

## Article

# An Enhanced LiDAR-Based SLAM Framework: Improving NDT Odometry with Efficient Feature Extraction and Loop Closure Detection

Yan Ren <sup>1</sup> , Zhendong Shen <sup>1,\*</sup>, Wanquan Liu <sup>2</sup>  and Xinyu Chen <sup>1</sup>

<sup>1</sup> School of Artificial Intelligence, Shenyang Aerospace University, Shenyang 110136, China; 20113269@sau.edu.cn (Y.R.); 20140011@sau.edu.cn (X.C.)

<sup>2</sup> School of Intelligence System Engineering, Sun Yat-sen University, Shenzhen 518107, China; liuwq63@mail.sysu.edu.cn

\* Correspondence: shenzhendong5311@gmail.com

**Abstract:** Simultaneous localization and mapping (SLAM) is crucial for autonomous driving, drone navigation, and robot localization, relying on efficient point cloud registration and loop closure detection. Traditional Normal Distributions Transform (NDT) odometry frameworks provide robust solutions but struggle with real-time performance due to the high computational complexity of processing large-scale point clouds. This paper introduces an improved NDT-based LiDAR odometry framework to address these challenges. The proposed method enhances computational efficiency and registration accuracy by introducing a unified feature point cloud framework that integrates planar and edge features, enabling more accurate and efficient inter-frame matching. To further improve loop closure detection, a parallel hybrid approach combining Radius Search and Scan Context is developed, which significantly enhances robustness and accuracy. Additionally, feature-based point cloud registration is seamlessly integrated with full cloud mapping in global optimization, ensuring high-precision pose estimation and detailed environmental reconstruction. Experiments on both public datasets and real-world environments validate the effectiveness of the proposed framework. Compared with traditional NDT, our method achieves trajectory estimation accuracy increases of 35.59% and over 35%, respectively, with and without loop detection. The average registration time is reduced by 66.7%, memory usage is decreased by 23.16%, and CPU usage drops by 19.25%. These results surpass those of existing SLAM systems, such as LOAM. The proposed method demonstrates superior robustness, enabling reliable pose estimation and map construction in dynamic, complex settings.



Academic Editor: Vijaya Raghavan

Received: 30 December 2024

Revised: 13 January 2025

Accepted: 16 January 2025

Published: 19 January 2025

**Citation:** Ren, Y.; Shen, Z.; Liu, W.; Chen, X. An Enhanced LiDAR-Based SLAM Framework: Improving NDT Odometry with Efficient Feature Extraction and Loop Closure Detection. *Processes* **2025**, *13*, 272. <https://doi.org/10.3390/pr13010272>

**Copyright:** © 2025 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/>).

**Keywords:** LiDAR odometry; simultaneous localization and mapping (SLAM); normal distributions transform (NDT); feature extraction; loop closure detection

## 1. Introduction

In contemporary domains such as autonomous driving, drone navigation, and mobile robot localization, Light Detection and Ranging (LiDAR) has become an indispensable sensor. Its popularity stems from its high precision, robustness to environmental variations, and ability to generate dense point cloud data, making it indispensable in complex and dynamic scenarios while facilitating real-time simultaneous localization and mapping (SLAM).

The central objective of SLAM is to continuously process sensor data in real-time to construct a reliable map of the environment while simultaneously estimating the robot's

pose [1]. This dual capability enables precise navigation and localization, which are essential for autonomous operation across various applications. However, achieving high accuracy and efficiency in SLAM systems is far from trivial and is primarily influenced by two key components: point cloud registration and loop closure detection. These components not only determine the fidelity of the constructed map but also impact the computational performance and overall robustness of the SLAM pipeline [2,3].

Point cloud registration aims to align multiple point clouds to reconstruct the environment within a unified coordinate system. Given that real-world environments often feature dynamic objects and noise, registration algorithms must achieve both high robustness and accuracy [4,5]. The Iterative Closest Point (ICP) algorithm, one of the most widely used methods for point cloud registration, operates by iteratively minimizing the distance between the closest points of two point clouds. However, ICP faces significant challenges, including slow convergence, a tendency to settle in local optima, and a strong dependency on the initial pose, making it less suitable for high-efficiency, high-precision applications [6,7]. An alternative approach, the Normal Distributions Transform (NDT) method, models point clouds using Gaussian distributions, and demonstrates greater robustness in noisy environments [8,9]. Nevertheless, the computational complexity of NDT is substantial, as it requires modeling the entire point cloud, which proves inefficient in real-time applications with limited computational resources, such as autonomous ground vehicles and drones [10,11].

To alleviate the computational burden, many researchers have proposed feature extraction-based strategies, simplifying the point cloud registration process by focusing on local features of the environment. Algorithms such as LOAM and LeGO-LOAM have successfully applied this approach, reducing computational redundancy while maintaining performance [12,13]. However, these feature extraction methods still face limitations in terms of both accuracy and efficiency, particularly when handling large-scale and complex environments, such as expansive outdoor settings [14,15].

Regarding loop closure detection, traditional SLAM systems typically rely on odometry information, using the Euclidean distance between odometry estimates to detect potential loop closures. However, as odometry errors accumulate over time, they can introduce drift, which degrades the accuracy of loop closure detection [16]. To mitigate this drift, geometric-based loop closure detection methods—such as Scan Context—have been incorporated into SLAM systems, improving detection accuracy, particularly in large-scale mapping scenarios [17,18].

To address the aforementioned challenges and improve the efficiency of NDT in large-scale point cloud registration, as well as to enhance the accuracy of loop closure detection, this paper presents an enhanced NDT odometry framework. The key contributions of this work are as follows:

- We propose a point cloud preprocessing and feature-based matching method that unifies extracted planar features and edge features into a single cohesive feature point cloud. This integrated approach streamlines the matching process, significantly reduces computational complexity, and enhances both efficiency and alignment precision, surpassing traditional methods that process features separately.
- We introduce an innovative enhancement to the loop closure detection module by replacing the traditional Radius Search (RS) method with a novel parallel approach that integrates Radius Search and Scan Context (SC). This dual-loop closure detection strategy not only improves the robustness of loop closure identification but also significantly boosts efficiency and accuracy, addressing limitations of conventional single-method approaches.

- We adopt a novel approach that integrates feature-based point cloud registration with full cloud mapping during global optimization, ensuring that the final map not only achieves high precision but also retains comprehensive and detailed environmental information, surpassing the capabilities of conventional methods.

Finally, extensive experimental validation is conducted to evaluate the proposed method. The results show that, compared to leading SLAM frameworks such as LOAM, the proposed method demonstrates superior performance in terms of registration efficiency, loop closure detection accuracy, and real-time capability. In particular, the proposed approach outperforms traditional methods in specific challenging scenarios.

## 2. Related Works

In simultaneous localization and mapping (SLAM) technology, point cloud registration and loop closure detection are pivotal factors that influence the overall performance of the system. To enhance the efficiency and accuracy of SLAM systems, researchers have made significant contributions in areas such as point cloud registration algorithms, computational efficiency optimization, and loop closure detection accuracy.

### 2.1. Advancements in Point Cloud Registration Algorithms

Point cloud registration is a fundamental and critical component of LiDAR-based SLAM. The inherent complexity of real-world environments, coupled with the physical limitations of LiDAR sensors, makes it impossible to capture a complete scene in a single scan. As a result, aligning point cloud data collected at different time intervals is essential for constructing an accurate environmental model. Traditional approaches to point cloud registration include the Iterative Closest Point (ICP) algorithm, Normal Distributions Transform (NDT), and modern deep learning-based techniques.

The Iterative Closest Point (ICP) algorithm aligns two point clouds by iteratively minimizing the distance between corresponding points. It adjusts their transformation until the error reaches a predefined threshold or the maximum number of iterations is achieved [19]. This method efficiently handles small, well-structured scenes, as it does not require point cloud segmentation. However, ICP encounters significant challenges related to computational complexity and susceptibility to local minima, particularly when the initial pose estimate is insufficiently accurate [6].

An alternative approach, the Normal Distributions Transform (NDT) algorithm, employs probabilistic techniques by discretizing the point cloud into a grid and modeling its distribution as a Gaussian. Based on this representation, NDT calculates the transformation between the target and incoming point clouds. Compared to the Iterative Closest Point (ICP) algorithm, NDT delivers enhanced computational efficiency. However, its performance is still limited when handling large-scale, dense point clouds, especially in real-time applications such as autonomous vehicles or drones, where computational resources are constrained [10,11].

Recently, deep learning-based point cloud registration methods have garnered significant attention, with notable approaches including PointNetLK [20], Deep ICP [21], and PRNet [22]. These methods utilize deep learning models to extract features from point clouds, enabling faster computations and improved matching accuracy. However, they remain in the early stages of development and continue to face challenges concerning precision and robustness [16].

