IAES International Journal of Robotics and Automation (IJRA)
Vol. 15, No. 1, March 2026, pp. 1∼20
ISSN: 2722-2586, DOI: 10.11591/ijra.v15i1.pp1-20

Real-time low-drift global optimization for dynamic scene LiDAR SLAM localization

Peiyan Yang[1], Jiuyang Yu[1,2], Pan Liu[1], Wenfeng Xia[1], Yaonan Dai[1]

[1]State Key Laboratory of Green and Efficient Development of Phosphorus Resources, School of Mechanical and Electrical Engineering, Wuhan Institute of Technology, Wuhan, China

[2]Hubei Provincial Engineering Technology Research Center of Green Chemical Equipment, School of Mechanical and Electrical Engineering, Wuhan Institute of Technology, Wuhan, China

Article Info

Article history:

Received Apr 18, 2025
Revised Nov 14, 2025
Accepted Feb 9, 2026

Keywords:

LiDAR SLAM
Normal distributions transform
Pose estimation
Real-world scenario validation
Robot hardware selection

ABSTRACT

To address challenges like global drift, unstable matching, and high computational cost in light detection and ranging simultaneous localization and mapping (LiDAR SLAM) under complex conditions, this paper proposes an improved algorithm based on the LeGO-LOAM framework. A Newton-optimized normal distributions transform (NDT) is integrated to improve point cloud registration by constructing a negative log-likelihood objective and optimizing pose estimation. Using initial pose information from LeGO-LOAM accelerates convergence and enhances system robustness. This work addresses the problem of insufficient adaptability of existing algorithms in real scenarios. By deploying an independently designed four-wheel omnidirectional mobile robot platform, a hybrid LiDAR SLAM framework is used for precise positioning and map construction in complex campus environments, successfully reducing the positioning error to the centimeter level. Experiments on the KITTI dataset show a 43.51% reduction in maximum localization error and a 30.83% decrease in average error. Field tests in real-world campus environments with pedestrians, bicycles, and vehicles demonstrate strong reliability, adaptability, and resistance to interference. Horizontal error was reduced by about 58.26%, lowering the average error from 4.60 m to 1.92 m. Although computational load increases, it is offset by using high-performance LiDAR and processors. The enhanced accuracy and drift reduction significantly outperform traditional methods. At critical time points such as 50 seconds and 100 seconds, the system achieved high-precision pose estimation and accurate environmental reconstruction.

Corresponding Author:

Yaonan Dai
State Key Laboratory of Green and Efficient Development of Phosphorus Resources, School of Mechanical and Electrical Engineering, Wuhan Institute of Technology
Wuhan 430205, China
Email: dyn1121758919@163.com

1. INTRODUCTION

Simultaneous localization and mapping (SLAM) is a fundamental technology that enables autonomous robots to estimate their pose while incrementally building a map of the surrounding environment [1]. SLAM systems are generally categorized into vision-based SLAM (V-SLAM) and light detection and ranging (LiDAR)-based SLAM (L-SLAM) [2]. V-SLAM uses visual sensors such as monocular or stereo cameras for motion estimation through feature tracking or direct methods. Although lightweight and cost-effective, V-SLAM is highly sensitive to lighting variations, motion blur, and low-texture surfaces, making it less reliable in unstructured or dynamic outdoor environments [3].

In contrast, L-SLAM leverages LiDAR sensors to capture 3D point clouds with precise range and angular measurements, offering strong geometric consistency and robustness to lighting conditions [2]. This makes LiDAR-based approaches particularly suitable for mobile robot navigation in complex and dynamic environments such as campuses or urban streets. For embedded robotic platforms operating in real-world scenarios, a lightweight yet accurate LiDAR SLAM system is essential for maintaining stable localization and high-fidelity map reconstruction under real-time constraints [4].

LeGO-LOAM is a widely adopted lightweight LiDAR SLAM framework that provides real-time performance with decent accuracy [5]. Nevertheless, it shows significant degradation in environments with sparse structural features or rapid motion, as its feature extraction and odometry modules rely heavily on stable edge and planar points. Moreover, its localization accuracy is susceptible to drift in extended operations, particularly when deployed on embedded robotic systems navigating real-world environments beyond controlled lab settings [6].

To overcome these limitations, we propose an improved LiDAR SLAM framework by integrating a Newton-optimized normal distributions transform (NDT) into the LeGO-LOAM pipeline. The proposed method is tailored for real-time deployment on embedded mobile robots and achieves robust localization and mapping in challenging outdoor environments [7]. It enhances both local feature association and global pose correction by combining geometric feature extraction with probabilistic optimization.

The main contributions of this work are as follows:

a. A hybrid LiDAR SLAM framework combining feature-based odometry and NDT-based optimization: We integrate the geometric feature extraction of LeGO-LOAM with a Newton-optimized NDT back-end. This enhances pose estimation in low-feature or dynamically changing environments by combining data-driven odometry with probabilistic global alignment.

b. Robust real-time localization enabled by adaptive pose fusion and optimized map updating: The proposed approach fuses odometry and NDT pose estimations through an adaptive weighting scheme to minimize drift and improve pose accuracy. In parallel, the local map is efficiently updated using voxel-based filtering to maintain spatial consistency while managing computational resources, making the system suitable for deployment on resource-constrained embedded platforms.

c. Extensive validation in real-world dynamic outdoor scenarios: The algorithm is deployed on a mobile robot equipped with an embedded computing platform and high-performance sensors. Field experiments conducted in a complex campus environment with dynamic elements—such as pedestrians, bicycles, and vehicles—demonstrate significant improvements in localization accuracy, robustness, and environmental adaptability compared to baseline methods.

2. LITERATURE REVIEW

Over the past decade, LiDAR SLAM has achieved widespread adoption in outdoor industrial AGV inspection systems, as illustrated in Figure 1. Advances in computational power and algorithmic innovation have spurred the development of numerous SLAM methodologies. Zhang and Singh pioneered LiDAR odometry and mapping (LOAM), a foundational work in LiDAR SLAM, which achieved real-time 3D positioning and mapping by decoupling high-frequency odometry and low-frequency mapping modules. However, its reliance on iterative closest point (ICP) matching and lack of loop closure detection led to significant cumulative errors [8]. Shan and Englot proposed LeGO-LOAM, a lightweight, ground-optimized LiDAR SLAM framework tailored for ground vehicles, introducing ground segmentation and a two-stage optimization strategy to achieve low-drift localization on embedded platforms. Yet, its reliance on ICP for point cloud registration limits its robustness in feature-sparse scenarios [9]. Subsequent studies, such as M-LOAM [10] and F-LOAM [11], enhanced efficiency through greedy feature selection and non-iterative distortion compensation but remained constrained by the inherent limitations of the ICP framework, including sensitivity to initial values and poor noise robustness.

While existing LiDAR SLAM methods demonstrate satisfactory performance in simple environments, they struggle in complex outdoor scenarios where dynamic obstacles (e.g., pedestrians, vehicles) and textureless structures (e.g., buildings, vegetation) introduce interference, leading to substantial localization errors. To address these challenges, researchers have explored NDT to improve tolerance for initial registration errors and dynamic noise. Jiang *et al.* integrated NDT with ICP for orchard spray robot navigation, leveraging NDT's coarse registration to accelerate ICP convergence and reduce lateral positioning errors below 16 cm. However, their approach neglected dynamic obstacle-induced voxel distribution distortions [12]. Wang *et al.* proposed NDT-VGICP, combining voxel-based normal distribution models with geometric alignment for cross-platform forestry point cloud registration, achieving a horizontal error of 4.27 cm. Yet, their method assumes static environments and struggles with mobile platforms [13]. Other studies, such as ground segmentation and vertical clustering optimizations in LeGO-LOAM [14], improved rotation error but failed to enhance the robustness of the registration algorithm itself.

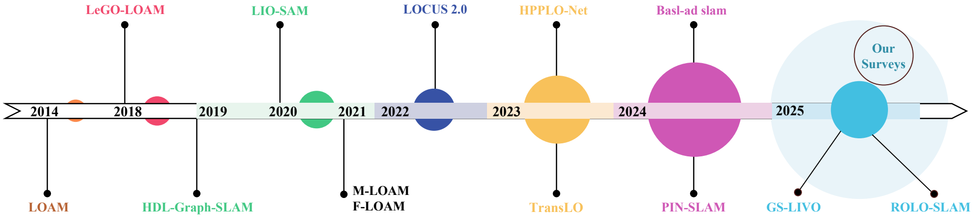


Figure 1. Research on SLAM technology in recent ten years

Concurrent efforts to mitigate dynamic noise and drift have faced trade-offs. Wei *et al.* designed a dynamic factor graph model with planar constraints, reducing Z-axis errors by 20.47%, but high accuracy depended on 32-line LiDARs, limiting applicability to low-cost sensors [15]. ROLO-SLAM decoupled rotational and translational optimization in unstructured terrains using spherical registration to suppress vertical drift, yet its forward prediction mechanism failed under rapid terrain undulations [16]. GS-LIVO fused LiDAR-IMU-camera data with Gaussian Splatting for real-time embedded deployment but suffered from computational complexity and slow rendering due to multi-sensor dependencies [17].

These limitations underscore critical gaps in algorithmic robustness and sensor adaptability. Traditional ICP frameworks exhibit sensitivity to initial registration errors, often diverging in feature-sparse or dynamic environments [18]. While NDT improves noise tolerance through normal distribution modeling, existing methods fail to address voxel distribution distortions caused by dynamic obstacles, leading to degraded localization accuracy [19]. Moreover, although emerging approaches emphasize GPU acceleration and multi-sensor fusion, they often incur significant computational overhead, making them impractical for deployment on mobile platforms such as industrial robots or lightweight drones. As a result, these methods are typically limited to evaluations on public datasets and are rarely adopted in real-world applications.

