

Article

A Speedy Point Cloud Registration Method Based on Region Feature Extraction in Intelligent Driving Scene

Deli Yan ^{1,2,3} , Weiwang Wang ^{1,2}, Shaohua Li ^{3,*}, Pengyue Sun ², Weiqi Duan ² and Sixuan Liu ²

¹ Hebei Provincial Collaborative Innovation Center of Transportation Power Grid Intelligent Integration Technology and Equipment, Shijiazhuang Tiedao University, Shijiazhuang 050043, China; yandl@stdu.edu.cn (D.Y.); lef1357@163.com (W.W.)

² School of Electrical and Electronic Engineering, Shijiazhuang Tiedao University, Shijiazhuang 050043, China; pengyuesun25@gmail.com (P.S.); 18512413572@163.com (W.D.); liusixuan_echo@163.com (S.L.)

³ State Key Laboratory of Mechanical Behavior and System Safety of Traffic Engineering Structures, Shijiazhuang Tiedao University, Shijiazhuang 050043, China

* Correspondence: lshsjz@163.com

Abstract: The challenges of point cloud registration in intelligent vehicle driving lie in the large scale, complex distribution, high noise, and strong sparsity of lidar point cloud data. This paper proposes an efficient registration algorithm for large-scale outdoor road scenes by selecting the continuous distribution of key area laser point clouds as the registration point cloud. The algorithm extracts feature descriptions of the key point cloud and introduces local geometric features of the point cloud to complete rough and fine registration under constraints of key point clouds and point cloud features. The algorithm is verified through extensive experiments under multiple scenarios, with an average registration time of 0.5831 s and an average accuracy of 0.06996 m, showing significant improvement compared to other algorithms. The algorithm is also validated through real-vehicle experiments, demonstrating strong versatility, reliability, and efficiency. This research has the potential to improve environment perception capabilities of autonomous vehicles by solving the point cloud registration problem in large outdoor scenes.

Keywords: environment perception; cloud registration; feature extraction; rigid body change



Citation: Yan, D.; Wang, W.; Li, S.; Sun, P.; Duan, W.; Liu, S. A Speedy Point Cloud Registration Method Based on Region Feature Extraction in Intelligent Driving Scene. *Sensors* **2023**, *23*, 4505. <https://doi.org/10.3390/s23094505>

Academic Editors: Danilo Orlando and Pia Addabbo

Received: 10 April 2023

Revised: 28 April 2023

Accepted: 3 May 2023

Published: 5 May 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Accurately obtaining information about the environment around the vehicles is critical to the safety of autonomous vehicles [1]. Environment perception is a mechanism that provides a natural and dense understanding of the relationship between objects and their surroundings. It plays a critical role in enabling autonomous vehicles to operate safely and make informed decisions by classifying the importance of environmental factors and providing real-time and accurate information about the surroundings. Lidar is commonly used in self-driving cars to obtain real-time information about the vehicle's surroundings [2], and 3D point cloud data is used to perceive the changes in the environment as the car moves. However, 3D point cloud data usually contain noise, lack obvious spatial topological relationships, and have the problem of repeatedly scanning an object many times. In view of the above problems, current researchers solve them through point cloud registration technology. Point cloud registration is a fundamental problem in computer vision and plays a vital role in autonomous driving [3].

The Iterative Closest Point (ICP) algorithm [4] is a widely used point cloud registration technique. It works by establishing a distance threshold between two sets of point cloud data, identifying corresponding points between them, and iteratively solving for the optimal transformation matrix between the two sets. However, the accuracy and efficiency of the algorithm can be significantly affected by the initial position of the point cloud. In

response to this problem, some scholars have proposed several point cloud registration algorithms that combine ICP and different coarse registration algorithms [5–7]. To achieve the optimal estimation of the global point cloud, some scholars have proposed several different versions based on ICP [8–10]. In addition, some scholars proposed to use the neighborhood feature descriptor of the point cloud to accelerate point cloud registration [11–13]. This method only uses a small number of point clouds for point cloud registration, which can shorten the registration time of point clouds, but the registration accuracy of point clouds cannot be guaranteed. As deep learning continues to evolve, several point cloud registration algorithms based on deep learning have emerged [14–17]. However, these algorithms are currently limited to object-level or indoor point clouds and may not be suitable for large-scale outdoor autonomous-driving point clouds.

To improve the registration accuracy of lidar point clouds and overcome various challenges, several approaches have been proposed in the recent literature. NRLLI-UAV, proposed in [18], addresses the issue of registration failure when rigid transformations are used between images and lidar point clouds. In [19], the global registration of subway-tunnel point clouds is studied by using an enhanced extended Kalman filter and central axis constraints; [20] proposes a novel solution for point cloud registration in areas with low overlap. The data acquired by the oblique airborne photogrammetry system AOS-Tx8 are processed using a new processing scheme developed in [21], which aims to create large-scale regional thermal property maps. A 3D global localization method for identifying the location of objects in underground mines is proposed in [22]. Using mobile lidar mapping and point cloud registration, [23] proposes an adaptive feature region extraction method for simplifying point cloud data and improving registration accuracy. Ref. [24] proposes a registration algorithm that integrates road information constraints to improve the registration accuracy of Lidar-Inertial measurement unit (Imu) odometry in urban environments. Ref. [25] proposes a Lidar-Imu-Global Navigation Satellite System (GNSS) fusion positioning algorithm based on voxelized precise registration to address the problem of insufficient registration accuracy and cumulative errors. Ref. [26] proposes a pipeline internal localization method based on data fusion and point cloud registration to improve the positioning accuracy of pipeline robots. Ref. [27] proposes a symmetry-based lidar point registration method, which derives 3-D central axes from multi-source point clouds using object symmetry.

Intelligent-vehicle lidar point clouds have unique characteristics, such as high sparsity, large spatial extent, and complex and variable distributions [28], which make traditional point cloud registration algorithms inadequate. Many existing algorithms have the problems of slow convergence, long processing time, and single-application scenarios, making them unsuitable for intelligent-vehicle point cloud registration. This paper proposes a point cloud registration algorithm under the constraints of key point clouds and point cloud features, which solves the key problems of positioning or mileage calculation methods in intelligent driving and promotes the development of point cloud object detection and recognition.

In summary, our main contributions are as follows:

- (1) A new point cloud registration algorithm is proposed in this paper, which exhibits high accuracy, real-time performance, and reliability.
- (2) The selection of point clouds in key regions makes only a very small number of point clouds available for registration.
- (3) The local geometric features of the point cloud are introduced in our method to complete the point cloud registration process under the constraints of the key point cloud.

This research paper is structured as follows: Section 2 presents the point cloud registration algorithm proposed in this paper. In Section 3, the algorithm for the point cloud coarse registration phase is described. Section 4 elaborates the algorithm for the point cloud fine registration. Experimental validation is shown in Section 5. The conclusion of this research paper is given in Section 6.

2. Research Methods

In this section, the general framework of the algorithm and the method for continuous point cloud region extraction are introduced.

2.1. Algorithm Framework

The objective of point cloud registration is to determine the rigid body transformation matrix between two point clouds, such that the spatial distance between the transformed point cloud and the target point cloud is minimized. The mathematical language is expressed as follows [29]:

$$T = \operatorname{argmin} \sum_{i=1}^n (P_i - T \otimes P_i)^T \bullet (Q_i - T \otimes Q_i) \quad (1)$$

In Equation (1), T is the rigid body transformation matrix between the two sets of point clouds, and P_i and Q_i ($i = 1, 2, 3 \dots n$) are the n pairs of matching points in the two sets of point clouds. The overall framework of the point cloud registration algorithm proposed in this paper is shown in Figure 1.

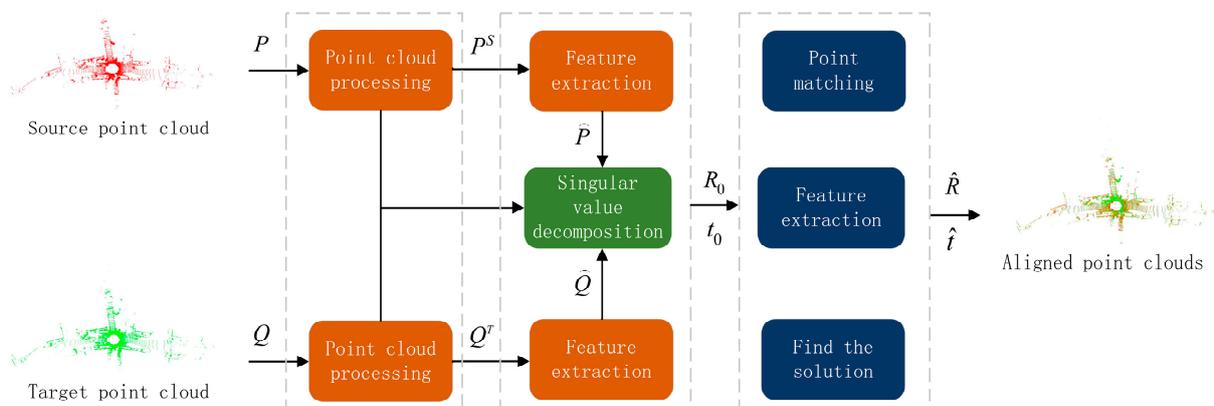


Figure 1. The algorithm framework.