### 2.2. Optimization of Feature Extraction Methods

To accelerate point cloud registration and reduce computational complexity, feature-based registration methods have been developed. These methods extract key geometric

features from the environment, minimizing redundant calculations and mitigating interference from irrelevant points. A classic example is the LOAM (LiDAR Odometry and Mapping) algorithm, which segments the point cloud into planar and edge features based on point curvature, subsequently using these features for coordinate transformation [12].

Several LOAM variants have been introduced, such as LeGO-LOAM and Loam Livox. LeGO-LOAM optimizes ground point separation to cater to environments with limited computational capabilities, such as those involving Automated Guided Vehicles (AGVs) [13]. In contrast, Loam Livox is tailored specifically for solid-state LiDAR, refining point cloud filtering strategies to accommodate the unique scanning patterns of these devices [23].

These feature extraction approaches significantly enhance the registration efficiency and reduce computational resource demands. However, a critical challenge persists: how to further improve computational efficiency without compromising accuracy, particularly in complex and dynamic environments.

### 2.3. Enhancements in Loop Closure Detection Techniques

Loop closure detection is a critical component of SLAM systems, aimed at identifying loops in the robot's trajectory and correcting accumulated errors. Common loop closure detection methods are generally categorized into descriptor-based and deep learning-based approaches.

A widely adopted descriptor-based approach is the Scan Context method, introduced by KAIST in 2018 [17]. This method is computationally efficient and achieves high accuracy in loop closure detection with relatively low overhead, making it particularly suitable for resource-constrained environments. Scan Context divides the point cloud into azimuthal and radial bins, computing the maximum Z value in each bin to create a descriptor. This reduces computational complexity while improving detection accuracy. Various SLAM frameworks, such as SC-LeGO-LOAM [18] and SC-LIO-SAM, have integrated Scan Context. However, the method is susceptible to false loop closures under certain conditions [24].

Deep learning-based methods, such as OverlapNet [25] and OverlapTransformer [26], have also gained traction. These techniques generally offer higher accuracy and better adaptability to diverse environments. However, they come with high training costs and limited generalization, requiring fine-tuning for different LiDAR models and sensor configurations.

In addition to these advanced methods, traditional Euclidean distance-based loop closure detection approaches remain widely used. While effective in simpler environments, these methods are highly dependent on odometry accuracy and are vulnerable to failure due to accumulated errors, especially during extended operations [27].

## 3. Methodology

The overview of our proposed method is illustrated in Figure 1. The overall framework consists of four main components: point cloud preprocessing, LiDAR odometry, loop closure detection, and global optimization.

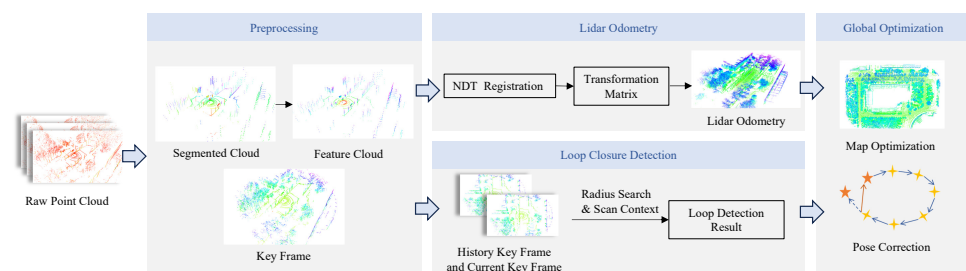


Figure 1. The system structure.

### 3.1. Point Cloud Preprocessing

Point cloud preprocessing focuses on minimizing redundancy in raw laser scan data and alleviating the computational burden during point cloud registration. By doing so, it enhances both the precision and robustness of the registration process. The preprocessing methodology proposed in this section is inspired by the approach outlined in [13], which optimizes point cloud segmentation and feature extraction to preserve the essential features crucial for accurate matching and transformation calculations in subsequent stages.

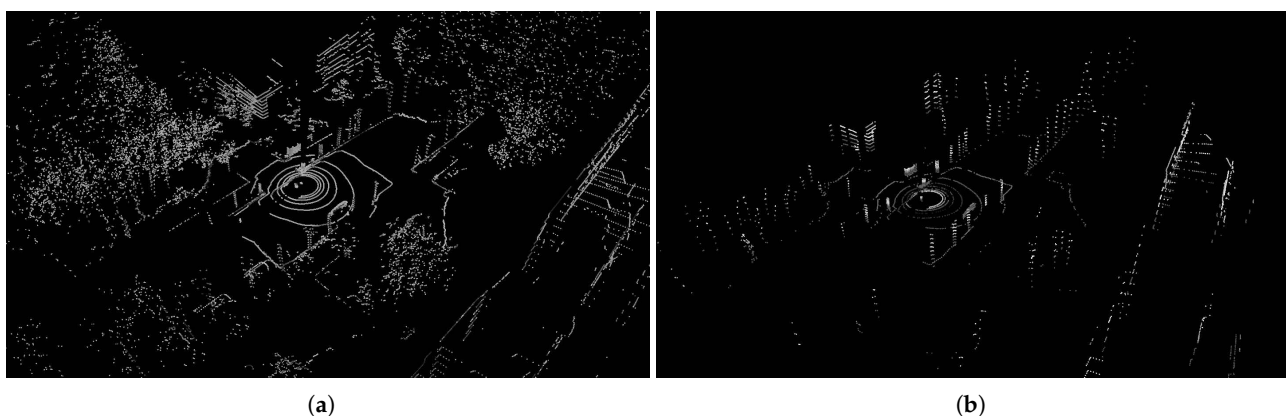
The segmentation step removes environmental noise and preserves key information, while the feature extraction process reduces the number of points representing the environment, thus lowering computational complexity. This results in improved system stability and overall performance by ensuring that only the most relevant data are used in the later stages of the system. Let the point cloud captured by the laser at time  $t$  be represented as  $P_t = \{p_i | p_i \in \mathbb{R}^3, i = 1, 2, \dots, M\}$ , where  $p_i$  is the  $i$ -th point in  $P_t$ , and  $M$  is the total number of points in the frame at time  $t$ . The point cloud is segmented into  $N$  scan lines based on the vertical angles.

Prior to segmentation, the ground points are extracted and labeled as  $P_{ground}$ , and these points are excluded from the subsequent segmentation process. The remaining non-ground points,  $P_{non-ground}$ , are used for object segmentation, where the points are clustered, each assigned a specific category. Furthermore, outliers and small point clouds are filtered out, ensuring that only the point clouds corresponding to larger objects are retained.

Next, feature extraction is performed on the non-ground points. Let  $S$  be the set of neighboring points of  $p_i$  within the same row. The smoothness  $c_i$  of point  $p_i$  is calculated via Equation (1):

$$c_i = \frac{1}{|S| \cdot \|r_i\|} \left\| \sum_{j \in S, j \neq i} (r_j - r_i) \right\| \quad (1)$$

where  $r_i$  is the position vector of  $p_i$ , and  $r_j$  is the position vector of a neighboring point in  $S$ . And the points  $p_i$  in  $P_t$  with smoothness  $c_i$  greater than a given threshold  $c$  are categorized as edge points, while those with smoothness  $c_i$  smaller than  $c$  are classified as planar points. After sorting the smoothness values, the planar and edge features are combined with the ground points to form the required feature point cloud as shown in Figure 2.



**Figure 2.** Combined feature point cloud. (a) is the raw point cloud acquired by LiDAR, and (b) is the feature point cloud. The feature point cloud is composed of planar points, edge points, and ground points; the outlier points and small-scale points in the environment are removed; and only large-scale point clouds are retained. Compared to the original point cloud, the feature point cloud significantly reduces the number of points while effectively preserving environmental features.

Unlike traditional downsampling methods that indiscriminately reduce the number of points in the cloud, this feature extraction approach retains the most critical features while



reducing the point count, leading to lower computational cost and more robust matching during the registration process. This is achieved by first categorizing points based on their smoothness values. The resulting feature point cloud ensures both efficiency and accuracy.

### 3.2. LiDAR Odometry

After preprocessing the point cloud, the feature point cloud of the environment is utilized to compute the robot's relative motion and position within the environment. This process is performed using the Normal Distributions Transform (NDT). NDT is a probabilistic method, initially introduced for 2D scan matching in [8] and later extended to 3D point cloud data by Magnusson et al. [28]. The core principle of NDT involves discretizing the point cloud into a grid and modeling the distribution of points within each cell as a Gaussian distribution. This approach reformulates point cloud registration as a statistical optimization problem.

During this process, the environmental feature point cloud is divided into multiple grid cells, with each cell containing a set of points. The points within each cell are modeled using a Gaussian distribution, providing a localized probabilistic representation. This approach aims to capture the statistical characteristics of the point cloud distribution within each grid cell. By employing this modeling method, point cloud registration is transformed from a point-to-point matching problem into a distribution-based matching problem.