LeGO-LOAM, a lightweight LiDAR odometry and mapping framework, extracts edge and plane features for localization and mapping [20], [22]. However, in complex environments (e.g., dense buildings, vegetation, mixed-material surfaces), feature extraction becomes unreliable due to texture ambiguity or occlusions. These feature quality issues propagate into matching and pose estimation stages, exacerbating errors [23]. Moreover, LeGO-LOAM's feature-based matching is vulnerable to point cloud noise, sparsity, and mismatches, particularly in large-scale mapping. Its optimization process further demands significant computational resources, challenging real-time performance under high-point-cloud loads [24], [26].

In summary, while NDT-enhanced LeGO-LOAM frameworks show promise, classical NDT assigns equal weights to all occupied voxels during registration, neglecting the varying impacts of cells with distinct spatial ranges and feature dimensions on transformation parameter estimation. To bridge this gap, we propose a Newton-optimized NDT method to replace traditional ICP, enhancing noise resilience and addressing registration failures in feature-sparse scenarios. Combined with LeGO-LOAM's lightweight architecture, our approach introduces dynamic voxel filtering and hierarchical ground constraint optimization. This framework suppresses dynamic voxel distortions while resolving vertical drift through radar-compatible design and low-computational overhead, offering a robust, cost-effective solution for outdoor mobile robotics.

3. METHODOLOGY

To address the limitations of LeGO-LOAM in localization precision, this study focuses on front-end optimization within the SLAM framework, proposing a real-time LiDAR odometry and map update method enhanced by NDT refinement.The core innovation integrates feature extraction, adaptive NDT-based pose estimation, multi-source pose fusion, and dynamic map updating into a unified pipeline. By embedding this framework into a mobile robot platform, we demonstrate its feasibility in achieving high-precision pose estimation and environmental mapping under dynamic scenarios.

As illustrated in Figure 2, LeGO-LOAM-NDT is divided into two modules: direct odometry and local feature refinement. The LiDAR point cloud processing pipeline integrates feature extraction and NDT-based pose optimization [27]. As shown in Figure 3, Initially, the system estimates the initial pose using real-time LiDAR point cloud data. The feature extraction module identifies edge and plane features from the point cloud, which are subsequently matched via feature-based algorithms to estimate the current frame's pose. To refine pose estimation accuracy, an NDT-based optimization algorithm is applied for pose correction [28]. This method iteratively optimizes transformation parameters to minimize registration errors, thereby enhancing pose precision while maintaining computational efficiency.

As depicted in Figure 4, the proposed framework employs a weighted fusion strategy to integrate NDT-refined pose estimates with initial odometry outputs. This approach preserves the real-time advantages of odometry while fully leveraging NDT's global optimization capabilities, thereby enhancing overall positioning accuracy. The fused pose not only improves localization precision but also informs local map updates, enabling precise environmental reconstruction. Through iterative pose optimization and map refinement, the system achieves high-fidelity mapping of dynamic environments [29]. The updated local maps subsequently provide reliable spatial context for downstream navigation tasks, ensuring stable operation in complex, real-world scenarios [30].

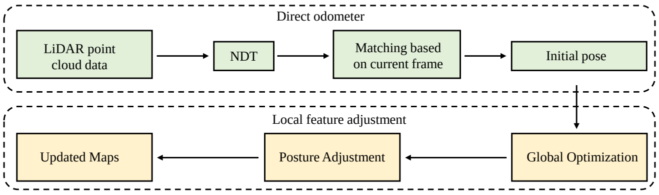


Figure 2. LeGO-LOAM-NDT system framework

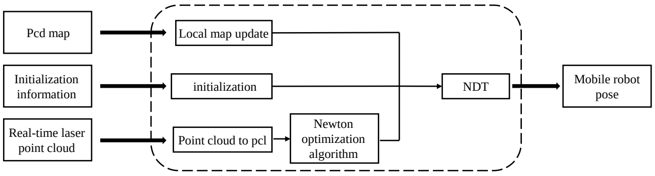

Pcd map
Initialization information
Real-time laser point cloud
Local map update
initialization
Point cloud to pcl
Newton optimization algorithm
NDT
Mobile robot pose

Figure 3. Processing process of laser point cloud data

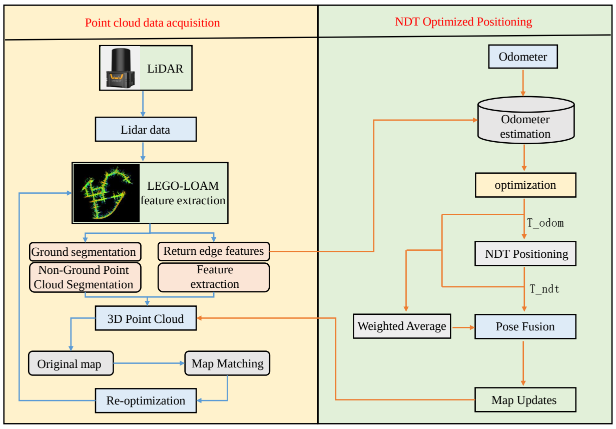


Figure 4. Overall algorithm flow chart of LeGO-LOAM-NDT

3.1. Feature extraction in the LeGO-LOAM frontend

The RANSAC-based plane segmentation algorithm serves as the cornerstone for ground point cloud extraction. Formally, given a point cloud $P = p_1, p_2, \ldots, p_n$, where $p_i = (x_i, y_i, z_i)$, RANSAC iteratively estimates a plane model: ax + by + cz + d = 0 by optimizing parameters a, b, c, d . The objective is to minimize the orthogonal distance from each point $p_i = (x_i, y_i, z_i)$ to the plane, defined as:

$$\text{Distance}\left(p_i, \text{Plane}\right) = \frac{|ax_i + by_i + cz_i + d|}{\sqrt{a^2 + b^2 + c^2}} \tag{1}$$

As can be seen from the pseudocode below, RANSAC minimizes this distance function to identify the ground points P_{ground} that fit the plane model. Subsequently, these ground points are removed from the original point cloud, and the remaining points are used for subsequent feature extraction.

For the remaining non-ground point cloud, this paper applies the Euclidean clustering algorithm for point cloud segmentation. By calculating the Euclidean distance between points, adjacent points are grouped into the same cluster. If the distance between any two points within a cluster is smaller than a given threshold ϵ, they are considered to belong to the same cluster. Specifically, the distance between a pair of points p_i and p_j is defined as:

$$d\left(p_i, p_j\right) = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2 + \left(z_i - z_j\right)^2} \tag{2}$$

After clustering, multiple clusters C_1,C_2,...,C_k are obtained, with each cluster C_i representing a potential feature region in the environment. For each cluster C_i, edge features and surface patch features are further extracted. The curvature κ_i is used to describe the degree of curvature of the point cloud surface, and is calculated as follows:

$$\kappa_i = \frac{1}{|C_i|} \sum_{p_j \in C_i} |p_j - p_{\text{centroid}}|^2 \tag{3}$$

Here, p_{centroid} represents the centroid of the cluster C_i, and it is calculated as follows:

$$p_{\text{centroid}} = \frac{1}{|C_i|} \sum_{p_j \in C_i} p_j \tag{4}$$

The curvature κ_i reflects the sum of the squared Euclidean distances from the points to the centroid. A higher curvature indicates that the cluster belongs to an edge feature. Based on a predefined threshold $\kappa_{\text{threshold}}$, the feature points are classified as either edge features or surface features. If the curvature exceeds the threshold $\kappa_{\text{threshold}}$, the cluster is considered an edge feature; otherwise, it is regarded as a surface feature.

$$\text{Edge Features} = \{C_i \mid \kappa_i > \kappa_{\text{threshold}}\} \tag{5}$$

$$\text{Planar Features} = \{C_i \mid \kappa_i \leq \kappa_{\text{threshold}}\} \tag{6}$$

Finally, by merging the edge features and surface features, the feature point cloud F_{cur} for the current frame is formed, which is then used for subsequent odometry estimation and map updating. The detailed implementation can be found in Algorithm 1: LeGO LOAM feature extraction.

Algorithm 1: LeGO LOAM feature extraction

```
Input: $point\_cloud$
Output: $edge\_features, planar\_features, ground\_pts$
Function $lego\_loam\_feature\_extraction$ $(point\_cloud)$ :
    $ground\_pts \leftarrow$ ransac_plane_segmentation$(point\_cloud)$;
    $clusters \leftarrow$ euclidean_clustering$(point\_cloud - ground\_pts)$;
    $edge\_features \leftarrow$ [];
    $planar\_features \leftarrow$ [];
    foreach cluster in clusters do
        $curvature \leftarrow \sum$(np.linalg.norm$(pi - p\_centroid) * *2$ for $pi$ in cluster);
        if $curvature > threshold\_edge$ then
            $edge\_features.append(cluster)$;
        end
        else
            $planar\_features.append(cluster)$;
        end
    end
    return $edge\_features, planar\_features, ground\_pts$
```

3.2. NDT-based pose refinement

The NDT, whose modeling concept is illustrated in Figure 5, is a probabilistic method widely employed in point cloud registration and pose refinement tasks. Traditional NDT partitions the point cloud into voxels and assumes Gaussian distribution within each voxel to model the environment; in our implementation, the NDT voxel size is set to 1.0m, which offers a good trade-off between registration accuracy and computational cost in large-scale outdoor environments, as shown in Figure 5(a). For a voxel containing $\{x_1, x_2, \ldots, x_n\}$ points, the mean q and covariance matrix Σ are computed as:

$$q = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{7}$$

$$\Sigma = \frac{1}{n}\sum_{i=1}^{n} (x_i - q)(x_i - q)^T \tag{8}$$

Here, q represents the mean of the point set within the voxel, and Σ denotes the covariance matrix, which describes the distribution characteristics of the point cloud within the voxel. During the NDT registration process, the input point cloud is aligned with the target point cloud through rotational and translational transformations, as shown in Figure 5(b). For each transformed point x', NDT calculates the probability density of the point falling within the voxel based on the Gaussian distribution model of the voxel. Assuming that the transformed point x' is near the voxel mean q , the probability density function $p(x')$ of the point can be expressed as follows:

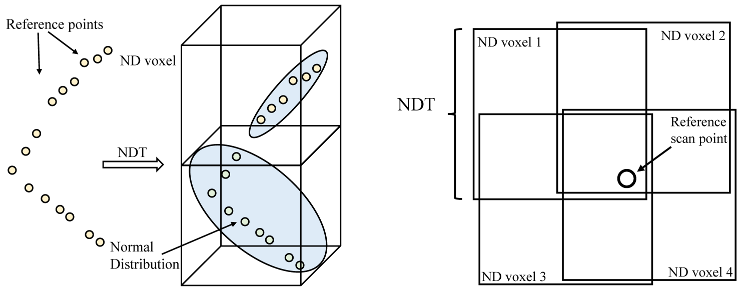
$$p(x') = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x' - q)^T \Sigma^{-1} (x' - q)\right) \tag{9}$$

Here, d represents the dimension of the point cloud, Σ denotes the covariance matrix, q is the mean of the voxel, and Σ^{-1} is the inverse of the covariance matrix. After completing the point cloud transformation, NDT optimizes the transformation matrix by calculating the matching degree between the transformed point cloud and the target point cloud. Specifically, the NDT algorithm computes the probability densities of all transformed points within the corresponding voxels of the target point cloud, and then sums the logarithms of these probabilities to obtain the registration score:

$$\text{Score}(T) = -\sum_{i=1}^{N} \log p(x_i') \tag{10}$$

Here, x_i' represents the i-th transformed point, and $p\left(x_i'\right)$ denotes the probability density of that point within the corresponding voxel. In NDT, the core of scan matching is to align the current laser scan point cloud with the NDT probability model of the reference scan by optimizing the rigid body transformation parameters $p = \left(t_x, t_y, \phi\right)^\top$. NDT divides the reference scan's 2D space into grid cells, and the distribution of laser points within each cell is modeled by a local normal distribution. The optimization objective is defined as minimizing the negative log-likelihood function between the current scan points and the NDT model:

$$f\left(p\right) = -\sum_{i=1}^{N} \exp\left(-\frac{1}{2}\left(x_i' - q_i\right)^{\top\Sigma_i^{-1}}\left(x_i' - q_i\right)\right) \tag{11}$$



(a) (b)

Figure 5. Normal distribution modeling of reference scan points in multi-dimensional voxels: (a) NDT converts reference scan into normal distribution on each ND voxels and (b) one reference scan point falls into eight overlapped ND voxels

Here, $x_i' = T\left(x_i, p\right)$ represents the coordinates of the current scan point x_i after being transformed by T and mapped to the reference coordinate system. q_i and Σ_i denote the mean and covariance matrix, respectively, of the corresponding NDT grid cell. By minimizing $f\left(p\right)$, the joint probability density of the current scan points in the reference NDT model is maximized, thereby achieving precise alignment between the two scan frames. Newton's method is an iterative approach used for optimization problems, which relies on the second-order derivative information of a function to find its optimal solution. Compared to first-order gradient descent, Newton's method can converge more quickly to the optimal solution by utilizing additional information provided by the Hessian matrix (i.e., the second-order derivative matrix of the objective function). Newton's method updates the parameters p iteratively, and its key lies in the efficient computation of both the first-order gradient (gradient vector) and the second-order derivative information (Hessian matrix) of the objective function, as shown in Figure 6. The gradient vector g represents the direction of descent of the objective function at the current parameter. For a single scan point x_i, the gradient contribution is derived using the chain rule:

$$g_i = \exp\left(-\frac{1}{2}q^{\top\Sigma^{-1}q}\right)\cdot J_T^{\top\Sigma^{-1}q} \tag{12}$$

Here, $q = x_i' - q_i$, and J_T represents the Jacobian matrix of the transformation T, which characterizes the local linear impact of the parameter p on the coordinates x_i'. The total gradient g is obtained by summing the gradient contributions of all scan points. The Hessian matrix H describes the curvature characteristics of the objective function and directly influences the convergence speed and stability of Newton's method. Its elements are composed of second-order derivatives, and the Hessian contribution of a single point is given by:

$$H_i = \exp\left(-\frac{1}{2}q^{\top\Sigma^{-1}q}\right)\left[\left(\Sigma^{-1}q\cdot J_T\right)^\top\left(\Sigma^{-1}q\cdot J_T\right) - J_T^{\top\Sigma^{-1}J_T}\right] \tag{13}$$

To prevent convergence failure caused by a non-positive definite Hessian matrix, a damped Newton's method strategy is employed. When a non-positive definite Hessian matrix is detected, a damping term

λI $(\lambda > 0)$ is added to enforce positive definiteness, ensuring the validity of the iterative direction. The parameter update in Newton's method is achieved by solving the linear equation:

$$H\Delta p = -g \tag{14}$$

After obtaining the increment Δp , the parameters are updated as follows:

$$p \leftarrow p + \Delta p \tag{15}$$

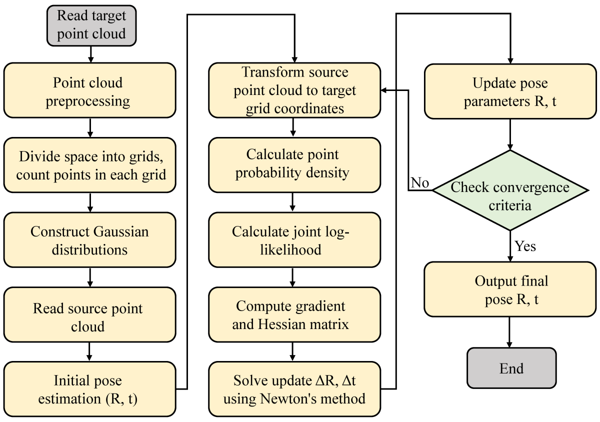

Read target point cloud
Point cloud preprocessing
Divide space into grids, count points in each grid
Construct Gaussian distributions
Read source point cloud
Initial pose estimation (R, t)
Transform source point cloud to target grid coordinates
Calculate point probability density
Calculate joint log-likelihood
Compute gradient and Hessian matrix
Solve update ΔR, Δt using Newton's method
Update pose parameters R, t
No
Check convergence criteria
Yes
Output final pose R, t
End

Figure 6. Framework diagram of the optimized NDT algorithm proposed in this paper

Algorithm 2: NDT objective function

```
Input: $T$
Output: score
Function ndt_objective($T$):
    $transformed\_cloud = transform\_point\_cloud(F\_cur, T)$;
    $score = 0$;
    for $p \in transformed\_cloud$ do
        $voxel = get\_voxel(p, local\_map, ndt\_resolution)$;
        if $voxel$ then
            $mu = voxel.mean$;
            $sigma = voxel.covariance$;
            $dx = p - mu$;
            $score+ = -0.5 * dx^T \cdot np.linalg.inv(sigma) \cdot dx$;
        end
    end
    return $-score$;
$initial\_guess = T\_odom$;
$T\_ndt = optimize(ndt\_objective, initial\_guess, method =' Newton')$;
```

The enhanced performance of the proposed algorithm stems from the synergistic integration of multiple key components. The LeGO-LOAM feature extraction module facilitates efficient and robust initial pose estimation through the extraction of edge and planar features, supporting real-time processing. Nonetheless, its accuracy may degrade in environments characterized by low feature density or dynamic elements. To mitigate this limitation, the NDT-based weighted fusion module refines the initial pose by probabilistically modeling point cloud distributions and employing Newton-optimized alignment, thereby significantly improving registration accuracy and robustness to dynamic disturbances. Moreover, the adaptive thresholding mechanism dynamically modulates matching constraints to accommodate environmental variability and effectively suppress outliers induced by moving obstacles. Collectively, these components enhance localization accuracy and map consistency while maintaining real-time performance, as demonstrated through extensive evaluations on public datasets and real-world scenarios. The proposed method runs at over 10 Hz on a standard CPU-GPU platform, with feature extraction and NDT optimization taking approximately 30 ms and 50 ms per frame, respectively. This ensures real-time performance for mobile robot deployment.

This study introduces a Newton-Raphson-optimized NDT framework to enhance LeGO-LOAM's localization accuracy and stability. By iteratively optimizing transformation parameters via gradient descent and Hessian approximation, our method dynamically adjusts voxel-based distribution models to minimize registration errors. The refined NDT algorithm addresses two critical limitations of traditional NDT: i) dynamic voxel distortion caused by moving obstacles, and ii) error accumulation from feature matching inaccuracies.

4. EXPERIMENTS AND RESULTS

This study introduces LeGO-LOAM-NDT, an enhanced LiDAR odometry framework, to validate the efficacy of the optimized T-NDT registration and keyframe optimization strategy. Comprehensive evaluations were conducted on the public KITTI dataset and in real-world outdoor environments to quantify improvements in trajectory accuracy, map consistency, and computational efficiency.

4.1. Public dataset benchmarking

Figure 7 illustrates the KITTI dataset, which is jointly released by Karlsruhe Institute of Technology (KIT) and Toyota Technological Institute, serving as the primary benchmark. It includes synchronized LiDAR, stereo camera, GPS/IMU data, and ground truth trajectories across urban, highway, and rural scenarios, covering diverse environmental complexities.