Given source and target point clouds $P, Q \in \mathbb{R}^{N \times 3}$, the key point clouds $P^S = \{P_1^S, P_2^S, P_3^S \dots P_i^S\}$ and $Q^T = \{Q_1^T, Q_2^T, Q_3^T \dots Q_i^T\}$ are obtained by using the continuous key point cloud selection module of this paper. Coarsely registered point clouds $\widehat{P} = \{\widehat{P}_1, \widehat{P}_2, \widehat{P}_3 \dots \widehat{P}_a\}$ and $\widehat{Q} = \{\widehat{Q}_1, \widehat{Q}_2, \widehat{Q}_3 \dots \widehat{Q}_a\}$ are obtained under the Fast Point Feature Histogram (FPFH) feature constraint [30], which is used to solve the initial rotation matrix R_0 and initial translation vector t_0 . After coarse registration, the relative positional relationship between point clouds is improved. Based on the evaluation of the point cloud normal vector and curvature, the optimal transformation matrix $\widehat{T} = \begin{bmatrix} \widehat{R} & \widehat{t} \end{bmatrix}$ is solved iteratively. \widehat{R} represents the rotation matrix, and \widehat{t} represents the translation vector.

2.2. Region of Interest Area

The pre-processed point cloud data in Figure 2 appears to be cleaner and more organized compared to the raw point cloud data, which may contain more noise and invalid points. The pre-processing step helps to remove these unwanted points and improve the quality of the point cloud data, which can lead to more accurate point cloud registration results.

It should be noted that the distribution of point clouds in intelligent driving scenarios is complex and changeable, and there is no obvious spatial topological relationship between point clouds. Figure 3 shows the vertical and horizontal distribution of point clouds in such scenes. To avoid focusing on a single scene, experiments on the KITTI dataset [31] (dataset for research in the field of autonomous driving jointly sponsored by Karlsruhe Institute of Technology and Toyota Technological University Chicago) have shown that

point clouds are mainly distributed in the regions of $(-26.7083, 20.5833)$ m in the x direction and $(-20.2343, 23.9661)$ m in the y direction. Therefore, the selected Region of Interest (ROI) for point cloud registration should be within this range.

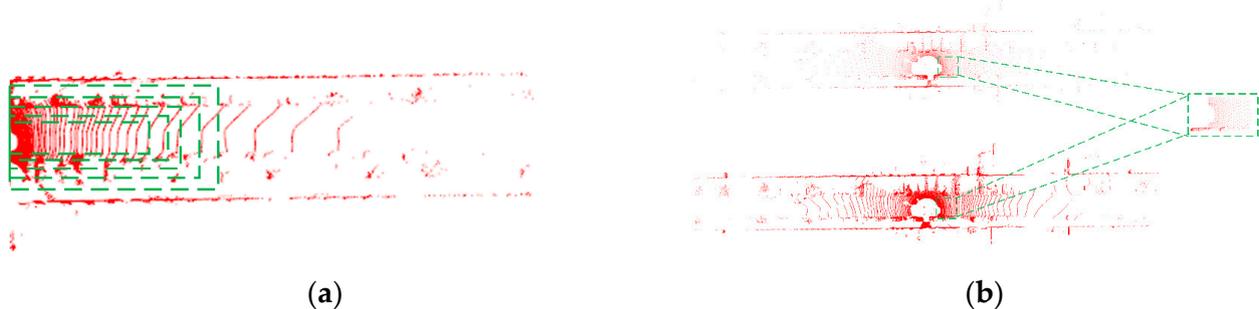


Figure 2. The algorithm framework: (a) Region of Interest region selection; (b) point cloud data preprocessing.

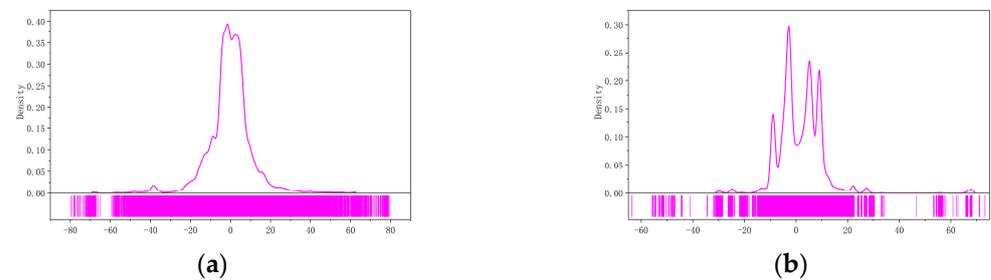


Figure 3. Vertical and horizontal distribution of point clouds: (a) x distribution; (b) y distribution.

In this study, different ROI regions are delineated by adjusting the vertical and horizontal thresholds of the point cloud. Figure 4 shows the impact of different ROI regions on the accuracy and efficiency of point cloud registration. Among them, the overall effect of ROI area 1 is ideal.

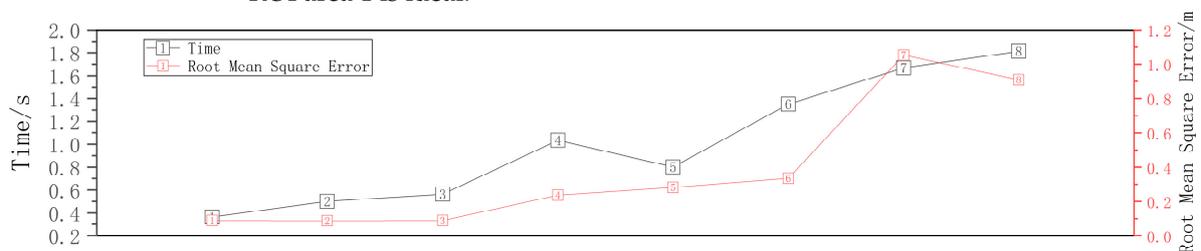


Figure 4. Time and root mean square error values for different ROI sizes.

3. Point Cloud Coarse Registration

In this section, point clouds are characterized by FPFH features, and the initial change matrix T_0 between two frames of point clouds is computed.

3.1. FPFH Feature Descriptor

Different from the Point Feature Histogram (PFH) [32], the FPFH reduces the computational complexity from $O(nk^2)$ to $O(nk)$ by weighting the Simple Point Feature Histogram (SPFH). n represents the number of calculation points, and k represents the number of points within the radius of the calculation point field. Figure 5 is a comparison of the affected areas of the PFH and the FPFH.

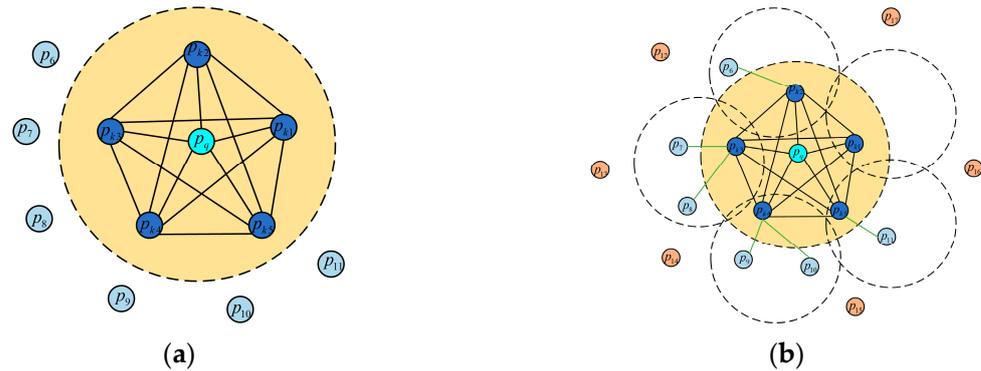


Figure 5. Comparison of influence areas: (a) PFH influence area; (b) FPFH influence area.

For every pair of points p_i and p_j ($i \neq j$) in the k -neighborhood of each point p and their estimated normal n_i and n_j , we compute the angular variations of n_i and n_j as follows:

$$\alpha = (p_j - p_i) \times n_i \bullet n_j \quad (2)$$

$$\phi = (n_i \bullet (p_j - p_i)) / \|p_j - p_i\| \quad (3)$$

$$\theta = \arctan(n_i \times (p_j - p_i) \times n_i \bullet n_j, n_i \bullet n_j) \quad (4)$$

In Equations (2)–(4), n_i and n_j are the estimated normal vectors of points p_i and p_j , $\theta \in (0, 2\pi)$. For each query point p , we compute the relationships (as shown in Equations (2)–(4)) between itself and its nearest neighbors, which is referred to as the “SPFH” calculation. For every point, we re-determine its nearest neighbors and utilize the SPFH values from k neighbors to weight the final calculation of the point’s feature descriptor, known as the “FPFH” descriptor:

$$FPFH(p) = SPFH(p) + \frac{1}{k} \sum_{i=1}^k \frac{1}{\omega_k} SPFH(p_k) \quad (5)$$

In Equation (5), ω_k represents the distance between query point p and a neighbor point p_k in a given metric space.