The Gaussian distribution for points within each grid cell is expressed by the following probability density function:

$$f(x) = \frac{1}{(2\pi)^{D/2} \sqrt{|\Sigma|}} \exp\left(-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu)\right) \quad (2)$$

where,  $x$  represents a specific point within the point cloud that is being evaluated to determine its probability of occurrence in a grid cell, based on the Gaussian distribution.  $D$  denotes the dimensionality of the point cloud (e.g.,  $D = 3$  for 3D data). Formula (2) reflects the probability that the point  $x$  occurs within a particular grid cell.  $\Sigma$  is the covariance matrix, and  $|\Sigma|$  is its determinant.  $\mu$  is the mean (center) of the points in the grid cell. By leveraging this probability distribution, the point cloud in each grid cell is transformed into a continuous mathematical representation.

In the point cloud registration process, the NDT method seeks to determine the transformation  $T$  that aligns the source point cloud. This is achieved by maximizing the likelihood that the points in the transformed source point cloud match the distribution of the target point cloud. The transformation is defined as

$$F(\xi) = \sum_{i=1}^n f(T(\xi, p_i)) \quad (3)$$

where  $\xi = [t_x, t_y, t_z, \varphi_x, \varphi_y, \varphi_z]$  represents the transformation parameters (both translation and rotation).  $n$  is the number of points in the source point cloud.  $p_i$  is the  $i$ -th point in the source point cloud,  $T(\xi, p_i)$  represents the position of  $p_i$  after being transformed by the rigid body transformation parameters  $\xi$ , and  $f(T(\xi, p_i))$  is the probability value of  $p_i$  under the probability distribution of the target point cloud. The transformation expression of  $T(\xi, p_i)$  is

$$T(\xi, p_i) = R_{\mathbf{t}} p_i + \mathbf{t} \quad (4)$$

where  $R_{\mathbf{t}}$  is the rotation matrix derived from the rotation components  $[\varphi_x, \varphi_y, \varphi_z]$ , and  $\mathbf{t}$  is the translation vector between the target point and source point.

After completing the point cloud registration, the core task of the LiDAR odometry is to generate the pose trajectory of the robot in three-dimensional space by calculating the

pose changes between consecutive frames. Using the registration results obtained by NDT, the objective function  $F(\zeta)$  is maximized.

For each frame  $f$ , the transformation matrix  $T_f$  is computed, which includes the rotation matrix  $R_{t_f}$  and the translation vector  $t_f$  between two consecutive point clouds:

$$T_f = \begin{bmatrix} R_{t_f} & t_f \\ 0 & 1 \end{bmatrix} \quad (5)$$

The global pose trajectory is generated by iteratively updating the global pose of the robot. Assuming the global pose of the previous frame is  $T_{f-1}^{\text{global}}$ , the global pose of the current frame  $T_f^{\text{global}}$  is calculated as

$$T_f^{\text{global}} = T_{f-1}^{\text{global}} \cdot T_f \quad (6)$$

This iterative update of the global pose matrix accumulates the transformations of the robot's position and orientation from the initial frame to the current frame, effectively generating the robot's trajectory in real time.

### 3.3. Loop Closure Detection

Loop closure detection is a crucial component of SLAM systems, designed to identify revisited locations in the robot's trajectory and mitigate cumulative errors caused by odometry drift. Traditional methods for loop closure detection often rely on either global feature matching or local geometric alignment, each with its inherent limitations. Global matching methods may lack precision in densely environments, while local methods often fail to detect loops when significant odometric drift occurs.

To address these challenges, this approach combines Radius Search and Scan Context, leveraging their complementary strengths to achieve a balance between computational efficiency and accuracy. Radius Search uses the KD-tree [29] to search for historical key frames near the current key frame according to the specified radius threshold and serves as the target point cloud for ICP registration. Scan Context converts the current frame point cloud information to polar coordinates for calculation and obtains the ring value [17]. Search under KD-tree by descriptor to complete the loop detection.

In this process, we use  $C_{RS}$  and  $C_{SC}$  to represent the loop closure detection results of Radius Search and Scan Context, respectively. Here,  $C_{RS} \in \{0, 1\}$  is the confidence score of Radius Search, indicating the reliability of the geometrically closest match.  $C_{SC} \in \{0, 1\}$  is the confidence score of Scan Context, representing the global descriptor similarity. If Radius Search successfully detects a loop closure, i.e.,  $C_{RS} = 1$ , the transformation matrix  $T_{k,i}^{RS}$  obtained from Radius Search is used to update the pose. Conversely, if Radius Search fails to detect a loop closure, i.e.,  $C_{RS} = 0$ , but Scan Context detects one ( $C_{SC} = 1$ ), the transformation matrix  $T_{k,i}^{SC}$  derived from Scan Context is applied to correct the pose.

The updated pose  $T_k^{\text{new}}$  of the current frame  $k$  is a weighted combination of the transformations provided by the two methods:

$$T_k^{\text{new}} = T_k \cdot (C_{RS} \cdot T_{k,i}^{RS} + (1 - C_{RS}) \cdot C_{SC} \cdot T_{k,i}^{SC}) \quad (7)$$

where  $T_k$  is the pose of the current frame before the transformation, and  $T_{k,i}^{RS}$  and  $T_{k,i}^{SC}$  are the pose correction relations obtained by the two methods, respectively. The confidence weighting mechanism ensures that the final pose correction prioritizes the method with higher reliability for the given scenario.

Although Scan Context excels in global matching, its performance can degrade in complex or highly dynamic environments, where the global descriptors may be ambiguous

or not distinct enough. Similarly, Radius Search is effective for local refinement but relies on spatial proximity, which can fail when the robot's trajectory has experienced significant drift. To address these limitations, the system combines the global perspective of Scan Context with the local precision of Radius Search, ensuring that the loop closure detection is both robust and accurate in environments with complex geometries or substantial odometric drift. First, the Radius Search method is used to loopback the current location to find the closest historical key frame, and ICP is used for point cloud registration. If the registration is successful and the registration quality meets the conditions, the corrected pose is added to the graph optimization. If Radius Search fails to match the loop, it indicates that the odometer has shifted significantly, causing the Radius Search to fail. Then, we use the Scan Context method to search globally until potential loop keyframes are found, and ICP is used for registration.

This integrated approach significantly outperforms traditional methods by addressing the limitations of both global and local matching techniques. It ensures reliable and accurate loop closure detection, particularly in environments with complex geometries or significant odometric drift, thereby improving the overall robustness and performance of the SLAM system.

### 3.4. Global Optimization

To ensure the global consistency of the point cloud map and enhance the accuracy of trajectory estimation, this project performs systematic map and pose optimization after completing point cloud registration and loop closure detection. The optimization process involves two key steps: the incorporation of loop closure constraints, and backend graph optimization. These steps collectively update the map and correct the trajectory, resulting in a globally consistent representation.

During the loop closure detection phase, when a loop is successfully detected, the relative pose between the current frame and a historical frame is computed using the registration algorithm. This relative pose is then introduced as a new constraint into the backend optimization process. These constraints effectively compensate for the impact of accumulated errors on the consistency of the trajectory and map, significantly improving the global accuracy of the system, especially during long-duration operations.

In backend optimization, the map and trajectory are modeled as a factor graph, where the poses of point cloud frames act as vertices, and the relative transformations between frames, along with the constraints generated by loop closure detection, serve as edges. During the point cloud mapping process, feature point clouds (including edge and plane points) are used for inter-frame registration. Compared to using the full point cloud directly, the sparsity of feature point clouds significantly reduces the computational complexity of the registration process while maintaining high matching accuracy. Once the inter-frame registration is completed, the full point cloud is projected into the global coordinate system using the pose transformation matrix derived from the registration. This approach effectively combines the computational efficiency of feature point clouds with the rich detail provided by the full point cloud, resulting in a map that is both accurate and visually detailed.

The full point cloud map  $P_{map}$ , is generated by applying the pose transformation matrices  $T_f$  for each frame  $f$  as follows:

$$P_{map} = \{T_f P_f \mid f = 1, 2, \dots, N\} \quad (8)$$

where  $P_f$  represents the point cloud of frame  $f$ .