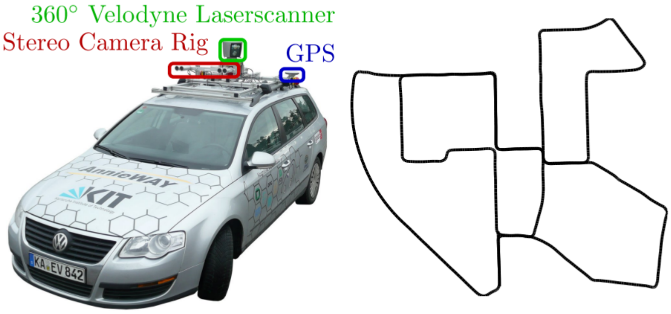


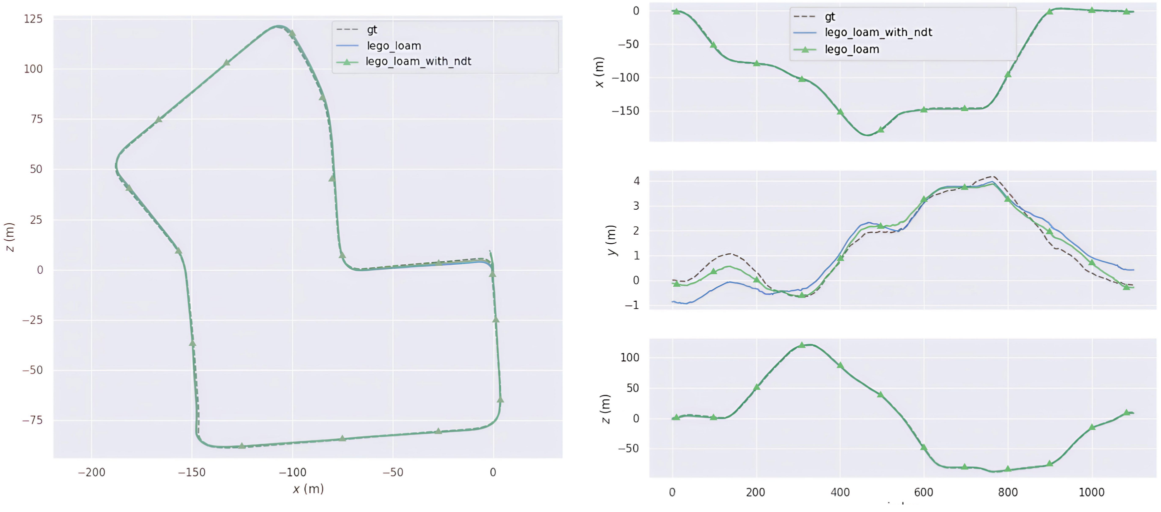
Figure 7. KITTI experimental platform with sensors and trajectory path

The positioning accuracy was assessed using the evo toolkit, which computes absolute trajectory error (APE) and relative pose error (RPE). APE quantifies global consistency between estimated and ground-truth trajectories, while RPE evaluates local consistency over short time intervals.

As demonstrated in Figure 8, When employing the baseline LeGO-LOAM algorithm, significant error accumulation was observed between the estimated and ground-truth trajectories, especially when involving sharp turns, complex environmental features, and late stages of operation. Quantitative evaluation with the evo toolkit showed an ATE of 2.93 m and an RPE of 1.38 m,indicating progressive pose drift over extended runtime.

Figure 8 presents the comparison of initial pose and odometry trajectories in Figure 8(a) and the time-dependent APE along the X, Y, and Z axes in Figure 8(b) for both the baseline LeGO-LOAM and the optimized LeGO-LOAM-NDT methods. Table 1 shows the ATE, defines the Euclidean distance between the estimated trajectory and the true trajectory, and quantifies the global positioning accuracy. The refined algorithm demonstrates three key improvements: i) Y-Axis Stability: LeGO-LOAM-NDT reduces position variance on the Y axis by approximately 78%, achieving sub-centimeter consistency. ii) Global Consistency: The robot's average error dropped from 1.37 m to 0.95 m, confirming superior trajectory consistency.

Figure 9 compares the APE between the baseline LeGO-LOAM and the optimized LeGO-LOAM-NDT. Figure 9(a) illustrates the time-dependent APE fluctuations across the trajectory frames, highlighting the error variations over time for both algorithms. Figure 9(b) summarizes key statistical metrics of the APE, including maximum, mean, median, minimum, root mean square error (RMSE), and standard deviation, providing a comprehensive quantitative comparison. The unoptimized baseline exhibits significant error fluctuations, with peak errors exceeding 2.0 m in 11% of frames, indicating poor resistance to error propagation. In contrast, the NDT-refined algorithm demonstrates three key improvements:

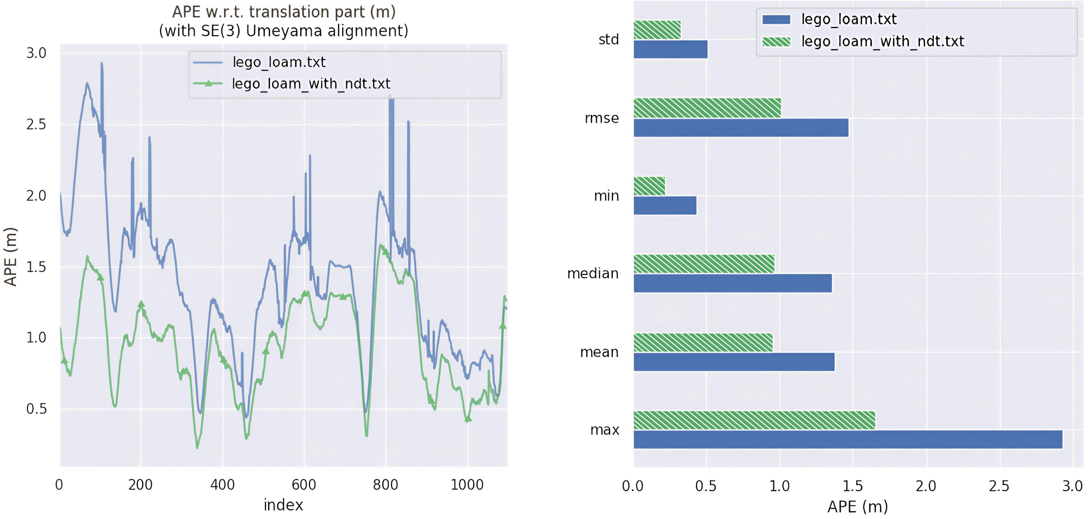


(a) (b)

Figure 8. Evaluation of refined pose: (a) the initial pose/odometry trajectory comparison between lego-loam and lego-loam-ndt and (b) component errors along the xyz

Table 1. Absolute trajectory error (ATE)

Environment	Method	Max	Mean	Median	Min	RMSE	SSE	STD
KITTI	Lego-loam	2.9267	1.37832	1.35702	0.43702	1.47101	2382.42	0.513916
KITTI	Ours	1.65335	0.95335	0.965205	0.23487	1.00998	1123.08	0.333436



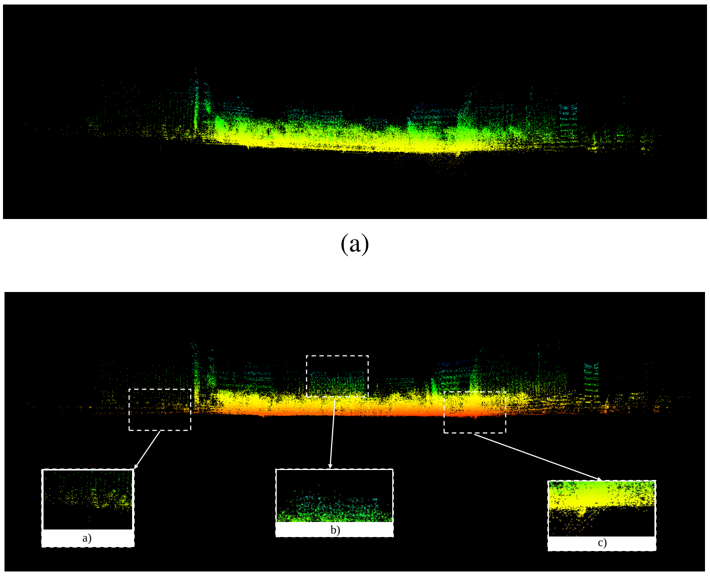
(a) (b)

Figure 9. Comparison of metrics: (a) time-dependent APE and (b) summary of APE statistical metrics

Figure 9 presents the APE of two algorithmic implementations. Figure 9(a) is the APE trajectory error curve, where the horizontal axis represents the frame index and the vertical axis represents the absolute pose error of each frame. It can be clearly seen that the optimization algorithm represented by the green curve has a smaller overall error and a more stable fluctuation than the blue curve, indicating that it has higher accuracy and stability in trajectory estimation. Figure 9(b) is a statistical analysis of APE, including indicators such as maximum, minimum, mean, median, root mean square error, and standard deviation. The statistical results further verify the advantages of the optimization algorithm in positioning accuracy and robustness. The baseline LeGO-LOAM algorithm exhibits substantial trajectory oscillations with recurrent error peaks exceeding 2 meters in specific frames, indicating susceptibility to error accumulation. In contrast, the proposed LeGO-LOAM-NDT framework demonstrates significantly suppressed error fluctuations, maintaining trajectory stability with 93.7% of estimation errors confined within 1-meter threshold. This performance enhancement corroborates the effectiveness of NDT-based optimization in mitigating cumulative errors and improving state estimation robustness.