3.2. Singular Value Decomposition to Solve the Transformation Matrix

Singular Value Decomposition (SVD) [33] is a commonly used method for solving the rigid transformation between two point clouds by minimizing the least squares error. By computing the covariance matrix of the two point clouds, SVD can extract the rotation matrix and translation vector that align the two point clouds in the same coordinate system.

For the matched point sets $\widehat{P} = \{\widehat{P}_1, \widehat{P}_2, \widehat{P}_3 \dots \widehat{P}_a\}$ and $\widehat{Q} = \{\widehat{Q}_1, \widehat{Q}_2, \widehat{Q}_3 \dots \widehat{Q}_a\}$, the initial change matrix $T_0 = [R_0 \ t_0]$ is solved to minimize the spatial distance of the registered corresponding points, that is, to find the minimum error function:

$$E(R_0 \ t_0) = \sum_{i=1}^a \left\| \widehat{Q}_i - R_0 \widehat{P}_i - t_0 \right\|^2 \quad (6)$$

In Equation (6), \widehat{Q}_i and \widehat{P}_i represent the points in the matched point sets \widehat{P} and \widehat{Q} . To find the rotation matrix R_0 , we change Equation (6) as follows:

$$E(R_0) = \sum_{i=1}^a \left(\left(\widehat{Q}_i - \widehat{Q}_0 \right)^T \left(\widehat{Q}_i - \widehat{Q}_0 \right) + \left(\widehat{P}_i - \widehat{P}_0 \right)^T \left(\widehat{P}_i - \widehat{P}_0 \right) - 2 \left(\widehat{Q}_i - \widehat{Q}_0 \right)^T R_0 \left(\widehat{P}_i - \widehat{P}_0 \right) \right) \quad (7)$$

In Equation (7), \widehat{Q}_0 and \widehat{P}_0 are the centroids of point sets \widehat{Q} and \widehat{P} . To minimize $E(R_0)$, the calculation needs to take the derivative of $E(R_0)$ and maximize $dE(R_0)$.

$$dE(R_0) = tr \left(R_0 \left(\widehat{P}_i - \widehat{P}_0 \right) \left(\widehat{Q}_i - \widehat{Q}_0 \right)^T \right) \tag{8}$$

In Equation (8), $\left(\widehat{P}_i - \widehat{P}_0 \right) \left(\widehat{Q}_i - \widehat{Q}_0 \right)^T$ is a third-order square matrix. In Equation (9), we denote $\left(\widehat{P}_i - \widehat{P}_0 \right) \left(\widehat{Q}_i - \widehat{Q}_0 \right)^T$ by H .

$$H = U \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} V^T \tag{9}$$

In Equation (9), U and V are orthogonal matrices. $\lambda_1, \lambda_2, \lambda_3$ are the eigenvalues of the matrix H . The rotation matrix R_0 can be obtained by the following formula:

$$R_0 = VU^T \tag{10}$$

According to the obtained rotation matrix R_0 , the translation vector t_0 is obtained by the following formula:

$$t_0 = \widehat{Q}_0 - R_0 \frac{1}{a} \sum_{i=1}^a \widehat{P}_i \tag{11}$$

4. Point Cloud Fine Registration

In this section, local features of the point cloud are introduced, and the transformation matrix between matched point pairs is computed.

4.1. Extracting Point Cloud Features

Different from ICP, Normal Iterative Closest Point (NICP) [34] takes the local features of the point cloud into consideration when matching two frames of point clouds, and uses the Levenberg-Marquardt (LM) algorithm to iteratively solve the point cloud transformation matrix. Figure 6 shows the principle of NICP.

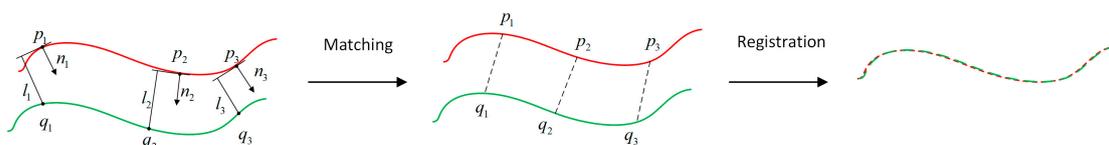


Figure 6. The principle of NICP. The red line represents the source point cloud, and the green line represents the target point cloud.

To solve for the normal vector, we compute the covariance matrix of the Gaussian distribution of all points within a sphere of radius R around the target point p_i . If the point cloud surfaces in the target point domain are well-defined, they can be approximated as a plane, and only one of the eigenvalues of the covariance is much smaller than the other two. The eigenvector corresponding to the smallest eigenvalue is considered to be the normal vector of the point cloud plane of this point domain.

$$\mu_i^s = \frac{1}{|v_i|} \sum_{p_j \in v_i} p_j \tag{12}$$

$$\sum_i^s = \frac{1}{|v_i|} \sum_{p_j \in v_i} (p_j - \mu_i^s)^T (p_j - \mu_i^s) \tag{13}$$

In Equation (12), μ_i^s represents the centroid of the point cloud, and v_i represents a cluster of point clouds in the field of a sphere of radius R in the target point cloud p_i . In Equation (13), Σ_i^s is the Gaussian distribution covariance matrix.

$$\Sigma_i^s = X \begin{bmatrix} \omega_1 & 0 & 0 \\ 0 & \omega_2 & 0 \\ 0 & 0 & \omega_3 \end{bmatrix} Y^T \quad (14)$$

In Equation (14), X and Y denote an orthogonal matrix after decomposing the matrix Σ_i^s . $\omega_1, \omega_2, \omega_3$ are the eigenvalues of Σ_i^s in ascending order.

4.2. Searching for Matching Point Pairs

The points of the two sets of point clouds are projected into the same depth map, and those points that fall on the same pixel and have consistent normal vectors and curvatures are considered as matching point pairs. If multiple point clouds fall on the same pixel, the closest point pair is chosen, and their normal vectors must point to the same point. These alternative matching point pairs will be filtered by the following conditions, and if at least one of the following conditions is met, the group of point pairs will be removed.

The distance between two points exceeds a threshold:

$$\|p_i^c - \hat{T} \oplus p_j^r\| > \varepsilon_d \quad (15)$$

The absolute value of the difference between the logarithms of curvature of two points exceeds a threshold:

$$|\log \sigma_i^c - \log \sigma_j^r| > \varepsilon_\sigma \quad (16)$$

The angle between the normal vectors of two points exceeds a threshold:

$$n_i^c \bullet \hat{T} \oplus n_j^r < \varepsilon_n \quad (17)$$

In Equation (15), ε_d represents the distance threshold, \hat{T} represents the projection matrix, and p_i^c and p_j^r represent two matched points. In Equation (16), ε_σ represents the curvature threshold, and σ_i^c and σ_j^r represent the curvatures of the two matching points. In Equation (17), ε_n represents the angle threshold, and n_i^c and n_j^r represent the normal vectors of the two matching points.

4.3. Calculating the Transformation Matrix

When given a pair of matching points and the current transformation \hat{T} , the error function $e_{ij}(\hat{T})$ is a six-dimensional vector.

$$e_{ij}(\hat{T}) = (\tilde{p}_i^c - \tilde{p}_j^r) \quad (18)$$

In Equation (18), \tilde{p}_i^c and \tilde{p}_j^r represent two matching points with normal vectors. Thus, the objective function composed of all point pairs is expressed as follows:

$$\sum_c e_{ij}(\hat{T})^T \tilde{\Omega}_{ij} e_{ij}(\hat{T}), \quad (19)$$

In Equation (19), $\tilde{\Omega}_{ij}$ is a 6×6 information matrix. The NICP algorithm uses a local parameterization of incremental perturbations to minimize the objective function $\sum_c e_{ij}(\hat{T})^T \tilde{\Omega}_{ij} e_{ij}(\hat{T})$; the increment is expressed as $\Delta T = (\Delta t_x \ \Delta t_y \ \Delta t_z \ \Delta q_x \ \Delta q_y \ \Delta q_z)^T$, which contains the translation vector Δt and the imaginary part Δq of the rotation unit quaternion. By using the damped Gauss-Newton algorithm in each iteration of the calculation:

$$(H + \lambda I)\Delta T = b \quad (20)$$

In Equation (20), $H = \sum J_{ij}^T \tilde{\Omega} J_{ij}$ is the approximated Hessian matrix, $b = J_{ij}^T \tilde{\Omega} e_{ij}(\hat{T})$. The increment ΔT can be calculated by Equation (20), and \hat{T} is updated.

$$\hat{T} \leftarrow \Delta T \oplus \hat{T} \quad (21)$$

J_{ij} represents the Jacobian calculation, which is defined as follows:

$$J_{ij} = \frac{\delta e_{ij}(\Delta T \oplus \hat{T})}{\delta \Delta T} \quad (22)$$

5. Experiments

This section begins with the visual and experimental data analysis of the proposed algorithm under various working conditions and scenarios. The raw data for the experiment are obtained from two lidars with completely different metrological characteristics. The point cloud registration experiment is conducted on a computer hardware environment consisting of an Intel(R) Core™ i5-11400H CPU and 16 GB memory.