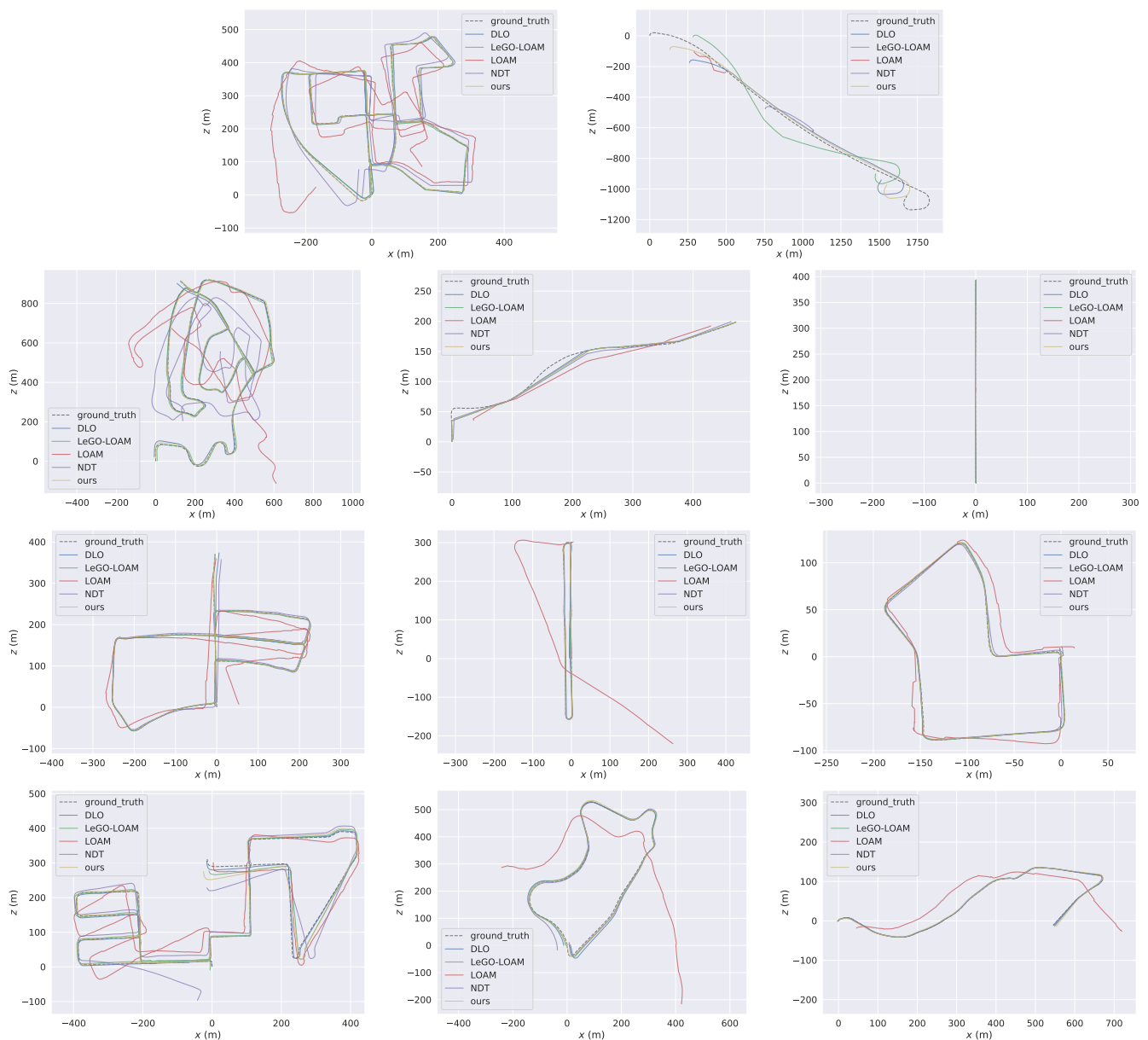
The core objective of the optimization is to generate a globally consistent map and trajectory by minimizing the weighted squared error of all pose constraints. Upon completion



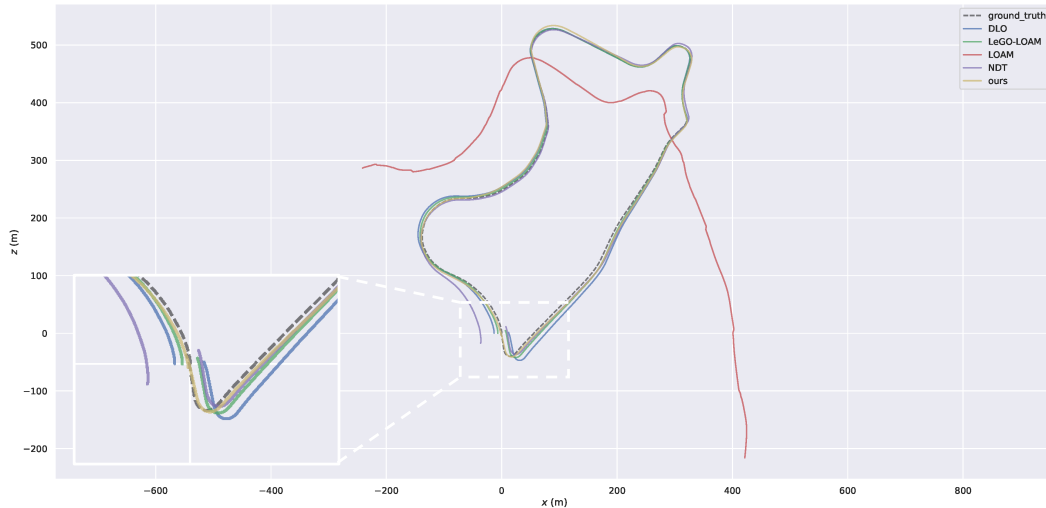


8 GB of RAM, running Ubuntu 18.04 and ROS Melodic [32] to ensure consistency in the experimental results.

Figure 4 provides a visual comparison of the trajectories estimated by different algorithms across various environmental datasets. As shown in Figure 4, our improved method estimates trajectories that closely match the ground truth in most scenes. In the highway scene of Sequence 01, high vehicle speed and sparse feature points lead to frame matching failures and frame shrinkage. As a result, the performance in this scenario is poorer. The experimental results show that none of the selected methods fully reconstruct the trajectory and map for this dataset. Notably, in more complex environments (e.g., loop closure scenarios), our improved method more accurately constructs the corresponding maps and trajectories compared to LeGO-LOAM and DLO. Especially in loop closure detection, the improved algorithm that combines Radius Search and Scan Context strategies shows higher accuracy, which is particularly evident in Sequence 09. Figure 5 illustrates the loop closure detection results for this scenario.



**Figure 4.** Comparison of trajectories across different algorithm frameworks for Sequence 00-10. The trajectories generated during mapping for LOAM, LeGO-LOAM, DLO, the original NDT, and our method are compared.



**Figure 5.** Loop closure detection results for various methods on Sequence 09. It can be seen that our improved method effectively identifies the loop closure. The parallel strategy using two loop closure detection methods greatly improves detection accuracy.

Furthermore, Figure 4 highlights the accuracy limitations of the pre-improved NDT method. Its inter-frame registration relies on the full point cloud, resulting in longer computation times and potential frame skipping at high frequencies, which in turn reduces odometry accuracy. When large odometry drift occurs, loop closure cannot be detected in time, which affects the overall mapping performance. To improve computational efficiency, the original NDT method requires point cloud downsampling, but this process loses important features during significant downsampling, negatively impacting the accuracy of the final map and trajectory.

In addition to visualizing and comparing the trajectories, we also computed the Absolute Pose Error (APE) and Relative Pose Error (RPE) for each method across various environmental datasets. APE measures the absolute position and orientation error between the estimated trajectory and the ground truth trajectory, while RPE calculates the relative position and orientation error between consecutive frames, and they are defined as

$$\text{APE}(t) = \|\mathbf{T}_{est}(t) - \mathbf{T}_{gt}(t)\| \quad (9)$$

$$\text{RPE}(t_1, t_2) = \left\| (\mathbf{T}_{est}(t_2) \cdot \mathbf{T}_{est}(t_1)^{-1}) - (\mathbf{T}_{gt}(t_2) \cdot \mathbf{T}_{gt}(t_1)^{-1}) \right\| \quad (10)$$

where  $\mathbf{T}_{est}(t)$  is the estimated pose, and  $\mathbf{T}_{gt}(t)$  is the ground truth pose at time  $t$ .  $\mathbf{T}_{est}(t_1)$  and  $\mathbf{T}_{gt}(t_1)$  represent the estimated and ground truth poses at time  $t_1$ , and RPE measures the relative error between these two time points.

The experimental results are presented in Tables 1 and 2. In the experiment, we evaluated our method under two configurations: with and without loop detection, referred to as “ours-looped” and “ours-no loop”, respectively, for comparison with other methods. For configurations other than “ours-looped”, results marked with “-” indicate failure in the experiment, where data could not be obtained. Our method demonstrates strong performance in terms of APE, with significant accuracy improvements in environments featuring loop closures. In loop-free environments, our method achieves a 35.59% improvement over the original NDT method. In environments with loop closures, the accuracy improvement rises significantly to 86.95%, highlighting a clear advantage. It is worth noting that the improvement in RPE is not as pronounced as in APE. This is because RPE primarily reflects the relative motion between consecutive frames, where error accumulation is less significant over shorter time scales, placing greater emphasis on local motion accuracy.

