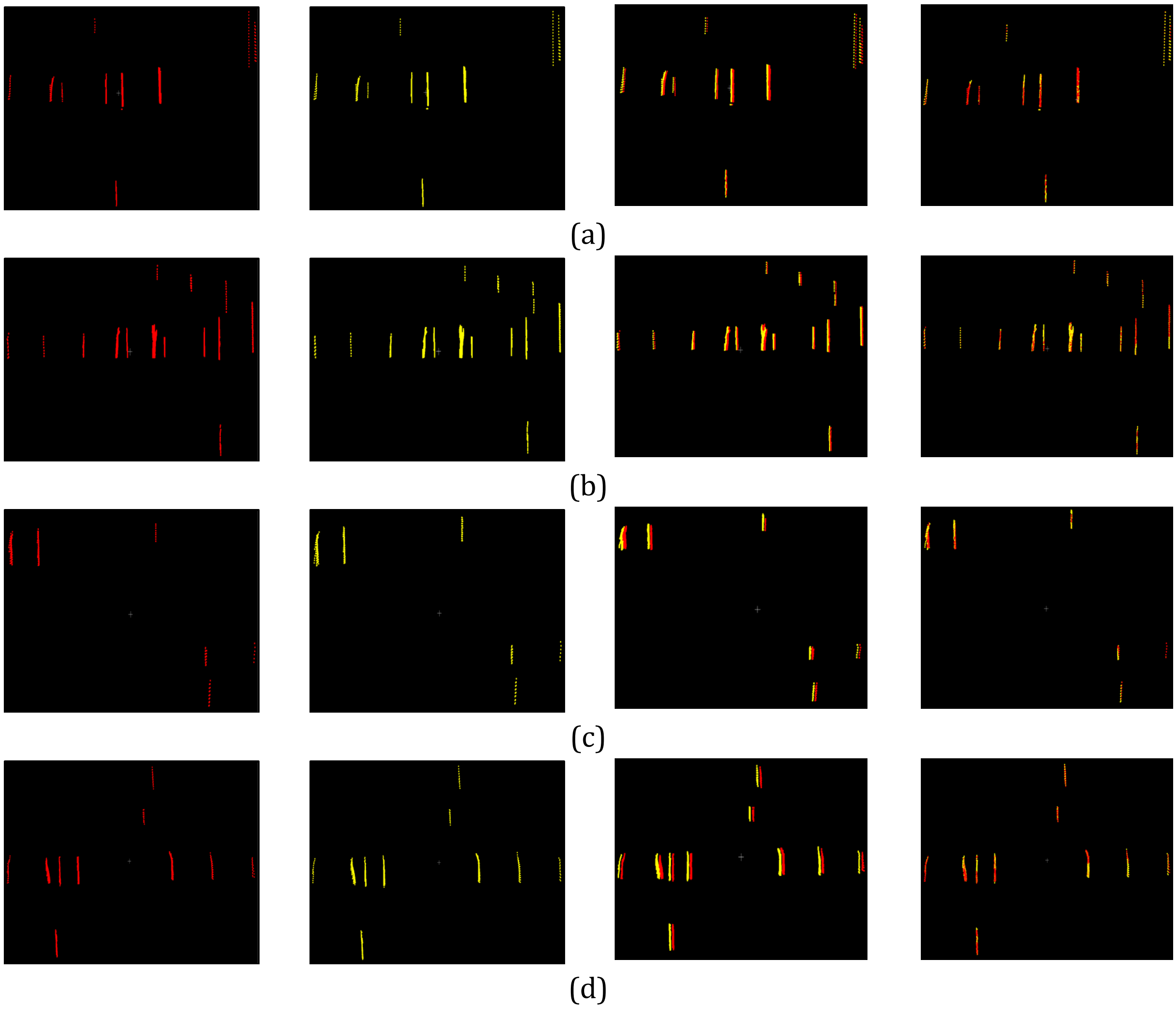

Figure 4. Adjacent frame extraction corresponding to rod-shaped object point cloud image

5. Conclusion

This study proposes a least squares deviation analysis method, combined with Euclidean clustering and centroid fitting techniques, to achieve efficient extraction of pole-like object features in adjacent frames of multi-line LiDAR data. The proposed method effectively identifies and matches corresponding pole-like object features between adjacent frames, providing reliable matching points for subsequent feature registration. Experimental results demonstrate that the method exhibits strong performance in terms of both accuracy and stability, significantly improving the extraction precision of pole-like object features and laying a solid foundation for enhancing frame-to-frame registration and localization accuracy.

References

[1] H. Li, T. Meng, X. Zhang, J. Wei, Y. Ma and Y. Liu: Improved Euclidean Clustering Point Cloud Segmentation Algorithm Based on Curvature Constraint, *Proc. Chinese Control and Decision Conference (CCDC)* (Yichang, China, 2023), Vol. 35, p. 2360-2365, doi: 10.1109/CCDC58219.2023.10326614.

[2] F. Chen et al.: Point Cloud Segmentation Algorithm Based on Improved Euclidean Clustering, *IEEE Access*, Vol. 12 (2024), p. 152959-152971, doi: 10.1109/ACCESS.2024.3480333.

[3] S.J. Miller: *The Method of Least Squares* (MS., Brown University, USA, 2006), p. 5-11.

[4] I. Markovsky and S. Van Huffel: Overview of Total Least-Squares Methods, *Signal Processing*, Vol. 87 (2007), p. 2283-2302, doi: 10.1016/j.sigpro.2007.04.004.

[5] M. Weinmann, B. Jutzi and C. Mallet: Geometric Features and Their Relevance for 3D Point Cloud Classification, *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*, Vol. IV-1/W1 (2017), p. 157-164, doi: 10.5194/isprs-annals-IV-1-W1-157-2017.

[6] Y. Zhang, G. Geng, X. Wei, S. Zhang and S. Li: A Statistical Approach for Extraction of Feature Lines from Point Clouds, *Computers & Graphics*, Vol. 56 (2016), p. 31-45, doi: 10.1016/j.cag.2016.01.004.

[7] X.-F. Han, Z.-A. Feng, S.-J. Sun and G.-Q. Xiao: 3D Point Cloud Descriptors: State-of-the-Art, *Artificial Intelligence Review*, Vol. 56 (2023), p. 12033-12083, doi: 10.1007/s10462-023-10486-4.

[8] R.B. Rusu, N. Blodow, Z.C. Marton and M. Beetz: Aligning Point Cloud Views Using Persistent Feature Histograms, *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems* (Nice, France, 2008), p. 3384-3391, doi: 10.1109/IROS.2008.4650967

[9] R.B. Rusu, N. Blodow and M. Beetz: Fast Point Feature Histograms (FPFH) for 3D Registration, *Proc. IEEE International Conference on Robotics and Automation* (Kobe, Japan, 2009), p. 3212-3217, doi: 10.1109/ROBOT.2009.5152473.

[10] R.B. Rusu, G. Bradski, R. Thibaux and J. Hsu: Fast 3D Recognition and Pose Using the Viewpoint Feature Histogram, *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems* (Taipei, Taiwan, 2010), p. 2155-2162, doi: 10.1109/IROS.2010.5651280.

[11] R.Q. Charles, H. Su, K. Mo and L.J. Guibas: PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Honolulu, HI, USA, 2017), p. 77-85, doi: 10.1109/CVPR.2017.16.

[12] C.R. Qi, L. Yi, H. Su and L.J. Guibas: *PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space*, 2017, DOI: 10.48550/arXiv.1706.02413