5.1. Object-Level Point Cloud Registration Experiment

Outdoor point clouds can be considered as being composed of object-level factor clouds. Therefore, firstly, we used the rabbit point cloud from the public point cloud library provided by Stanford University for registration experiments and compared it with different common point cloud registration algorithms such as Normal Distribution Transformation (NDT), Trimmed Iterative Closest Point (TRICP), and NICP. Table 1 indicates the experimental effects of distinct registration algorithms under the rabbit point cloud of Stanford University.

Table 1. Registration results of Stanford University rabbit.

Algorithm	Root Mean Square Error/m	Time/s
NDT	0.032	29.32
TRICP	0.036	22.71
NICP	0.034	7.46
Ours	0.030	1.53

The number of point clouds of the Stanford University rabbit model is about 40,000, which can be compressed to about 6000 after speedy sampling of the point cloud. From the consequences of rabbit registration at Stanford University in Table 1, it can be viewed that the running time of the algorithm proposed in this paper is 1.53 s. Compared with the different three registration algorithms, the time is significantly shortened, and the average registration time is shortened by 12.96 times. At the same time, the registration accuracy is slightly improved, and the average registration accuracy is increased by 11.56%. Figure 7 indicates the registration impact of different registration algorithms under the rabbit point cloud of Stanford University.

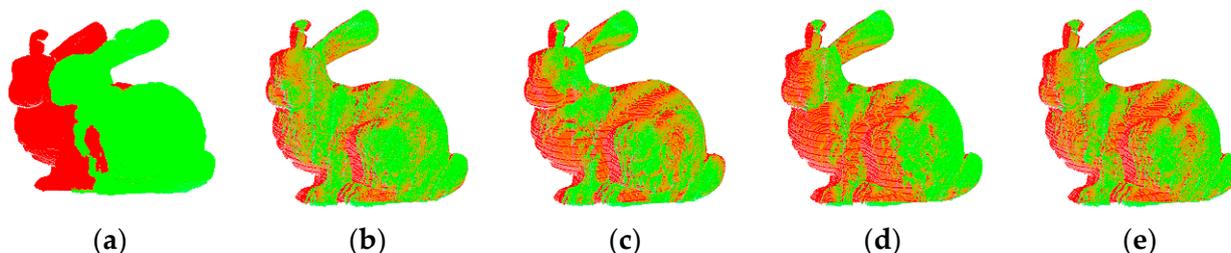


Figure 7. Stanford University Rabbit renderings. (a) Primitive point, (b) NDT, (c) TRICP, (d) NICP and (e) OURS.

5.2. Multi-Condition Registration Experiment

The KITTI dataset is a widely used benchmark dataset for evaluating algorithms related to autonomous driving, including point cloud registration. It contains data from a variety of urban, rural, and highway scenes with different levels of complexity, making it a good dataset for testing the robustness and accuracy of algorithms under different conditions. The Velodyne HDL-64E lidar used in the KITTI dataset is a high-performance sensor that can capture detailed and accurate point cloud information with a large coverage range, making it suitable for real-world autonomous driving applications. Figure 8 shows the point cloud distribution under different working conditions.

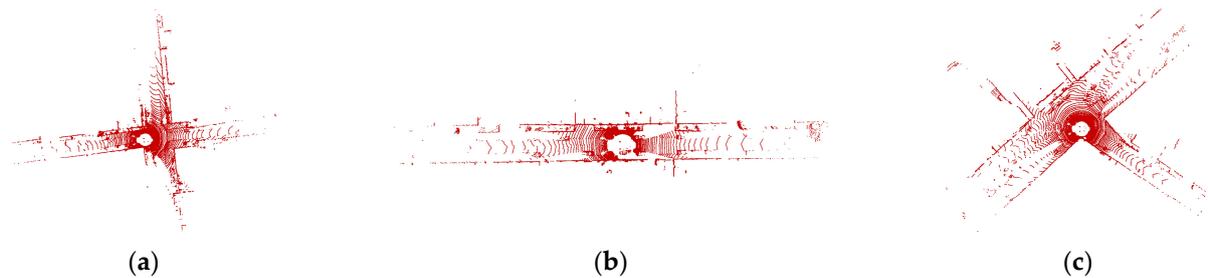


Figure 8. Different working conditions of point cloud registration. (a) Crossroads, (b) vehicle going straight, and (c) vehicle turning.

Table 2 shows the time registration results of different registration algorithms under the KITTI mileage dataset. In Table 2, scenarios 2, 3, and 6 are intersection conditions, scenarios 1, 4, and 8 are straight-going conditions, and scenarios 5, 7, and 9 are turning conditions.

Table 2. Time comparison of algorithms.

Method	Time/s								
	Scene1	Scene2	Scene3	Scene4	Scene5	Scene6	Scene7	Scene8	Scene9
NDT	13.80	12.50	12.75	15.47	18.48	14.83	19.42	16.08	14.32
TRICP	12.24	10.37	14.43	11.98	15.72	11.73	7.18	13.51	12.20
NICP	12.68	12.67	19.17	13.97	11.08	12.21	13.86	15.07	15.31
Ours	0.63	0.38	0.66	0.49	0.74	0.59	0.62	0.89	0.90

It is clear from Table 2 that the proposed algorithm has a significantly faster registration time compared to the other three algorithms, with an average registration time of 0.66 s. The NDT algorithm has the slowest registration time, with an average of 15.29 s, while the TRICP and NICP algorithms have registration times of 12.15 s and 14.00 s on average, respectively. In terms of registration accuracy, Figure 9 shows that the proposed algorithm also outperforms the other three algorithms, with an average root mean square error of 0.0787 m. On average, the proposed algorithm achieves an improvement of 17.75% in accuracy compared to the other three algorithms.

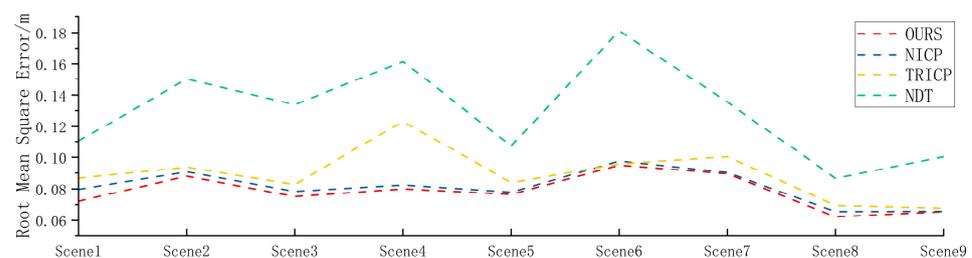


Figure 9. Accuracy comparison of multi-condition algorithms.

Figure 10 is a comparison of the effects of different point cloud registration algorithms under the KITTI mileage data. Among them, the registration effect of the NDT and NICP algorithms is not ideal, and the coincidence degree between the two frame point clouds after registration is not high. The TRICP registration effect is better, but the registration time is too long to meet the actual needs. It can be seen from Figure 10 that after the two frames of point clouds are registered by the algorithm of this paper, the ground, buildings, vehicles, and other objects have achieved a high degree of fusion. Moreover, the corresponding relationship between point clouds is very stable and will not change due to external interference. At the same time, the correspondence between point clouds is very complete, and there are no missing or repeated point clouds. The algorithm proposed in this paper has strong adaptability and can meet the needs of point cloud registration in different working conditions.