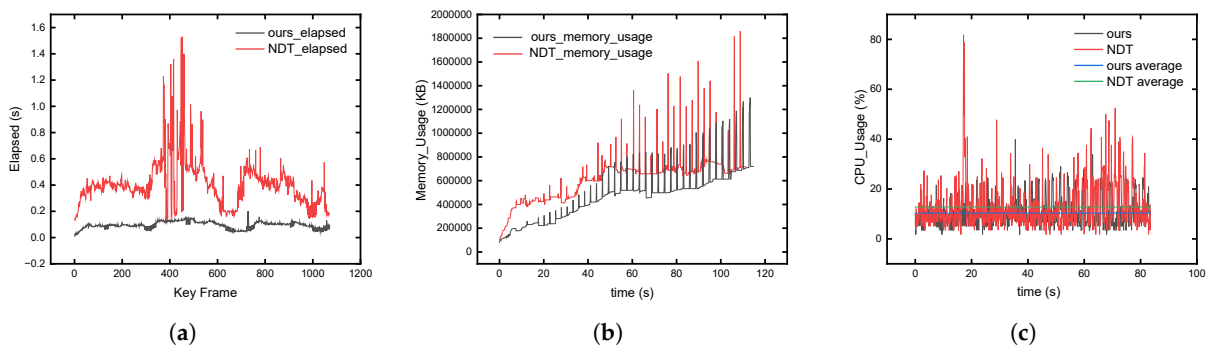
**Table 1.** Comparisons on APE.

	LOAM	LeGO-LOAM	DLO	NDT	Ours-No Loop	Ours-Looped
Sequence 00	59.509626	3.198303	5.188796	18.686969	2.105438	<b>0.860064</b>
Sequence 01	-	209.764716	165.749755	-	<b>110.493611</b>	-
Sequence 02	241.090825	8.906158	<b>6.97824</b>	148.489258	11.670094	8.902045
Sequence 03	25.825729	0.726263	0.783867	5.12398	<b>0.564025</b>	-
Sequence 04	-	0.378594	0.518841	0.805754	<b>0.283403</b>	-
Sequence 05	24.834642	2.295983	3.78769	4.766309	2.622927	<b>1.100858</b>
Sequence 06	184.256251	2.155455	2.543412	0.702749	1.277034	<b>0.662273</b>
Sequence 07	11.086145	1.219031	1.033187	1.074191	0.858492	<b>0.587579</b>
Sequence 08	32.752558	12.314772	<b>8.574063</b>	24.35511	9.634456	8.787474
Sequence 09	163.527914	<b>3.93364</b>	7.159924	10.382833	10.319081	6.294579
Sequence 10	58.84271	<b>1.552112</b>	2.423999	2.155243	3.585033	-
<b>Average</b>	89.08071111	22.40409336	20.3708587	21.6542396	13.94669036	<b>3.884981714</b>

**Table 2.** Comparisons on RPE.

	LOAM	LeGO-LOAM	DLO	NDT	Ours-No Loop	Ours-Looped
Sequence 00	2.226906	4.465059	<b>0.975846</b>	1.003241	1.079489	1.357211
Sequence 01	-	11.09831	<b>3.370803</b>	-	3.459235	-
Sequence 02	5.831719	6.252685	1.146399	1.194927	<b>1.145108</b>	1.547784
Sequence 03	16.843572	16.817886	3.944846	<b>3.209778</b>	3.549879	-
Sequence 04	-	8.375045	0.060854	0.111419	<b>0.052909</b>	-
Sequence 05	2.203399	4.515387	0.885576	<b>0.881588</b>	1.477781	1.881688
Sequence 06	7.107527	6.308099	<b>0.889477</b>	0.904039	0.912664	1.622717
Sequence 07	1.725298	3.451616	<b>0.756097</b>	0.765544	0.788823	1.039604
Sequence 08	2.323839	4.669723	<b>0.944999</b>	0.960861	0.949687	1.165852
Sequence 09	6.100783	6.275799	1.351646	<b>1.346625</b>	1.415438	1.517305
Sequence 10	4.66399	4.687313	1.077134	1.025371	<b>1.023478</b>	-

Additionally, we evaluated the improvement in system performance due to the feature-based inter-frame registration. Using the KITTI Sequence 07 dataset as an example, we recorded and assessed the entire mapping process before and after the improvement in terms of registration time, memory usage, and CPU usage. The results are shown in Figure 6a–c. It can be observed that the feature-based inter-frame registration greatly enhances registration efficiency. The average registration time was reduced by 66.7%, from 0.45 s to 0.15 s, memory usage decreased by 23.16%, from 608,663.7 KB to 467,672.8 KB, and CPU usage dropped by 19.25%, from 12.82% to 10.35%.



**Figure 6.** (a–c) Inter-frame registration time, memory usage, and CPU usage before and after the improvement. Our improved method effectively reduces matching time and computational load.

These results clearly demonstrate that the proposed improvements offer significant enhancements in accuracy, efficiency, and resource usage compared to traditional methods, providing a precise and efficient SLAM solution. In most scenarios, our method performs at or even exceeds the capabilities of existing mainstream methods.

#### 4.2. Real-World Experiments

This section evaluates the performance of the improved NDT framework in real-world mapping and trajectory estimation. The experimental setup involved a differential-drive mobile robot equipped with a laptop featuring an Intel i7-10750H processor and 16 GB RAM, running the Ubuntu 18.04 operating system with the ROS Melodic framework. A Hesai Pandar XT-16 mechanical 16-line LiDAR was used for environmental scanning. The robot platform and sensor configurations are shown in Figure 7. All experiments were conducted with consistent parameter settings to ensure fairness.



**Figure 7.** Mobile robot platform.

To evaluate the algorithm's adaptability and robustness, we selected the following typical scenarios for testing. These scenarios encompass different environmental characteristics to assess the improved NDT framework's performance under diverse conditions:

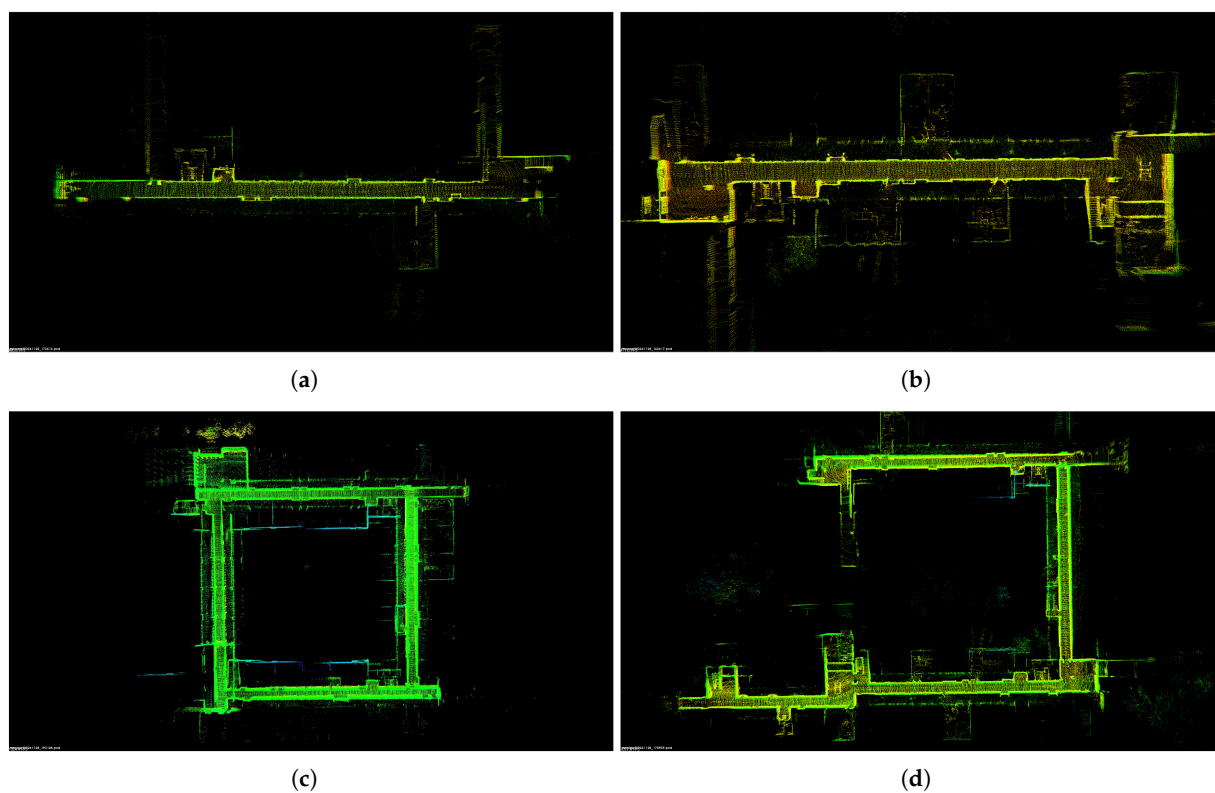
- **Indoor Environments:** This includes one-way corridors, round-trip corridors, loop corridors, and a long corridor with minimal features. Corridor scenarios feature relatively simple structures with regular walls and doorways and minimal dynamic interference, making them ideal for evaluating mapping accuracy and efficiency in static, regular environments. The long corridor scenario, with sparse features and a high degree of similarity, tests the algorithm's ability to handle environments with highly similar features.
- **Outdoor Environments:** The testing was conducted on a university campus, covering complex open spaces and building perimeters with dynamic objects such as pedestrians and vegetation. These environments enabled the assessment of the algorithm's robustness in dynamic and uncertain conditions, with a focus on loop closure detection, feature extraction, and map optimization strategies.