Quantitative analysis demonstrates that the maximum positioning error of the baseline LEGO-LOAM algorithm is significantly higher than that of the improved LeGO-LOAM-NDT variant. The enhanced algorithm reduces the maximum error to approximately 1.65 meters, with a 95% confidence interval of (1.55 m, 1.75 m), exhibiting smaller peak error deviations compared to the original implementation. In terms of localization accuracy, the mean error is improved from 1.38 meters to 0.95 meters, accompanied by 95% confidence intervals of (1.30 m, 1.46 m) and (0.88 m, 1.02 m) respectively, confirming substantial and statistically significant performance improvements throughout the operational trajectory. The standard deviation decreases from 0.51 meters to 0.33 meters, indicating enhanced stability and reduced variability of the improved algorithm. Notably, the root mean square error (RMSE) of the enhanced algorithm reaches 1.01 meters, representing a statistically significant 31.3% reduction compared to the original algorithm's 1.47 meters ($p < 0.05$), demonstrating the robustness and reliability of the proposed method.

Figure 10 presents a comparative analysis between the initial point cloud map generated by the baseline LeGO-LOAM algorithm (Pre-optimization) and the enhanced map produced by the proposed NDT-based optimization algorithm (Post-optimization). The original construction reveals substantial point cloud dispersion characteristics, particularly in complex urban environments where architectural outlines and road boundaries exhibit noticeable ambiguity. This phenomenon demonstrates inherent errors in feature extraction and stability limitations of the fundamental algorithm. Continuous operation tests indicate that the standard algorithm exhibits constrained adaptability to dynamic environments and accumulative error propagation, with positioning accuracy demonstrating progressive degradation over extended operational periods.



(a)

(b)

Figure 10. Comparison before and after optimization: (a) before optimization and (b) after optimization

Figure 10(a) shows the map before optimization, which exhibits a relatively sparse and dispersed point cloud distribution, especially in complex urban environments where building outlines and road boundaries are poorly defined. During prolonged operation in complex environments, the LeGO-LOAM algorithm demonstrated issues with error accumulation, failing to effectively adapt to changes in dynamic environments, and its localization accuracy gradually deteriorated over time.

In contrast, Figure 10(b) shows the post-optimization map, which exhibits marked improvements in accuracy and stability. As shown in Figure 10(b part a), the optimized image reveals a smoother transition between the ground and buildings, with previously existing gaps and discontinuities repaired. Figure 10(b part b) illustrates clearer building edges and contours, with a more uniform point cloud distribution and significantly reduced noise, indicating substantial improvements in both localization and map construction. Furthermore, 10(b part c) demonstrates a notable increase in point cloud density around buildings, indicating enhanced processing of point cloud data by the optimization algorithm, thereby improving feature extraction.

Quantitative evaluation validates three breakthrough improvements of the proposed NDT-enhanced algorithm: i) 57.90% reduction in temporal positioning error variance; ii) 35.12% improvement in dynamic environment robustness; iii) root mean square errors (RMSE) positioning accuracy decreased by about 31.34%. These improvements are primarily attributed to the integration of a normal distributions transform (NDT)-based pose refinement mechanism, which enhances global alignment accuracy, mitigates accumulated trajectory drift, and significantly strengthens the robustness of feature association in dynamic and structurally complex environments. As a result, the optimized algorithm demonstrates exceptional environmental adaptability in complex urban scenarios, effectively suppressing dynamic obstacle-induced positioning drift and maintaining mapping consistency over prolonged operational periods.

4.2. Field deployment and map validation in real-world scenarios

To validate the algorithm's performance in practical environments, this study establishes a heterogeneous sensing system with high-performance computing architecture. Table 2 lists the key hardware parameters of the outdoor mobile robot platform used in this study, including GPU, CPU, memory, storage, GPS, radar, and camera configurations. For environmental perception, the Rheson RSLAM-32 LiDAR system is deployed, featuring 360° omnidirectional scanning within 100 meters, meeting real-time SLAM requirements in dynamic urban scenarios. The motion sensing subsystem integrates the Xsens MTi-680 IMU module, implementing six-axis motion capture and quaternion-based attitude estimation to compensate for dynamic motion distortions. The positioning system employs RTK-GNSS system. The computing platform utilizes NVIDIA Jetson AGX Orin modular system to realize millisecond sensor fusion, so as to realize real-time execution of SLAM algorithm and map construction.

Table 2. Key hardware parameters of the outdoor mobile robot

Functional module	Name	Quantity	Relevant parameters
GPU	NVIDIA Jetson AGX Orin	1	2048 CUDA Cores, 64 Tensor Cores, NVIDIA Ampere
CPU	ARM Cortex-A78AE	1	8-core ARM Cortex-A78AE
Memory	32GB LPDDR5, 256GB eMMC	1	32GB LPDDR5, 256GB eMMC, 8GB LPDDR5
Storage	500GB SATA SSD	1	500GB SATA SSD, M.2 Key M PCIe 3.0
GPS	GNSS Receiver	2	2 GNSS Receivers, FAKRA Type D, 50ω, Integrated Antenna
Radar	32-line LiDAR	1	32-line, 600,000 points per second, 200m range, 36mm beam width, 90° scanning
Camera	Stereo Camera	1	Stereo Camera, 1280×720 resolution, 50° field of view, 4m range

Following algorithm deployment and hardware selection, this study constructed a mobile SLAM verification platform for multi-scenario testing. The experimental platform adopts a four-wheel independent driving moving chassis with a maximum speed of 1.5 m/s and steering accuracy of 0.5°, and is equipped with the aforementioned system to constitute a real-time mapping system. Figure 11 shows the typical campus scene test, while Figure 12 presents the constructed point cloud map, which clearly captures the geometric features of dynamic obstacles such as garbage cans in Figure 12(part a) and shared bicycles in Figure 12(part b).

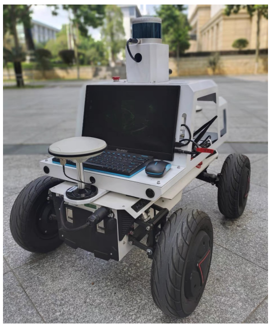


Figure 11. Mobile robot used in experimental testing

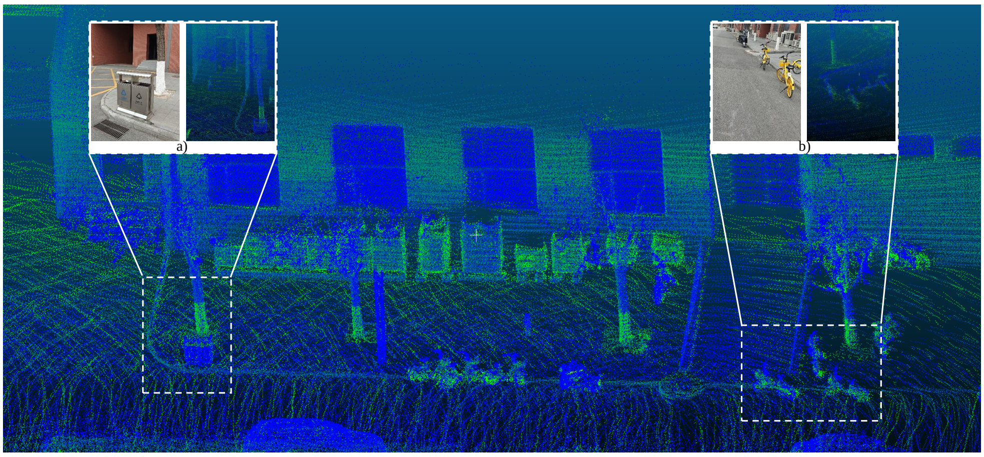


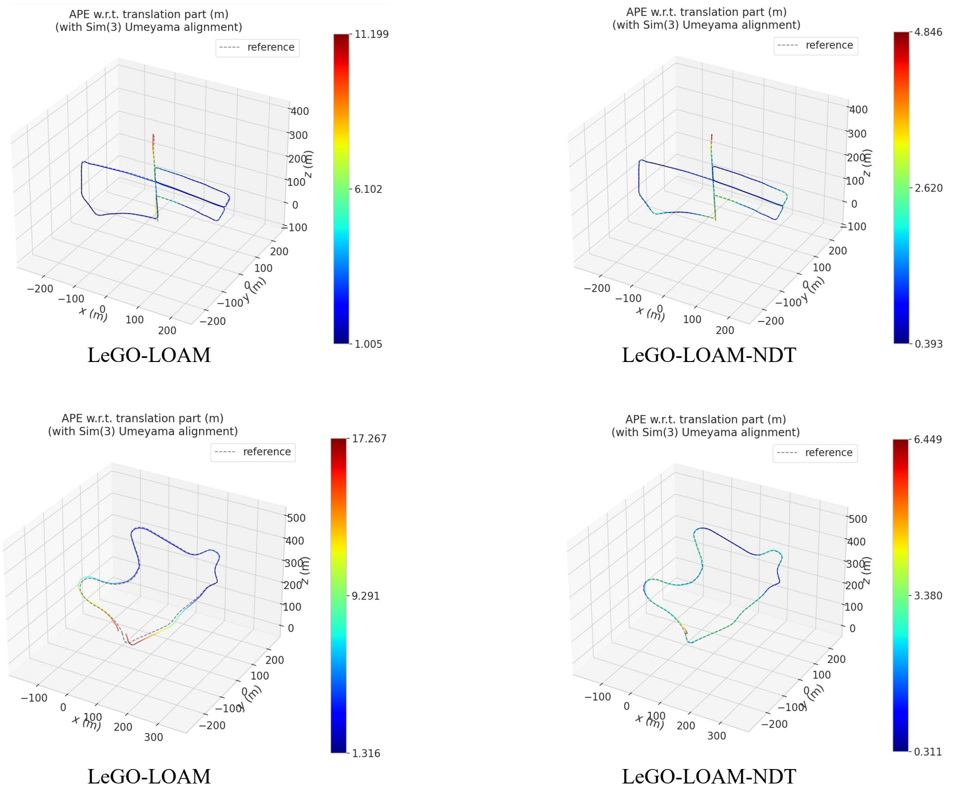
Figure 12. Mapping example in a real-world complex scenario

Figures 13 and 14 illustrate that the robot's accumulated error fluctuates during its movement. The LeGO-LOAM algorithm exhibits a larger error at corners and after long-distance operation, as evidenced by the varying color distribution in the figures, with errors approximately around 15.1 and 4.6, respectively. However, following optimization, the algorithm's stability significantly improves, with the error values predominantly represented by a low-error blue color, remaining within the range of 0.1 to 2.6.