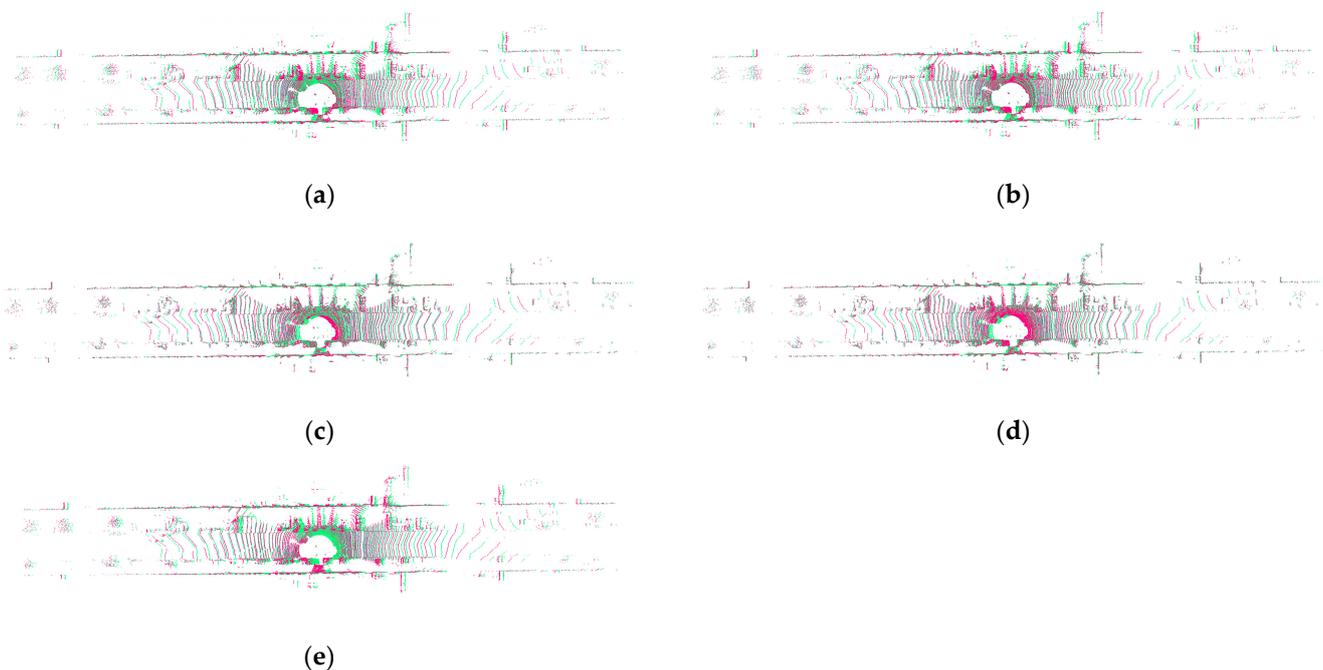


Figure 10. The registration effect of different algorithms under the straight driving condition in the street. (a) NDT, (b) NICP, (c) Ours, (d) TRICP, and (e) primitive point.

5.3. Multi-Scene Registration Experiment

The distribution of point clouds in different scenarios is very different, which brings great challenges to point cloud registration. In order to verify the versatility of the algorithm proposed in this paper in different scenarios, several frames of data in urban, rural, and road scenes were selected from the KITTI dataset to conduct extensive verification experiments. Figure 11 is the distribution of point clouds in different scenarios.

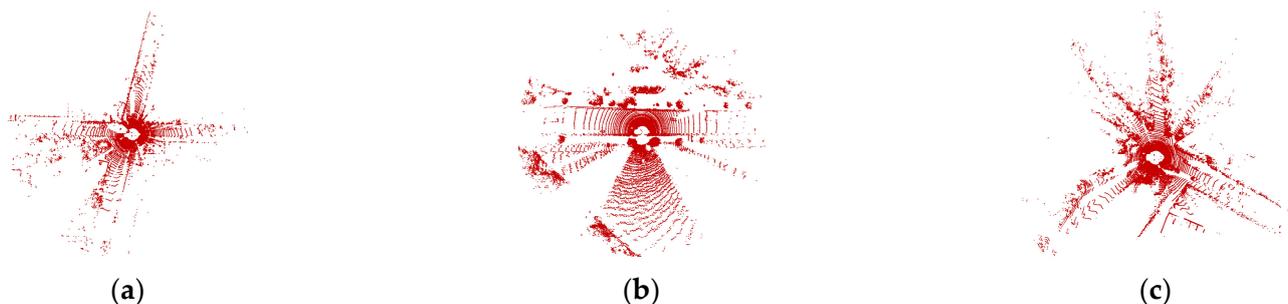


Figure 11. Different scenarios for point cloud registration. (a) City scene, (b) road scene, and (c) rural scene.

For the verification experiments, a total of 1059 frames of road point cloud data were selected. Figure 12 shows the temporal analysis of point cloud registration for four different algorithms in road scenes. The average point cloud registration time for the proposed algorithm in this paper is 0.5084 s, compared to 8.7548 s for the TRICP algorithm, 22.3652 s for the NICP algorithm, and 31.0748 s for the NDT algorithm.

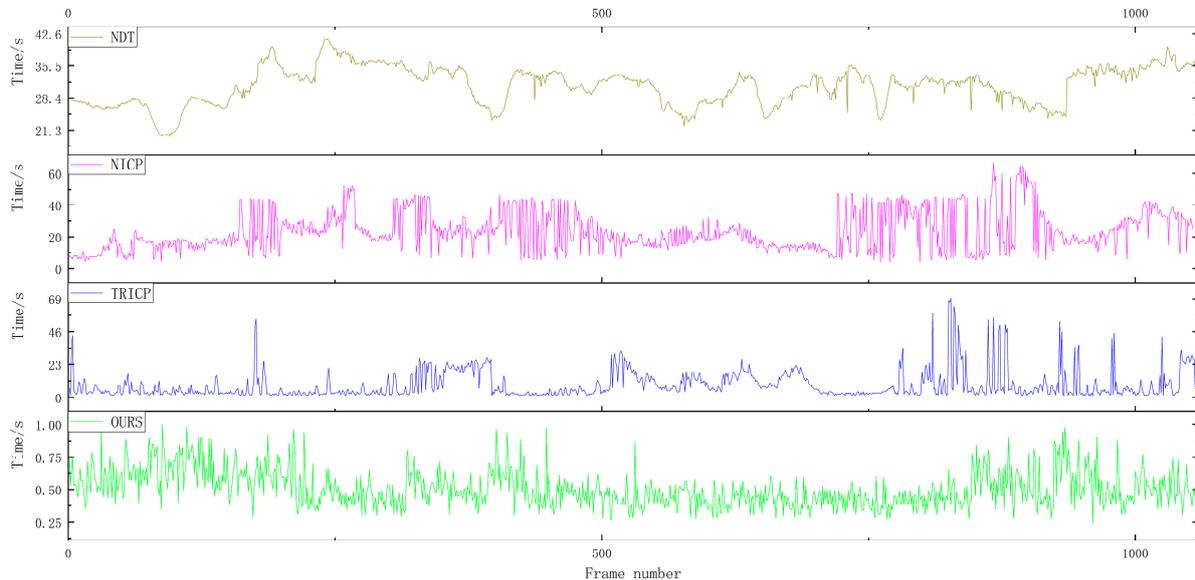


Figure 12. Time comparison of the four algorithms in the road scene.

A total of 1264 frames of rural point cloud data were selected for verification experiments. Figure 13 shows the time analysis of four different algorithms for point cloud registration in rural scenes. The proposed algorithm in this paper has an average point cloud registration time of 0.6555 s, which is significantly shorter compared to the TRICP algorithm with an average point cloud registration time of 11.8145 s, the NICP algorithm with an average point cloud registration time of 16.5529 s, and the NDT algorithm with an average point cloud registration time of 19.4471 s.

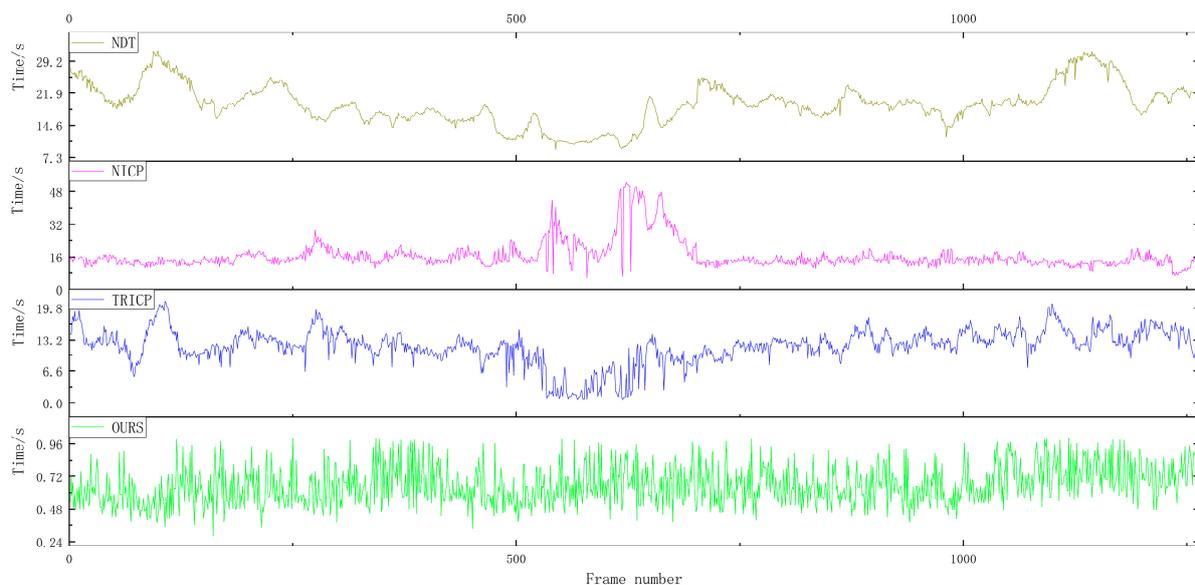


Figure 13. Time comparison of the four algorithms in the rural scene.

A total of 1469 frames of urban point cloud data were selected for verification experiments. Figure 14 shows the time analysis of four different algorithms for point cloud

registration in urban scenes. For the verification experiments, 1469 frames of urban point cloud data were selected. The average point cloud registration time of the proposed algorithm was 0.5855 s, while the average registration times of the TRICP, NICP, and NDT algorithms were 10.7177, 14.0146, and 21.5313 s, respectively.