##### 4.2.1. Indoor Environments

In the indoor scenarios, the robot platform was remotely controlled to move through the four environments mentioned above, recording data for analysis. Maps generated using the improved NDT framework are shown in Figure 8. The results demonstrate that

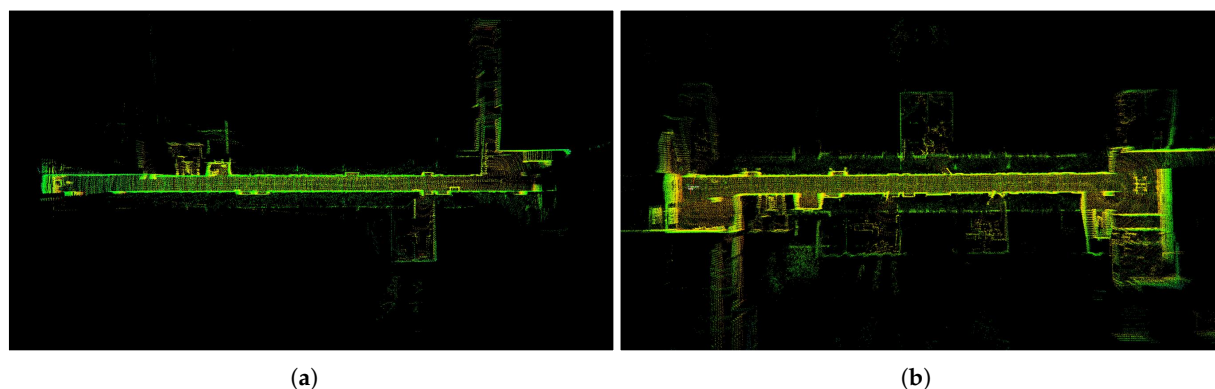


the proposed method performs excellently, accurately capturing details such as walls and doorways to create clear, complete maps. In round-trip and loop corridors, the improved algorithm effectively detected loops and performed precise registration and optimization without any artifacts or alignment errors.



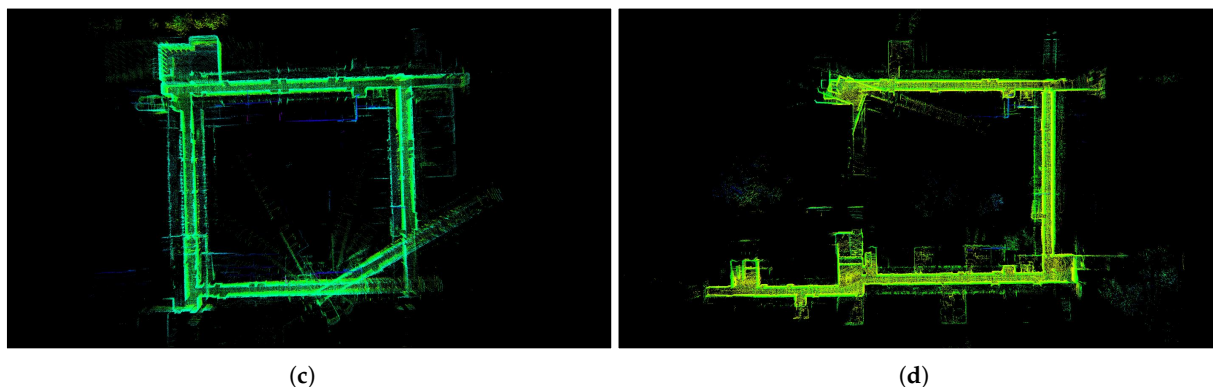
**Figure 8.** Maps generated using the improved method. (a–d) The one-way corridor, round-trip corridor, loop corridor, and long, feature-sparse corridor, respectively.

In contrast, maps generated using the original NDT framework under the same conditions are shown in Figure 9. While the original method could generally complete mapping in simple scenarios like one-way corridors, it failed in larger-scale environments like loop corridors, where loop closure detection errors and redundant mapping caused significant drift. This issue primarily stems from the original method's reliance on complete point clouds for registration, leading to slow computation that cannot keep up with high-frequency point cloud inputs, resulting in frame drops and registration errors. Although the original NDT method completed mapping in the one-way corridor, it lost details and failed to accurately reflect the environment as illustrated in Figure 10.

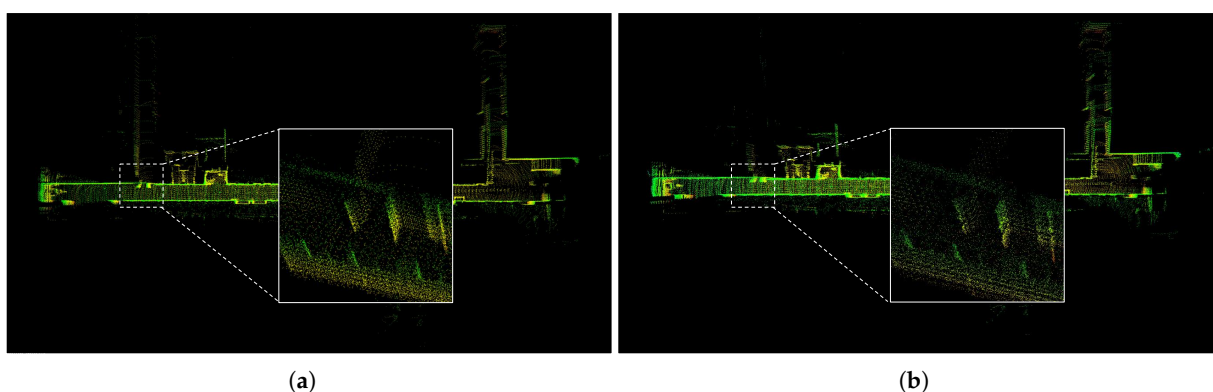


**Figure 9.** Cont.





**Figure 9.** (a–d) Maps generated by the original method. Significant mapping errors occurred in larger environments, such as (c,d).



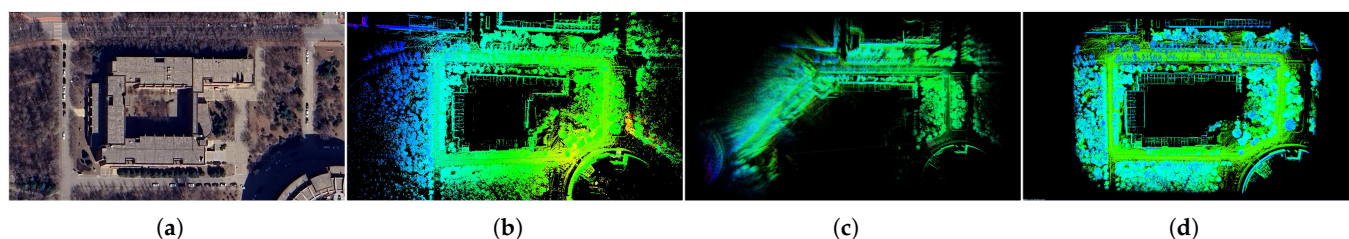
**Figure 10.** Detailed comparison between the improved and original methods. (a,b) The improved and original methods, respectively. The improved method balances detail preservation and computation speed, while the original sacrifices some environmental accuracy for mapping results.