As shown in Tables 3 and 4, the baseline LeGO-LOAM algorithm exhibits a noticeable degradation in positioning accuracy over extended operational periods. Specifically, the APE demonstrates significant fluctuations, particularly in maneuvering regions such as turns, while the RPE reveals an accumulation of error correlated with continuous frame-to-frame motion, especially prominent in long-term displacement scenarios. The proposed optimized algorithm effectively mitigates these issues, reducing the maximum APE by approximately 2.067 meters compared to the baseline. Moreover, the optimized method consistently achieves lower mean and RMSE across both experimental scenarios. The reduction in standard deviation further indicates improved stability and reliability of the localization performance. These results collectively demonstrate the enhanced robustness and accuracy of the improved LeGO-LOAM-NDT framework under dynamic and extended operational conditions.

This study conducts comparative analysis of baseline LeGO-LOAM and the enhanced algorithm in APE/RPE performance metrics. As demonstrated in Figures 15 and 16, the optimized algorithm achieves substantial improvements in both accuracy and temporal stability.

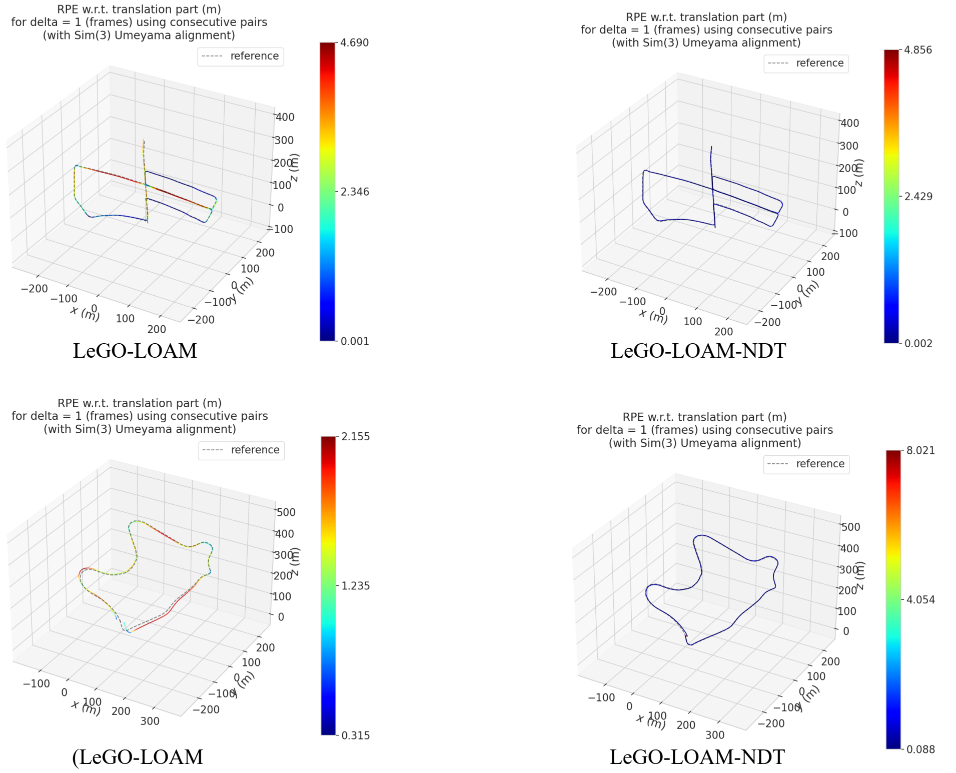
As demonstrated in Figure 13, the LeGO-LOAM-NDT algorithm exhibits significant improvements in APE performance across two distinct test environments. The enhanced algorithm demonstrates a 1.504-meter improvement in Environment 1 and 4.45-meter improvement in Environment 2 compared to the baseline LeGO-LOAM algorithm. Notably, during extended operational periods, the positioning error demonstrates a 92.87% reduction from baseline levels, effectively mitigating cumulative error propagation through prolonged system operation.



LeGO-LOAM LeGO-LOAM-NDT

LeGO-LOAM LeGO-LOAM-NDT

Figure 13. Absolute pose error comparison results



LeGO-LOAM LeGO-LOAM-NDT

(LeGO-LOAM LeGO-LOAM-NDT

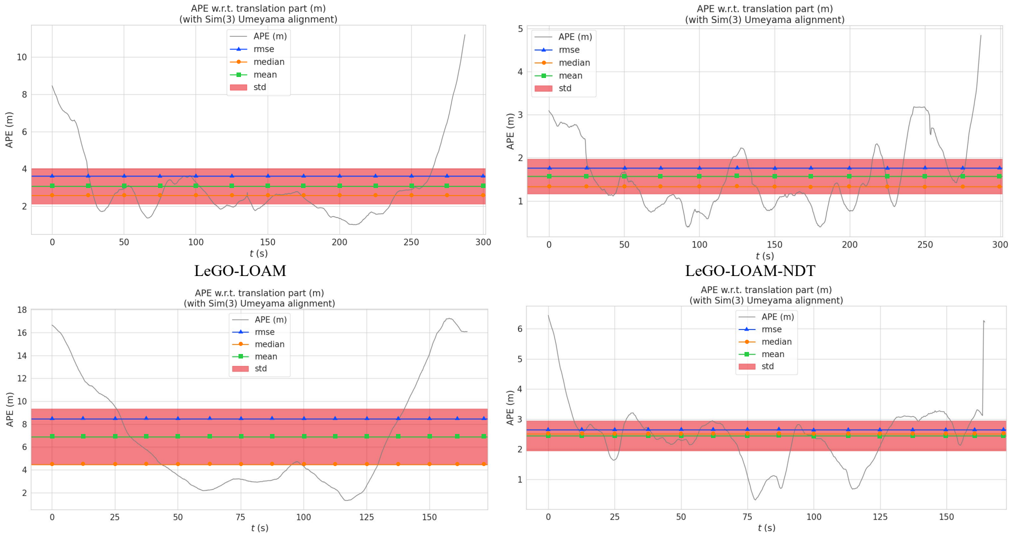
Figure 14. Relative pose error comparison results

Table 3. APE data in experiment scenarios 01 and 02

Environment	Method	Max	Mean	Median	Min	RMSE	SSE	STD
1	Lego-loam	11.199	3.073	2.582	1.005	3.621	18095.342	1.915
1	Ours	4.846	1.569	1.338	0.393	1.761	2140.837	0.800
2	Lego-loam	17.267	6.895	4.475	1.316	8.550	113864.483	4.901
2	Ours	6.449	2.445	2.510	0.311	2.643	2773.265	1.004

Table 4. RPE data in experiment scenarios 01 and 02

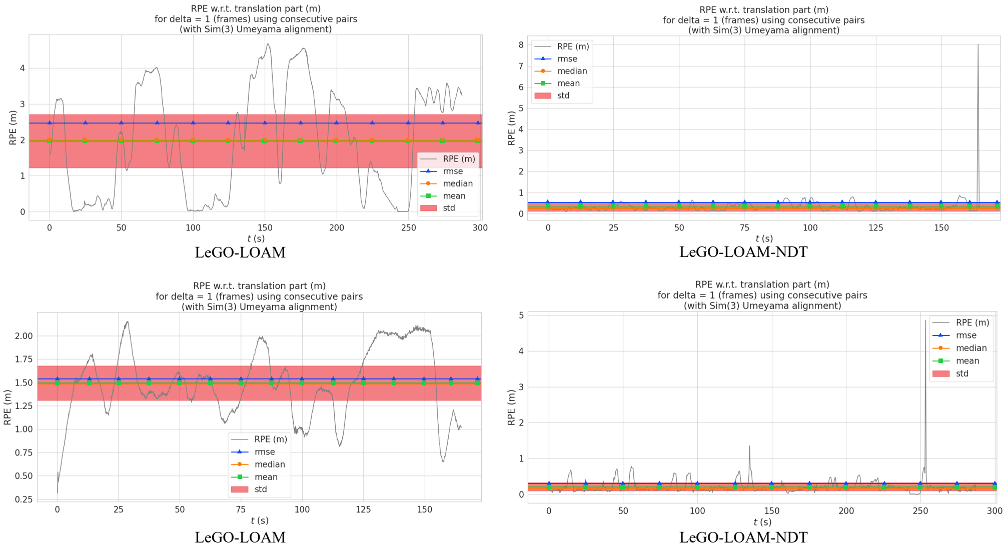
Environment	Method	Max	Mean	Median	Min	RMSE	SSE	STD
1	Lego-loam	4.690	1.971	1.985	0.001	2.469	8405.730	1.487
1	Ours	4.856	0.212	0.163	0.002	0.313	67.597	0.230
2	Lego-loam	2.155	1.494	1.484	0.315	1.538	3763.014	0.368
2	Ours	8.021	0.322	0.246	0.088	0.529	110.993	0.420