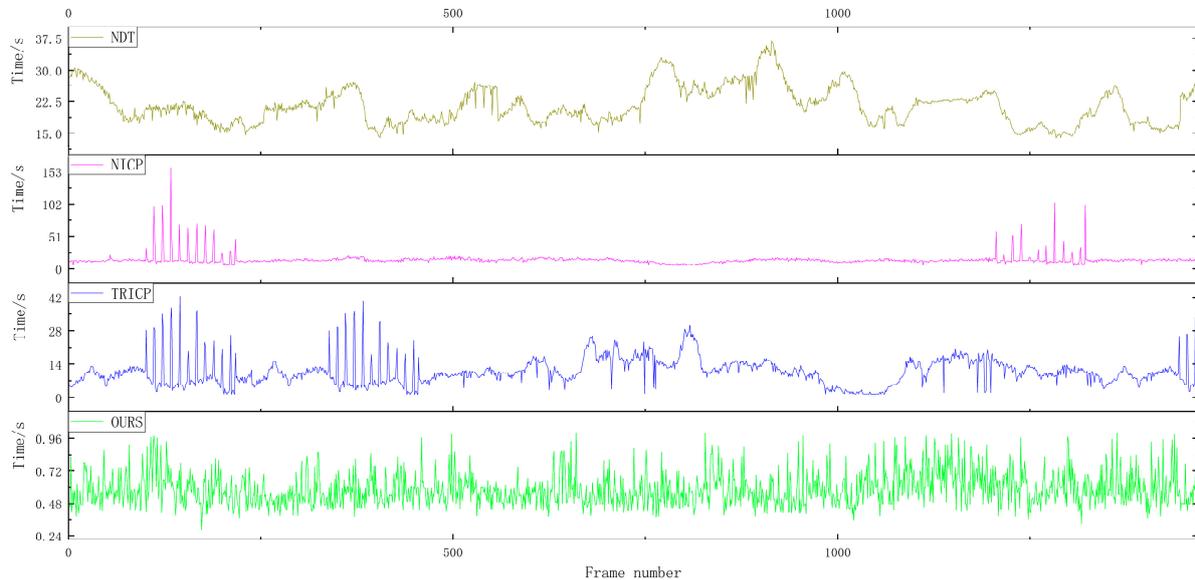


Figure 14. Time comparison of the four algorithms in the urban scene.

Among the four point cloud registration algorithms, NDT takes the longest time, and the registration efficiency varies greatly in different scenarios; the registration efficiency of NICP and TRICP algorithms is relatively stable, but the registration time in some scenarios is too long and the overall registration time cannot meet the real-time requirements. In the road, rural, and urban scenarios, the algorithm proposed in this paper maintains a stable calculation time, and the average registration time is 0.5831 s. Thanks to the selection of the ROI area of the point cloud in front of the vehicle, it avoids large-scale searching for the corresponding relationship between point clouds, which can greatly shorten the time for point cloud registration.

Table 3 shows the accuracy comparison of various algorithms in different scenarios. The average registration value of the NDT algorithm is 0.1268 m, the average registration value of the NICP algorithm is 0.08910 m, and the average registration value of the TRICP algorithm is 0.1011 m. Among them, the registration accuracy of the NDT algorithm is stable, but the error is large and does not meet the actual needs; the TRICP and NICP algorithms maintain excellent performance in some scenarios, but the algorithms are not universal and cannot adapt to different scenarios. The average value of the algorithm proposed in this paper is 0.06996 m. In road, rural, and urban scenarios, the algorithm in this paper maintains high accuracy and stability. Figure 15 is a comparison of multiple algorithms in different scenarios.

Table 3. Comparison of accuracy of algorithms.

Method	Road/m	Countryside/m	City/m
NDT	0.1427	0.1136	0.1241
TRICP	0.1346	0.08100	0.08766
NICP	0.1108	0.07548	0.08101
OURS	0.07231	0.06991	0.06766

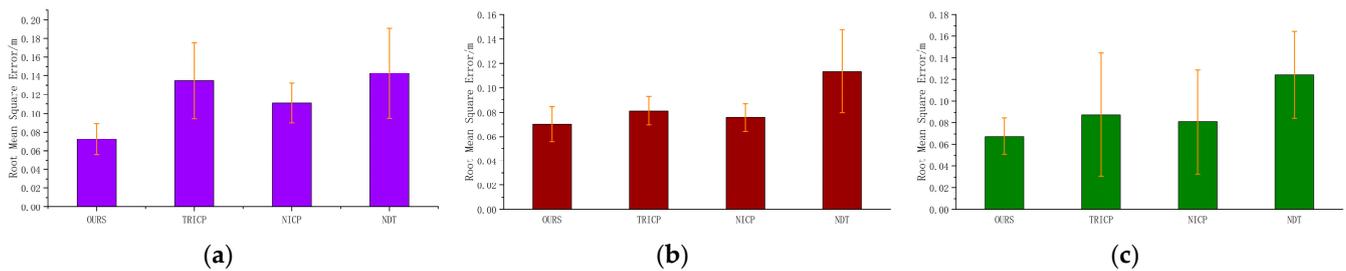


Figure 15. Accuracy comparison of four algorithms in different scenarios. (a) Road scene, (b) rural scene, and (c) urban scene.

5.4. Real-Vehicle Registration Experiment

To evaluate the effectiveness of the proposed point cloud registration algorithm, experiments were conducted using an electric vehicle with a Hesai 40-line lidar, a Changjiang No. 3 camera, and an inertial measurement unit, driven by the researchers at the State Key Laboratory. The sensors were rigidly connected to establish a stable coordinate system conversion relationship, and after data processing and coordinate system conversion, accurate three-dimensional space coordinates of each scanned object were obtained. Table 4 shows the lidar sensor parameters.

Table 4. Lidar parameters.

Technical Parameter			
Principle of distance measurement	time-of-flight measurement	Scanning frequency	10 Hz, 20 Hz
Scanning principle	mechanical rotation	Vertical field of view	40° (−25~+15°)
Number of threads	40	Vertical angular resolution	minimum 0.33°
Detection distance	0.3~200 m	Horizontal field of view	360°
Measurement accuracy	±5 cm (0.3~0.5 m) ±2 cm (0.5~200 m)	Horizontal angular resolution	0.2° (10 Hz) 0.4° (20 Hz)

The test area for the study was mainly located in the Shijiazhuang Railway University headquarters campus, which is composed of campus roads. The road conditions in this area are relatively flat, without any significant bumpy road sections. The data collection process involved collecting point clouds of various objects, including pedestrians, roads, trees, buildings, and speed bumps around the vehicle body. The collected data was very accurate, complete, and without any missing or repeated point clouds. This allowed for a complete reflection of the scene's characteristics and details, facilitating comprehensive analysis and processing. The data coverage was extensive, covering every corner and detail of the entire scene. Figure 16 shows the equipment installation location and real-vehicle data collection environment.



Figure 16. General situation of real-vehicle experiment. (a) Experimental platform, (b) straight driving condition, and (c) intersection condition.

A total of 1761 frames were selected from the collected point cloud data for verification experiments. Figure 17 shows the point cloud registration time analysis of the real vehicle. The average point cloud registration time of the algorithm proposed in this paper is 0.5734 s, the average point cloud registration time of the TRICP algorithm is 17.8066 s, the average point cloud registration time of the NICP algorithm is 21.1194 s, and the average point cloud registration time of the NDT algorithm is 27.8159 s. The time efficiency of the algorithm proposed in this paper still maintains a high real-time performance under the real-vehicle test. It can be seen from Figure 17 that the longest registration time of NDT is 47.3280 s, the longest registration time of NICP is 114.8751 s, and the longest registration time of TRICP is 82.8434 s. The algorithm in this paper maintains high stability, and there is no phenomenon where the registration time of a certain point cloud is too long.

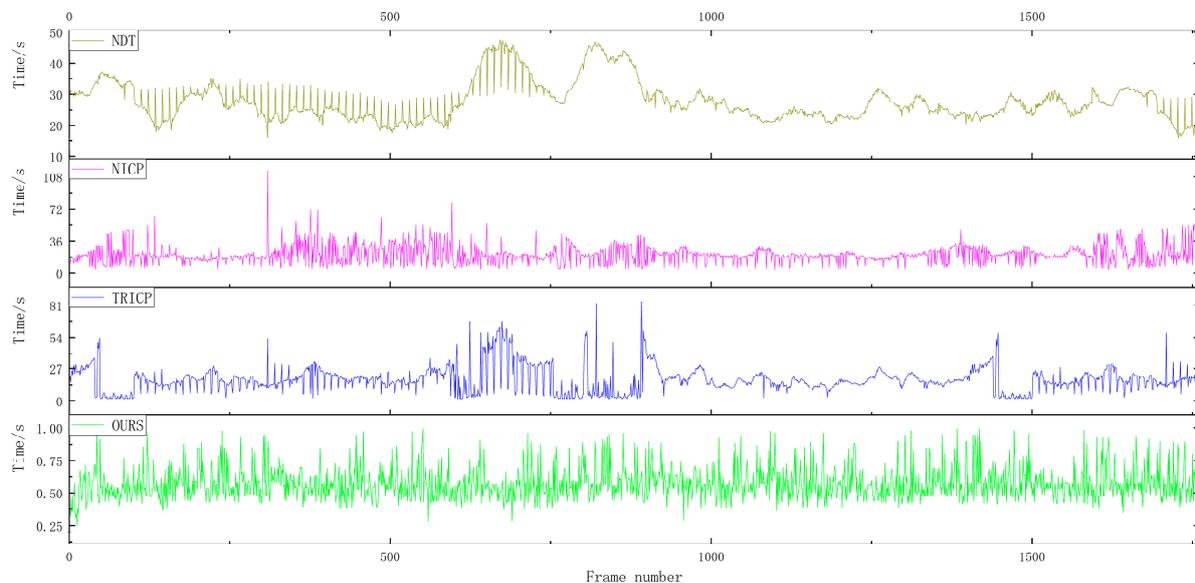


Figure 17. Analysis of registration time of real-vehicle point cloud.