#### 4.2.2. Outdoor Environments

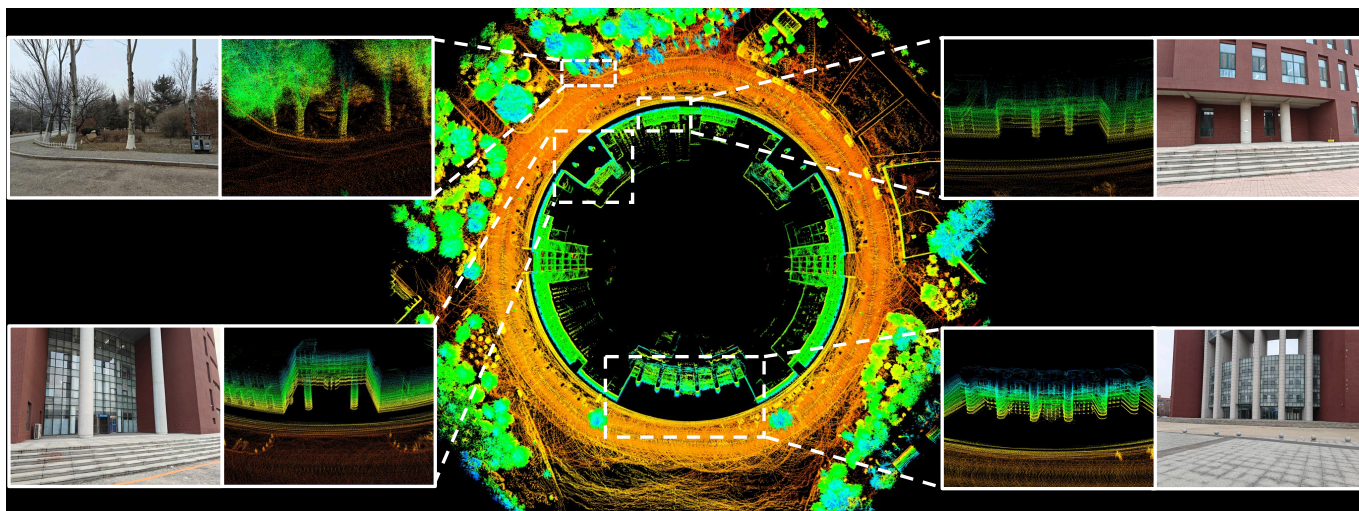
The outdoor experiments consisted of two scenarios:

- Scenario 1: A building perimeter primarily featuring straight paths with sharp or large-angle turns. Shown in Figure 11a.
- Scenario 2: A circular building with a loop path requiring continuous angular adjustments by the robot platform. Shown in Figure 12.

Both scenarios included dynamic features like vegetation and grass, adding challenges to the map construction.



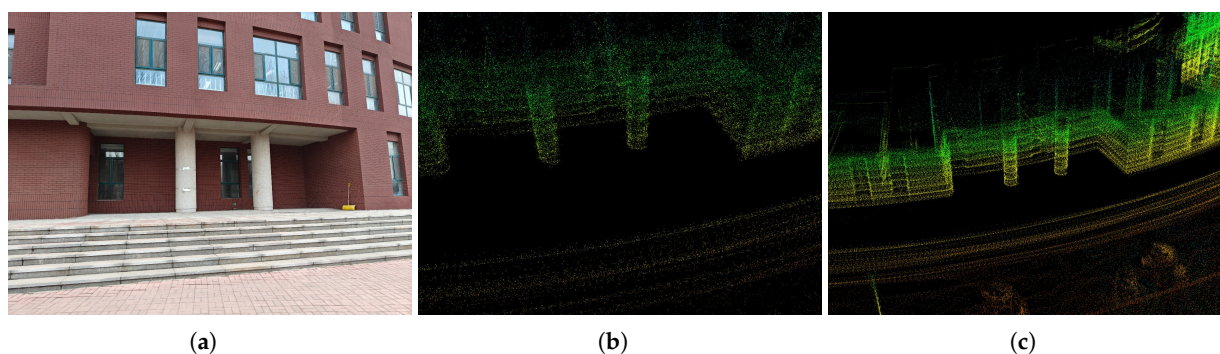
**Figure 11.** Map comparison. (a) The Google Earth image. (b) LeGO-LOAM failed to close the loop due to the lack of IMU data, leading to Z-axis drift. (c) The original NDT framework experienced significant drift in large-scale complex environments. (d) The improved method produced maps closely matching the real environment.



**Figure 12.** Detail of Scenario 2. The improved method preserved environmental details without artifacts or mismatches.

In Scenario 1, the instability of vegetation points posed higher demands on the algorithms' feature-handling capabilities. LeGO-LOAM performed relatively well in this environment, generating consistent maps with its lightweight design. However, due to the absence of IMU data in this experiment, LeGO-LOAM exhibited drift along the Z-axis during mapping and failed to close the loop during loop closure detection (Figure 11b). Meanwhile, the original NDT framework retained many unstable points during mapping, causing map drift (Figure 11c). In contrast, the improved NDT framework demonstrated significant advantages, removing dynamic interference points through a point cloud pre-processing module. With enhanced registration strategies and optimized loop closure detection, it generated a clear and accurate map closely resembling the real environment (Figure 11d), highlighting its adaptability to complex settings.

In Scenario 2, the circular building presented challenges for rotational computations in NDT-based algorithms. The original method, constrained by computation speed, required point cloud downsampling to perform basic mapping. However, this downsampling strategy resulted in maps lacking environmental detail and failing to accurately reflect the actual structure. The improved method, on the other hand, successfully completed mapping, as shown in Figure 12, retaining intricate structural details. Figure 13 further contrasts the maps generated by the original and improved methods, demonstrating the significant advantages of the improved method in terms of accuracy and detail retention.



**Figure 13.** (a–c) Scenario 2 map comparison. (b) The map generated by the original NDT method lacked details. (c) The improved method effectively preserved details.

## 5. Discussion

This paper presents an enhanced NDT framework aimed at improving the accuracy, efficiency, and robustness of LiDAR-based SLAM systems in complex environments. By integrating feature-based point cloud registration, loop closure detection, and global optimization, the proposed framework significantly improves the handling of high-frequency point clouds and challenging scenarios. This paper proposes an enhanced NDT framework that integrates feature-based point cloud registration, loop closure detection, and global optimization to address key challenges in LiDAR-based SLAM systems. The framework demonstrates superior performance in handling high-frequency point clouds and complex environments. Compared to traditional methods, it achieves significant improvements in accuracy, efficiency, and robustness, as evidenced by the experimental results.

To overcome the real-time mapping limitations of traditional NDT frameworks caused by high computational demands, we introduced a preprocessing step that removes outliers, separates the ground, and extracts key features from LiDAR point clouds. This approach effectively reduces computational load and enhances real-time performance. However, in environments with degraded features, such as those with sparse or noisy data, the feature extraction process becomes less effective, potentially leading to mapping inaccuracies. Future research could address this limitation by integrating adaptive feature extraction techniques or leveraging machine learning-based methods for more robust preprocessing.

To improve the accuracy and robustness of loop closure detection, we developed a hybrid approach that combines Radius Search and Scan Context methods. This integration leverages the complementary strengths of both methods and merges their confidence scores to enhance reliability in diverse and challenging environments. Compared to existing single-method approaches, our hybrid model significantly reduces false positive rates while maintaining high detection accuracy. This improvement is particularly important for ensuring the consistency of long-term mapping in dynamic or large-scale scenarios.

To address the issue of insufficient map precision in feature-based point cloud registration, we integrated feature-based registration with full cloud mapping during global optimization. This hybrid approach ensures that the final map retains both high accuracy and rich environmental details. The experimental results validate that this integration effectively balances computational efficiency with mapping precision, setting it apart from traditional feature-based or dense-mapping-only frameworks. Future studies could explore further improvements by incorporating real-time global optimization techniques or distributed computing strategies to handle even larger datasets.

While the proposed framework demonstrates strong performance across various scenarios, certain limitations remain. For example, the preprocessing step, though effective, may introduce biases in environments with non-standard features. Additionally, the robustness of the hybrid loop closure detection approach depends on the quality and distribution of the input point clouds, which could affect results in highly dynamic environments. Addressing these limitations will be a key focus of future research.

In conclusion, the enhanced NDT framework represents a significant advancement in LiDAR-based SLAM systems by addressing critical limitations of traditional methods. The integration of advanced preprocessing, hybrid loop closure detection, and global optimization improves the robustness and accuracy of SLAM in complex environments. These findings provide a solid foundation for further exploration and development of high-performance mapping and localization systems, especially in real-time applications.

## 6. Conclusions

This study introduced an enhanced NDT-based framework for LiDAR SLAM systems, focusing on improving computational efficiency, trajectory accuracy, and robustness in



complex environments. Experimental results demonstrate that the proposed method significantly outperforms traditional approaches in both computational speed and trajectory accuracy. Specifically, the average registration time was reduced by 66.7%, memory usage decreased by 23.16%, and CPU usage dropped by 19.25%. Furthermore, the accuracy of trajectory estimation improved by more than 35%, both in scenarios with and without loop closures. These improvements highlight the potential of the proposed framework in real-time and resource-constrained SLAM applications.

Despite its strong performance across various test environments, there is room for further optimization in key areas:

- **Feature Extraction and Matching:** Point cloud sparsity or occlusion can still affect local precision during feature extraction and matching. Future work could explore more efficient feature extraction algorithms and integrate multi-modal sensors (e.g., IMU and cameras) to enhance registration accuracy and robustness, especially in dynamic or cluttered environments.
- **Computational Efficiency in Large-Scale Environments:** While the framework performs well in typical scenarios, its computational efficiency in large-scale environments remains an area for improvement. Developing strategies to balance accuracy and processing speed is crucial for applications requiring high real-time performance, such as autonomous driving or large-scale mapping.

These directions represent promising avenues for advancing the capabilities of the proposed framework. By addressing these challenges, the framework can better meet the demands of increasingly complex and dynamic SLAM scenarios, providing a foundation for more robust and scalable localization and mapping systems in the future.