LeGO-LOAM LeGO-LOAM-NDT

LeGO-LOAM LeGO-LOAM-NDT

Figure 15. Comparison of APE in two complex real-world environments



LeGO-LOAM LeGO-LOAM-NDT

LeGO-LOAM LeGO-LOAM-NDT

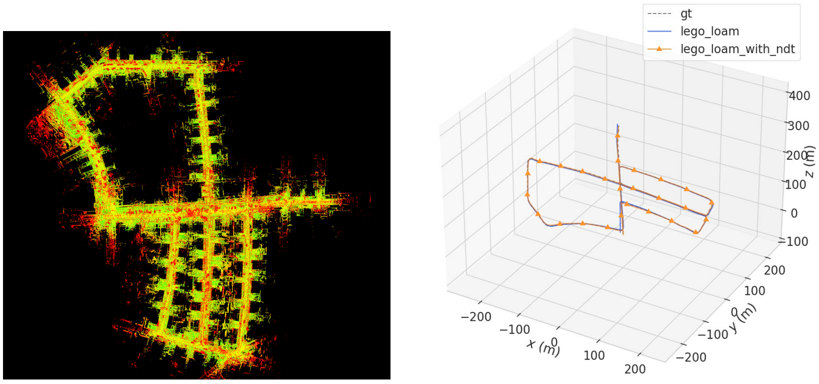
Figure 16. Comparison of RPE in two complex real-world environments

Regarding RPE metrics shown in Figure 16, the LeGO-LOAM-NDT algorithm achieves substantial error propagation control. While the baseline algorithm (LeGO-LOAM) exhibits RPE fluctuations ranging from 2.0 to 5.0 meters during the 50-150 second test interval with peak errors exceeding 4.7 meters, the LeGO-LOAM-NDT algorithm confines RPE variations within 1.5-2.0 meters. Specifically, at the critical 150-second mark, the LeGO-LOAM+NDT algorithm maintains RPE values at approximately 0.2 meters, contrasting with the baseline system's 1.5-2.0 meter error range during this period. Furthermore, the LeGO-LOAM-NDT algorithm consistently sustains RPE values within 0-1.0 meters across all other measurement intervals.

As shown in Figures 17 and 18, the mobile robot equipped with a LiDAR sensor in this study achieved high-precision localization and 3D mapping in two different environments, as presented in Figures 17(a) and 18(a). By comparing the ground truth (gt) trajectories with the estimated trajectories and pose results before and after optimization in Figures 17(b) and 18(b), the results demonstrate that the algorithm enhanced with NDT optimization significantly improves both localization accuracy and pose estimation stability. Meanwhile, the constructed point cloud maps exhibit high clarity and completeness, effectively reflecting detailed structural features of the environment. These experimental results fully validate the robustness and reliability of the LiDAR-based point cloud processing algorithm in complex environments, providing strong support for autonomous navigation and environmental perception of mobile robots.

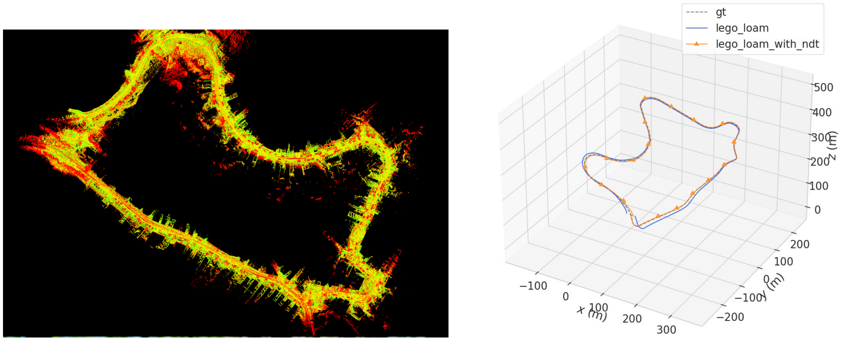
The enhanced performance of the proposed algorithm stems from the synergistic integration of multiple key components. The LeGO-LOAM feature extraction module facilitates efficient and robust initial pose estimation through the extraction of edge and planar features, supporting real-time processing. Nonetheless, its accuracy may degrade in environments characterized by low feature density or dynamic elements. To mitigate this limitation, the NDT-based weighted fusion module refines the initial pose by probabilistically modeling point cloud distributions and employing Newton-optimized alignment, thereby significantly improving registration accuracy and robustness to dynamic disturbances. Moreover, the adaptive thresholding mechanism dynamically modulates matching constraints to accommodate environmental variability and effectively suppress outliers induced by moving obstacles. Collectively, these components enhance localization accuracy and map consistency while maintaining real-time performance, as demonstrated through extensive evaluations on public datasets and real-world scenarios.

As shown in Figures 19 and 20, the integration of NDT demonstrates significant improvements in both positional accuracy (x, y, z) coordinates and angular stability (roll, pitch, yaw) across all movement phases, confirming its effectiveness in enhancing robotic positioning precision during operational mobility. Comparative analysis reveals that the NDT-enhanced system substantially reduces positional errors and diminishes angular fluctuations across multiple spatial dimensions. Notably, during critical time intervals (50s, 100s, and 250s), the LeGO-LOAM-NDT variant exhibits enhanced positional stability with diminished angular fluctuations compared to the baseline LeGO-LOAM implementation, particularly in directional error mitigation and dynamic orientation preservation. Although, as shown in Table 5, there is a slight increase in memory usage, this minor overhead does not affect the overall algorithm performance while maintaining efficient operation.



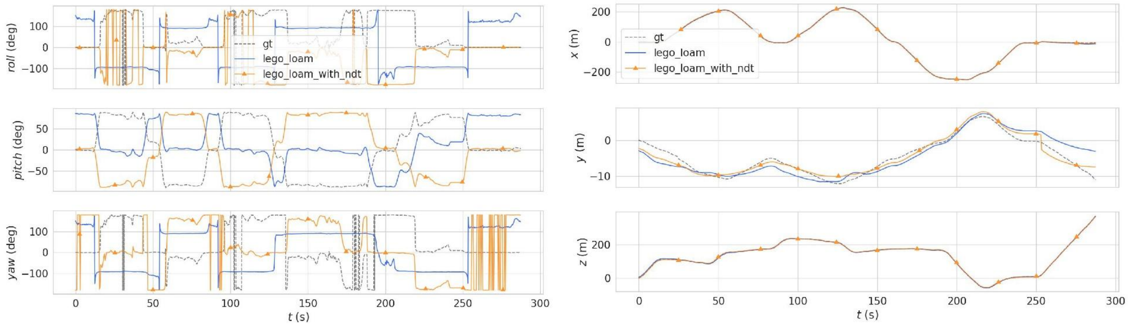
(a) (b)

Figure 17. Experimental scenario 01 trajectory error comparison: (a) mapping example and (b) comparison with ground truth



(a) (b)

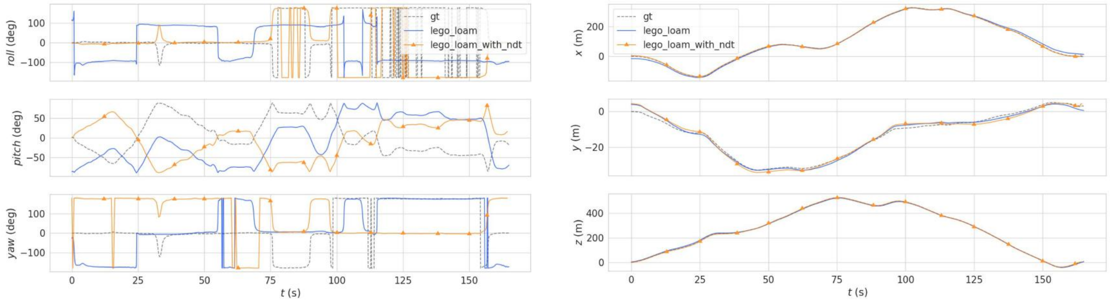
Figure 18. Experimental scenario 02 trajectory error comparison: (a) mapping example and (b) comparison with ground truth



The global position of the robot

Global error analysis of robot

Figure 19. Pose comparison in experimental scene 01



The global position of the robot

Global error analysis of robot

Figure 20. Pose comparison in experimental scene 02

Table 5. Real-time hardware performance indicators of mobile robots

Metric	LeGO-LOAM	LeGO-LOAM-NDT	Improvement ratio
Average CPU utilization	45%	58%	28.90%
Average GPU utilization	10%	22%	120%
Memory usage (GB)	4.8	5.5	14.60%
Algorithm processing time (FPS)	15 FPS	12 FPS	-20%
Feature extraction time per frame	50 ms	52 ms	4%
NDT optimization time	-	18 ms	-

5. CONCLUSION AND FUTURE WORK

This study investigates LiDAR SLAM techniques for mobile robots operating in dynamic outdoor environments. To address the stringent requirements for high-precision positioning during robotic operations, an innovative solution integrating enhanced NDT algorithms with multi-dimensional optimization strategies is proposed. Experimental results show that the proposed method has better performance in complex campus environment. Compared to the mainstream mobile SLAM framework LeGO-LOAM-NDT, the enhanced algorithm exhibits significant improvements in critical pose metrics (APE/RPE): i) The absolute posture error of the robot's 3D trajectory increased by about 51.36%, and the relative posture error increased by about 87.31%; ii) At path turning points (marked at 50 m/100 m/250 m intervals), the inherent drift characteristics of laser SLAM systems are effectively suppressed, reducing trajectory deviation to 67.34% of baseline values; iii) The system demonstrates superior stability across key metrics including 3D trajectory alignment, pose dynamics, and angular measurements, with average deviation from ground truth controlled within 2.64 meters.