Table 5 shows the accuracy analysis of various algorithms under real-vehicle experiments. Among them, the maximum error range of the registration accuracy of the TRICP algorithm is 0.06505 m, followed by NICP and NDT. The minimum error range of the algorithm proposed in this paper is 0.01751 m. At the same time, compared with the other three algorithms, the algorithm proposed in this paper has the highest registration accuracy. Experimental results demonstrate the versatility and high accuracy of the registration algorithm proposed in this paper under real-vehicle conditions. Figure 18 is the analysis of the registration accuracy of the real-vehicle point cloud.

Figure 19 presents a visual analysis of four registration algorithms using data from real-vehicle experiments. The NDT, TRICP, and NICP registration algorithms show a low degree of coincidence, with significant differences in objects such as buildings and vehicles. On the other hand, after registration by the algorithm proposed in this paper, the degree of coincidence between point clouds is high, and the corresponding relationship is strong. Additionally, the proposed algorithm does not ignore any part of the point cloud, and the relationship between each point is preserved.

Table 5. Comparison of algorithm accuracy in real vehicle.

Algorithm	Root Mean Square Error/m
NDT	0.1229 ± 0.03925
NICP	0.1182 ± 0.04029
TRICP	0.1244 ± 0.06505
OURS	0.05996 ± 0.01751

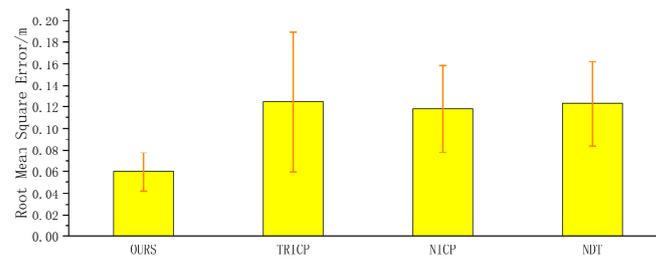


Figure 18. Accuracy analysis of point cloud registration in real vehicle.

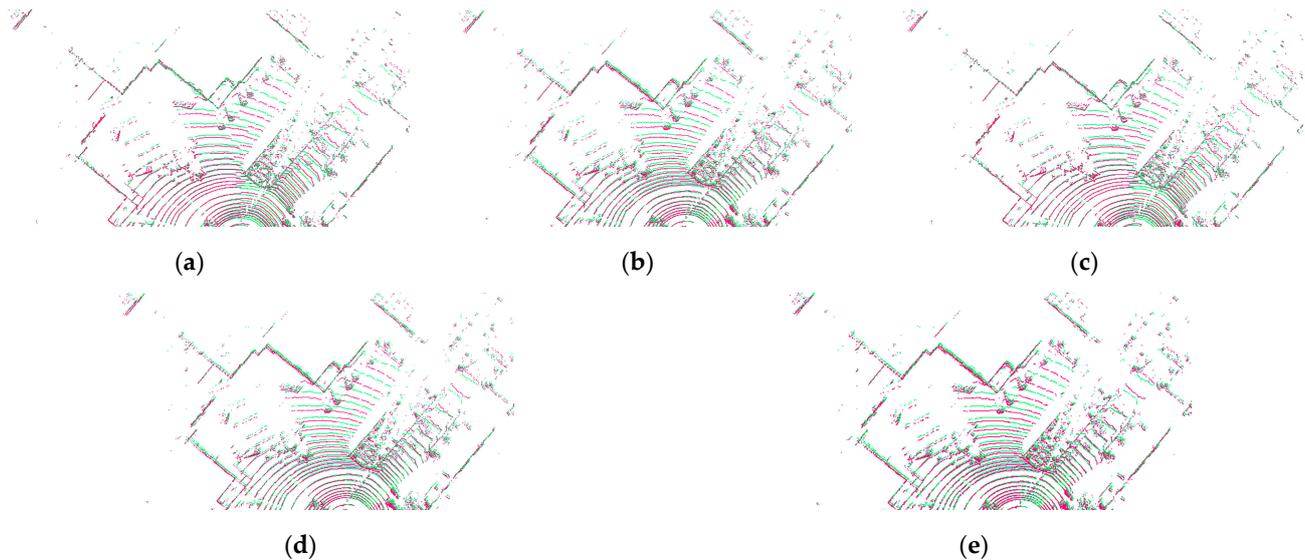


Figure 19. Visual analysis of different registration algorithms. (a) NDT, (b) NICP, (c) Ours, (d) TRICP, and (e) primitive point.

5.5. Metrological Characteristics Analysis of Lidar

The metrological characteristics of lidar include not only the measurement accuracy and angular resolution, but also other important factors, such as measurement range, point density, and reflectivity detection threshold. All of these characteristics can collectively affect the quality and accuracy of point cloud data collected by lidar, which can impact the result of point cloud registration. For example, the measurement accuracy of lidar determines its ability to accurately capture the 3D location of a target object. If the accuracy is low, the point cloud data collected by lidar will contain significant errors, which can negatively affect point cloud registration results. The point cloud data sources of the experiment are two different lidars, Pandar40P and Velodyne HDL-64E. Their metrological parameters are shown in Table 6.

Table 6. Metrological parameters of Pandar40P and Velodyne HDL-64E.

Device Model	Measuring Distance	Ranging Accuracy	Horizontal Field of View	Horizontal Angular Resolution	Vertical Field of View	Vertical Angular Resolution
Pandar40P	200 m	$\pm 2 \sim \pm 5$ cm	360°	$0.2 \sim 0.4^\circ$	40°	$0.33 \sim 6^\circ$
Velodyne HDL_64E	120 m	± 2 cm	360°	$0.08 \sim 0.35^\circ$	26.9°	0.4°

It can be seen from Table 6 that Pandar40P has a longer measurement distance and a larger vertical field of view than Velodyne HDL-64E, but Velodyne HDL-64E has high measurement accuracy and high angular resolution. Table 7 shows the accuracy and time analysis of four different registration algorithms.

Table 7. Accuracy and time analysis of four different registration algorithms.

Algorithm	Pandar40P		Velodyne HDL-64E	
	Root Square Mean Error/m	Time/s	Root Square Mean Error/m	Time/s
NDT	0.1229 ± 0.03925	27.8158 ± 6.2704	0.1169 ± 0.03418	23.5018 ± 6.4974
NICP	0.1182 ± 0.04029	21.1194 ± 9.9502	0.09416 ± 0.04526	17.4721 ± 9.9084
TRICP	0.1244 ± 0.06505	17.8066 ± 10.8586	0.1008 ± 0.05135	10.5351 ± 6.7416
OURS	0.05996 ± 0.01751	0.5739 ± 0.1227	0.06971 ± 0.01597	0.5873 ± 0.1470

It can be seen from Table 7 that the algorithm proposed in this paper has a wide range of applicability and robustness, and can perform well on different types and specifications of lidar equipment. It can handle various noises and errors, and has high registration accuracy and reliability. At the same time, the algorithm can effectively overcome the differences between different lidars, and can also make an accurate registration of poor-quality or incomplete data.

6. Conclusions

The algorithm proposed in this paper is designed to address the challenges posed by the large scale, complex distribution, high noise, and strong sparsity of point cloud data collected by lidar during intelligent driving. The algorithm leverages regional feature extraction, mapping from local to global point clouds, and point cloud feature descriptors and local geometric features to achieve robust registration of point clouds for outdoor road scenes. The proposed algorithm has been verified through visualization effect analysis and experimental data analysis, demonstrating its high efficiency and accuracy in different working conditions and scenarios. The algorithm was also compared with three other point cloud registration algorithms under lidar with different metrological characteristics, and found to be robust and general. Finally, the algorithm has been validated on a real vehicle through point cloud registration experiments, showing its effectiveness and suitability for real-time applications. Overall, the algorithm proposed in this paper represents an important advancement in the field of point cloud registration and has significant potential for improving the efficiency and accuracy of intelligent driving systems.