**Author Contributions:** Conceptualization, Y.R., Z.S. and X.C.; methodology, Y.R. and Z.S.; software, Y.R., Z.S. and X.C.; validation, Y.R., Z.S. and W.L.; formal analysis, Y.R., Z.S. and X.C.; investigation, Y.R. and Z.S.; resources, Y.R., X.C. and W.L.; data curation, Y.R. and Z.S.; writing—original draft preparation, Y.R. and Z.S.; writing—review and editing, Y.R., Z.S. and W.L.; visualization, Y.R., Z.S. and X.C.; supervision, Y.R., W.L. and X.C.; project administration, Y.R., W.L. and X.C.; funding acquisition, Y.R. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was partially supported by the Natural Science Foundation of Liaoning Province (No. 2021-MS-265), the fundamental research funds for the universities of Liaoning province (No. LJ212410143046), and the Basic Research Project (Key Research Project) of the Education Department of Liaoning Province (No. JYTZD2023009).

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author.

**Acknowledgments:** The authors would like to express their gratitude to the anonymous reviewers for their constructive suggestions, which have greatly enhanced the quality of this paper. We also thank the relevant teachers and students for their guidance and assistance throughout this research.

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

## References

1. Taheri, H.; Xia, Z.C. SLAM; Definition and Evolution. *Eng. Appl. Artif. Intell.* **2021**, *97*, 104032. [[CrossRef](#)]
2. Pomerleau, F.; Colas, F.; Siegwart, R. A Review of Point Cloud Registration Algorithms for Mobile Robotics. *Found. Trends® Robot.* **2015**, *4*, 1–104. [[CrossRef](#)]
3. Hess, W.; Kohler, D.; Rapp, H.; Andor, D. Real-Time Loop Closure in 2D LIDAR SLAM. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 1271–1278. [[CrossRef](#)]
4. 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](#)]

5. Cheng, L.; Chen, S.; Liu, X.; Xu, H.; Wu, Y.; Li, M.; Chen, Y. Registration of Laser Scanning Point Clouds: A Review. *Sensors* **2018**, *18*, 1641. [CrossRef] [PubMed]
6. Li, P.; Wang, R.; Wang, Y.; Tao, W. Evaluation of the ICP Algorithm in 3D Point Cloud Registration. *IEEE Access* **2020**, *8*, 68030–68048. [CrossRef]
7. Xin, W.; Pu, J. An Improved ICP Algorithm for Point Cloud Registration. In Proceedings of the 2010 International Conference on Computational and Information Sciences, Chengdu, China, 17–19 December 2010; pp. 565–568. [CrossRef]
8. Biber, P.; Strasser, W. The Normal Distributions Transform: A New Approach to Laser Scan Matching. In Proceedings of the Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453), Las Vegas, NV, USA, 27–31 October 2003; Volume 3, pp. 2743–2748. [CrossRef]
9. Saarinen, J.; Andreasson, H.; Stoyanov, T.; Lilienthal, A.J. Normal Distributions Transform Monte-Carlo Localization (NDT-MCL). In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013; pp. 382–389. [CrossRef]
10. Yang, J.; Wang, C.; Luo, W.; Zhang, Y.; Chang, B.; Wu, M. Research on Point Cloud Registering Method of Tunneling Roadway Based on 3D NDT-ICP Algorithm. *Sensors* **2021**, *21*, 4448. [CrossRef] [PubMed]
11. Hong, H.; Lee, B.H. Probabilistic Normal Distributions Transform Representation for Accurate 3D Point Cloud Registration. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 3333–3338. [CrossRef]
12. Zhang, J.; Singh, S. LOAM: Lidar Odometry and Mapping in Real-Time. *Robotics: Science and Systems Conference*. 2014. Available online: <https://www.scirp.org/reference/referencespapers?referenceid=2606507> (accessed on 27 November 2024).
13. Shan, T.; Englot, B. LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 4758–4765. [CrossRef]
14. Wang, J.; Xu, M.; Zhao, G.; Chen, Z. 3-D LiDAR Localization Based on Novel Nonlinear Optimization Method for Autonomous Ground Robot. *IEEE Trans. Ind. Electron.* **2024**, *71*, 2758–2768. [CrossRef]
15. Tang, J.; Zhang, X.; Zou, Y.; Li, Y.; Du, G. A High-Precision LiDAR-Inertial Odometry via Kalman Filter and Factor Graph Optimization. *IEEE Sens. J.* **2023**, *23*, 11218–11231. [CrossRef]
16. Wang, H.; Wang, C.; Chen, C.L.; Xie, L. F-LOAM: Fast LiDAR Odometry and Mapping. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; pp. 4390–4396. [CrossRef]
17. Kim, G.; Kim, A. Scan Context: Egocentric Spatial Descriptor for Place Recognition Within 3D Point Cloud Map. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 4802–4809. [CrossRef]
18. Xue, G.; Wei, J.; Li, R.; Cheng, J. LeGO-LOAM-SC: An Improved Simultaneous Localization and Mapping Method Fusing LeGO-LOAM and Scan Context for Underground Coalmine. *Sensors* **2022**, *22*, 520. [CrossRef]
19. Besl, P.; McKay, N.D. A Method for Registration of 3-D Shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **1992**, *14*, 239–256. [CrossRef]
20. Aoki, Y.; Goforth, H.; Srivatsan, R.A.; Lucey, S. PointNetLK: Robust & Efficient Point Cloud Registration Using PointNet. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7163–7172.
21. Lu, W.; Wan, G.; Zhou, Y.; Fu, X.; Yuan, P.; Song, S. DeepICP: An End-to-End Deep Neural Network for 3D Point Cloud Registration. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 12–21. [CrossRef]
22. Wang, Y.; Solomon, J.M. PRNet: Self-Supervised Learning for Partial-to-Partial Registration. *arXiv* **2019**, arXiv:1910.12240. [CrossRef]
23. Lin, J.; Zhang, F. Loam Livox: A Fast, Robust, High-Precision LiDAR Odometry and Mapping Package for LiDARs of Small FoV. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 3126–3131. [CrossRef]
24. Liao, L.; Fu, C.; Feng, B.; Su, T. Optimized SC-F-LOAM: Optimized Fast LiDAR Odometry and Mapping Using Scan Context. In Proceedings of the 2022 6th CAA International Conference on Vehicular Control and Intelligence (CVCI), Nanjing, China, 28–30 October 2022; pp. 1–6. [CrossRef]
25. Chen, X.; Läbe, T.; Milioto, A.; Röhling, T.; Behley, J.; Stachniss, C. OverlapNet: A Siamese Network for Computing LiDAR Scan Similarity with Applications to Loop Closing and Localization. *Auton. Robot.* **2022**, *46*, 61–81. [CrossRef]
26. Ma, J.; Zhang, J.; Xu, J.; Ai, R.; Gu, W.; Chen, X. OverlapTransformer: An Efficient and Yaw-Angle-Invariant Transformer Network for LiDAR-Based Place Recognition. *IEEE Robot. Autom. Lett.* **2022**, *7*, 6958–6965. [CrossRef]

27. Zhao, X.; Ren, X.; Li, L.; Wu, J.; Zhao, X. PA-SLAM: Fast and Robust Ground Segmentation and Optimized Loop Detection Laser SLAM System. *IEEE Trans. Aerosp. Electron. Syst.* **2024**, *60*, 5812–5822. [[CrossRef](#)]
28. Magnusson, M. The Three-Dimensional Normal-Distributions Transform: An Efficient Representation for Registration, Surface Analysis, and Loop Detection. Ph.D. Thesis, Örebro Universitet, Örebro, Sweden, 2009.
29. Cai, Y.; Xu, W.; Zhang, F. Ikd-Tree: An Incremental K-D Tree for Robotic Applications. *arXiv* **2021**, arXiv:2102.10808. [[CrossRef](#)]
30. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision Meets Robotics: The KITTI Dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [[CrossRef](#)]
31. Chen, K.; Lopez, B.T.; Agha-mohammadi, A.a.; Mehta, A. Direct LiDAR Odometry: Fast Localization with Dense Point Clouds. *IEEE Robot. Autom. Lett.* **2022**, *7*, 2000–2007. [[CrossRef](#)]
32. Quigley, M.; Gerkey, B.; Conley, K.; Faust, J.; Foote, T.; Leibs, J.; Berger, E.; Wheeler, R.; Ng, A. ROS: An Open-Source Robot Operating System. Available online: [http://lars.mec.ua.pt/public/LAR%20Projects/BinPicking/2016\\_RodrigoSalgueiro/LIB/ROS/icraoss09-ROS.pdf](http://lars.mec.ua.pt/public/LAR%20Projects/BinPicking/2016_RodrigoSalgueiro/LIB/ROS/icraoss09-ROS.pdf) (accessed on 27 November 2024).

**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.