Although the proposed algorithm has a more accurate registration performance, it still has some limitations. For example, the algorithm proposed in this paper only uses a single sensor to achieve point cloud registration. In our future work, it is intended to introduce a lightweight neural network that can fuse camera and lidar data to achieve precise estimation of lidar position and attitude. This will ensure the accuracy of point cloud registration even under extreme working conditions.

Author Contributions: Conceptualization, D.Y.; Methodology, D.Y., W.W. and S.L. (Sixuan Liu); Software, W.W., P.S. and S.L. (Sixuan Liu); Validation, W.D.; Formal analysis, W.D.; Investigation, D.Y. and S.L. (Shaohua Li); Data curation, P.S.; Writing—original draft, W.W.; Writing—review & editing, D.Y.; Project administration, S.L. (Shaohua Li); Funding acquisition, S.L. (Shaohua Li). All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by National Nature Science Foundation of China (Grant No. U22A20246, 11972238, in part by Key R & D Program Funded Project in Hebei Province under Grant 21342202D, and in part by Shijiazhuang Tiedao university Graduate Innovation Fund under Grant YC2023023).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used to support the findings of this study are included in the article.

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

References

1. Chen, Q.; Xie, Y.; Guo, S.; Bai, J.; Shu, Q. Sensing System of Environmental Perception Technologies for Driverless Vehicle: A Review of State of the Art and Challenges. *Sens. Actuators A Phys.* **2021**, *319*, 112566. [\[CrossRef\]](#)
2. Gao, H.; Zhang, X.; Zhao, J.; Li, D. Technology of intelligent driving radar perception based on driving brain. *CAAI Trans. Intell. Technol.* **2019**, *2*, 93–100. [\[CrossRef\]](#)
3. Nagy, B.; Benedek, C. Real-time point cloud alignment for vehicle localization in a high resolution 3d map. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops, Munich, Germany, 8–14 September 2018.
4. Besl, P.J.; McKay, N.D. A method for registration of 3-dshapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **1992**, *14*, 239–256. [\[CrossRef\]](#)
5. Wang, Q.; Zhang, J.; Liu, Y.; Zhang, X. Point cloud registration algorithm based on the combination of NDT and ICP. *Comput. Eng. Appl.* **2020**, *56*, 88–95.
6. Xue, S.; Zhang, Z.; Meng, X.; Lv, Q.; Tu, X. Point Cloud Registration Method for Pipeline Workpieces Based on RANSAC and Improved ICP Algorithms. *IOP Conf. Ser. Mater. Sci. Eng.* **2019**, *612*, 032190. [\[CrossRef\]](#)
7. Li, J.; Zhang, C.; Xu, Z.; Zhou, H.; Zhang, C. Iterative Distance-Aware Similarity Matrix Convolution with Mutual-Supervised Point Elimination for Efficient Point Cloud Registration. In Proceedings of the Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020.
8. Rosen, D.M.; Carlone, L.; Bandeira, A.S.; Leonard, J.J. Se-sync: A certifiably correct algorithm for synchronization over the special euclidean group. *Int. J. Robot. Res.* **2019**, *38*, 95–125. [\[CrossRef\]](#)
9. Maron, H.; Dym, N.; Kezurer, I.; Kovalsky, S.; Lipman, Y. Point registration via efficient convex relaxation. *ACM Trans. Graph. (TOG)* **2016**, *35*, 1–12. [\[CrossRef\]](#)
10. Yang, J.; Li, H.; Dylan, C.; Jia, Y. Go-ICP: A Globally Optimal Solution to 3D ICP Point-Set Registration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 2241–2254. [\[CrossRef\]](#)
11. Yang, J.; Cao, Z.; Zhang, Q. A fast and robust local descriptor for 3D point cloud registration. *Inf. Sci.* **2016**, *346–347*, 163–179. [\[CrossRef\]](#)
12. Lu, J.; Bin, W.; Xilu, F. Point cloud registration method based on SIFT feature points combined with ICP. *Laser Infrared* **2021**, *51*, 944–950.
13. Wang, W.; Tian, M.; Yu, J.; Song, C.; Li, J.; Zhou, M. Improved iterative nearest point cloud registration method. *Adv. Laser Optoelectron.* **2022**, *59*, 390–399.
14. Wang, Y.; Solomon, J.M. Deep closest point: Learning representations for point cloud registration. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3523–3532.
15. Wang, Y.; Solomon, J.M. Prnet: Self-supervised learning for partial-to-partial registration. *arXiv* **2019**, arXiv:1910.12240.
16. Qin, T.; Zhao, P.; Qin, P.; Zeng, J.; Chai, R.; Huang, Y. Point cloud registration algorithm based on residual attention mechanism. *Comput. Appl.* **2022**, *42*, 2184–2191.
17. Li, C.; Shi, H.; Li, Z. Point cloud registration method based on convolutional neural network combined with improved Harris-SIFT. *Adv. Laser Optoelectron.* **2020**, *57*, 238–247.
18. Li, J.; Yang, B.; Chen, C.; Habib, A. NRLL-UAV: Non-rigid registration of sequential raw laser scans and images for low-cost UAV LiDAR point cloud quality improvement. *ISPRS J. Photogramm. Remote Sens.* **2019**, *158*, 123–145. [\[CrossRef\]](#)
19. Kang, Z.; Chen, J.; Wang, B. Global registration of subway tunnel point clouds using an augmented extended Kalman filter and central-axis constraint. *PLoS ONE* **2015**, *10*, e0126862. [\[CrossRef\]](#) [\[PubMed\]](#)
20. Prokop, M.; Shaikh, S.A.; Kim, K.-S. Low Overlapping Point Cloud Registration Using Line Features Detection. *Remote Sens.* **2019**, *12*, 61. [\[CrossRef\]](#)
21. Lin, D.; Bannehr, L.; Ulrich, C.; Maas, H.-J. Evaluating Thermal Attribute Mapping Strategies for Oblique Airborne Photogrammetric System AOS-Tx8. *Remote Sens.* **2019**, *12*, 112. [\[CrossRef\]](#)
22. Jieun, B.; Junhyeok, P.; Seongjun, C.; Changwon, L. 3D Global Localization in the Underground Mine Environment Using Mobile LiDAR Mapping and Point Cloud Registration. *Sensors* **2022**, *22*, 2873.
23. Zhao, J.; Chu, J.; Feng, G.; Jiang, Z.; Shi, W.; Gao, Y. Multimodal Point Cloud Registration Based on Adaptive Feature Region in Radiotherapy Guidance. In Proceedings of the International Conference on Virtual Reality and Visualization, Recife, Brazil, 13–14 November 2020.
24. Gu, B.; Liu, J.; Xiong, H.; Li, T.; Pan, Y. ECPC-ICP: A 6D Vehicle Pose Estimation Method by Fusing the Roadside Lidar Point Cloud and Road Feature. *Sensors* **2021**, *21*, 3489. [\[CrossRef\]](#)
25. He, X.; Pan, S.; Gao, W.; Lu, X. LiDAR-Inertial-GNSS Fusion Positioning System in Urban Environment: Local Accurate Registration and Global Drift-Free. *Remote Sens.* **2022**, *14*, 2104. [\[CrossRef\]](#)
26. Xu, S.; Wang, G.; Wu, D. *Research on Internal Positioning Method of Pipeline Robot Based on Data Fusion and Point Cloud Registration*; Tsinghua University: Beijing, China, 2022.
27. Cheng, L.; Wu, Y.; Chen, S.; Zong, W.; Yuan, Y.; Sun, Y.; Zhuang, Q.; Li, M. A Symmetry-Based Method for LiDAR Point Registration. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 285–299. [\[CrossRef\]](#)
28. Lu, F.; Chen, G.; Liu, Y.; Zhang, L.; Qu, S.; Liu, S.; Gu, R. HRegNet: A Hierarchical Network for Large-Scale Outdoor LiDAR Point Cloud Registration. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 16014–16023.

29. Wang, W.; Zhao, C.; Zhang, H. PR-Alignment: Multidimensional Adaptive Registration Algorithm Based on Practical Application Scenarios. *Machines* **2023**, *11*, 254. [[CrossRef](#)]
30. Rusu, R.B.; Blodow, N.; Beetz, M. Fast Point Feature Histograms (FPFH) for 3D registration. In Proceedings of the IEEE International Conference on Robotics & Automation, Kobe, Japan, 12–17 May 2009.
31. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? the kitti vision benchmark suite. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3354–3361.
32. Rusu, R.B.; Blodow, N.; Marton, Z.C.; Beetz, M. Aligning point cloud views using persistent feature histograms. In Proceedings of the IEEE/Rsj International Conference on Intelligent Robots and Systems, Nice, France, 22–26 September 2008; pp. 3384–3391.
33. Eggert, D.W.; Lorusso, A.; Fisher, R.B. Estimating 3-D rigid body transformations: A comparison of four major algorithms. *Mach. Vis. Appl.* **1997**, *9*, 272–290. [[CrossRef](#)]
34. Chen, Y.; Medioni, G. Object Modeling by Registration of Multiple Range Images. *Int. J. Image Vis. Comput.* **1992**, *10*, 145–155. [[CrossRef](#)]

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