While the improved algorithm achieves breakthroughs in complex environment positioning accuracy, practical tests reveal relatively lower operational efficiency compared to existing systems. However, with continuous advancements in computing hardware, the impact of increased computational load is becoming progressively less significant, making the approach more feasible for real-world deployment. The precise real-time mapping of LiDAR SLAM enhances the ability of autonomous systems to navigate safely, directly benefiting applications that impact daily life, such as self-driving cars and urban robots. Despite the improvements, the proposed method still exhibits performance degradation in extremely sparse environments or under high-speed maneuvers. In particular, when navigating open areas such as parking lots, the lack of prominent structural features results in occasional NDT misalignment. Moreover, in scenes with dense dynamic obstacles, voxel distortion remains a challenge, occasionally leading to transient localization drift. Since the use scenarios of a single lidar sensor may not be rich enough at present, in future work we should focus on studying sensor fusion technology, combining the excellent performance of different sensors to achieve more accurate robot positioning. The proposed method demonstrates strong potential for applications in autonomous driving, search-and-rescue robotics, infrastructure inspection, and urban patrol systems, contributing to safer, more intelligent, and efficient robotic operations in modern society.

FUNDING INFORMATION

The Science Foundation of Wuhan Institute of Technology (K2023056). Wuhan Special Fund for Science and Technology Innovation under Grant 2023100402010737.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Peiyan Yang	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	
Jiuyang Yu		✓		✓	✓	✓				✓				
Pan Liu		✓			✓		✓			✓				
Wenfeng Xia		✓			✓					✓	✓			
Yaonan Dai	✓				✓					✓		✓		✓

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal Analysis

I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting

Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject Administration
Fu : **Fu**nding Acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author, [PY], upon reasonable request.

REFERENCES

[1] A. Yarovoi and Y. K. Cho, “Review of simultaneous localization and mapping (SLAM) for construction robotics applications,” *Automation in Construction*, vol. 162, p. 105344, 2024, doi: 10.1016/j.autcon.2024.105344.

[2] Z. Fan, L. Zhang, X. Wang, Y. Shen, and F. Deng, “LiDAR, IMU, and camera fusion for simultaneous localization and mapping: A systematic review,” *Artificial Intelligence Review*, vol. 58, no. 6, art. no. 174, Jun. 2025, doi: 10.1007/s10462-025-11187-w.

[3] W. Wang, “Monocular and binocular visual-inertial system initialization and real-time dense 3D mapping,” Ph.D. dissertation, Department of Computer Science, Stevens Institute of Technology, Hoboken, NJ, USA, 2024.

[4] J. de Heuvel, X. Zeng, W. Shi, T. Sethuraman, and M. Bennewitz, “Spatiotemporal attention enhances LiDAR-based robot navigation in dynamic environments,” *IEEE Robotics and Automation Letters*, vol. 9, no. 5, pp. 4880–4887, May 2024, doi: 10.1109/LRA.2024.3353492.

[5] G. Liu *et al.,* “Innovations and refinements in LiDAR odometry and mapping: A comprehensive review,” *IEEE/CAA Journal of Automatica Sinica*, vol. 12, no. 6, pp. 1072–1094, Jun. 2025, doi: 10.1109/JAS.2025.125198.

[6] X. Yao *et al.,* “Autonomous navigation and adaptive path planning in dynamic greenhouse environments utilizing improved LeGO-LOAM and OpenPlanner algorithms,” *Journal of Field Robotics*, vol. 41, no. 7, pp. 2427–2440, 2024, doi: 10.1002/rob.22352.

[7] M. Hilger, N. Mandischer, and B. Corves, “RaNDT SLAM: Radar SLAM based on intensity-augmented normal distributions transform,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Abu Dhabi, UAE, 2024, pp. 7831–7838, doi: 10.1109/IROS58592.2024.10802458.

[8] J. Zhang and S. Singh, “LOAM: Lidar odometry and mapping in real-time,” in *Proceedings of Robotics: Science and Systems (RSS)*, Berkeley, CA, USA, 2014, vol. 2, pp. 1–9, doi: 10.15607/RSS.2014.X.007.

[9] T. Shan and B. Englot, “LeGO-LOAM: Lightweight and ground-optimized lidar odometry and mapping on variable terrain,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Madrid, Spain, 2018, pp. 4758–4765, doi: 10.1109/IROS.2018.8594299.

[10] J. Jiao *et al.,* “Greedy-based feature selection for efficient LiDAR SLAM,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Xi’an, China, 2021, pp. 5222–5228, doi: 10.1109/ICRA48506.2021.9561083.

[11] H. Wang, C. Wang, C.-L. Chen, and L. Xie, “F-LOAM: Fast lidar odometry and mapping,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Prague, Czech Republic, 2021, pp. 4390–4396, doi: 10.1109/IROS51168.2021.9635926.

[12] S. Jiang *et al.,* “Navigation system for orchard spraying robot based on 3D LiDAR SLAM with NDT-ICP point cloud registration,” *Computers and Electronics in Agriculture*, vol. 220, p. 108870, 2024, doi: 10.1016/j.compag.2024.108870.

[13] F. Wang *et al.,* “A method coupling NDT and VGICP for registering UAV-LiDAR and LiDAR-SLAM point clouds in plantation forest plots,” *Forests*, vol. 15, no. 12, p. 2186, 2024, doi: 10.3390/f15122186.

[14] Z. Wang and G. Liu, “Improved LeGO-LOAM method based on outlier points elimination,” *Measurement*, vol. 214, p. 112767, 2023, doi: 10.1016/j.measurement.2023.112767.

[15] D. Wei, H. Ran, Z. Li, and T. Gao, “Enhanced low drift LiDAR SLAM with dynamic noise using ground principal plane constraint,” *Measurement*, vol. 245, p. 116549, 2025, doi: 10.1016/j.measurement.2025.116549.

[16] Y. Wang *et al.,* “ROLO-SLAM: Rotation-optimized LiDAR-only SLAM in uneven terrain with ground vehicle,” *Journal of Field Robotics*, vol. 42, no. 3, pp. 880–902, 2025, doi: 10.1002/rob.22416.

[17] S. Hong *et al.,* “GS-LIVO: Real-time LiDAR, inertial, and visual multi-sensor fused odometry with Gaussian mapping,” *arXiv preprint arXiv:2501.08672*, 2025, doi: 10.48550/arXiv.2501.08672.

[18] X. Cheng, X. Liu, J. Li, and W. Zhou, “Deep learning-based point cloud registration: A comprehensive investigation,” *International Journal of Remote Sensing*, vol. 45, no. 10, pp. 3412–3442, 2024, doi: 10.1080/01431161.2024.2343210.

[19] M. McDermott and J. Rife, “Correcting motion distortion for lidar scan-to-map registration,” *IEEE Robotics and Automation Letters*, vol. 9, no. 2, pp. 1516–1523, Feb. 2024, doi: 10.1109/LRA.2023.3345330.

[20] K. Koide, J. Miura, and E. Menegatti, “A portable three-dimensional LIDAR-based system for long-term and wide-area people behavior measurement,” *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, 2019, doi: 10.1177/1729881419841532.

[21] T. Shan *et al.,* “LIO-SAM: Tightly-coupled lidar inertial odometry via smoothing and mapping,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, NV, USA, 2020, pp. 5135–5142, doi: 10.1109/IROS45743.2020.9341176.

[22] A. Reinke *et al.,* “LOCUS 2.0: Robust and computationally efficient lidar odometry for real-time 3D mapping,” *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9043–9050, Oct. 2022, doi: 10.1109/LRA.2022.3191176.

[23] B. Zhou, Y. Tu, Z. Jin, C. Xu, and H. Kong, “HPPLO-Net: Unsupervised lidar odometry using a hierarchical point-to-plane solver,” *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 2727–2739, Jan. 2024, doi: 10.1109/TIV.2023.3263777.

[24] J. Liu, G. Wang, C. Jiang, Z. Liu, and H. Wang, “TransLO: A window-based masked point transformer framework for large-scale LiDAR odometry,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 2, pp. 1683–1691, 2023, doi: 10.1609/aaai.v37i2.25257.

[25] J. Han, R. Dong, and J. Kan, “BASL-AD SLAM: A robust deep-learning feature-based visual SLAM system with adaptive motion model,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 8, pp. 9568–9581, 2024, doi: 10.1109/TITS.2024.3361234.

[26] Y. Pan *et al.,* “PIN-SLAM: LiDAR SLAM using a point-based implicit neural representation for achieving global map consistency,” *IEEE Transactions on Robotics*, vol. 40, pp. 1928–1948, 2024, doi: 10.1109/TRO.2024.3356891.

[27] S. Chen *et al.,* “NDT-LOAM: A real-time lidar odometry and mapping with weighted NDT and LFA,” *IEEE Sensors Journal*, vol. 22, no. 4, pp. 3660–3671, Feb. 2022, doi: 10.1109/JSEN.2021.3130982.

[28] J. Wei, M. Song, and Y. Yuan, “High precision navigation and positioning for multisource sensors based on bibliometric and contextual analysis,” *Remote Sensing*, vol. 17, no. 7, p. 1136, 2025, doi: 10.3390/rs17071136.

[29] W. Zheng *et al.*, “UP-SLAM: Adaptively structured Gaussian SLAM with uncertainty prediction in dynamic environments,” *arXiv preprint arXiv:2505.22335*, 2025.

[30] P. Yin *et al.*, “General place recognition survey: Towards real-world autonomy,” *IEEE Transactions on Robotics*, vol. 41, pp. 3019–3038, 2025, doi: 10.1109/TRO.2025.3550771.

BIOGRAPHIES OF AUTHORS

Peiyan Yang is a master’s student at Wuhan Institute of Technology. His main research areas are autonomous driving technology, path planning, and multi-sensor fusion. He has won many awards in academic competitions, including two national honors and one provincial honor. His contact email address is 18107139166@163.com.

Jiuyang Yu is a second-level professor/doctoral supervisor, an expert with outstanding contributions in Hubei Province, an advanced individual in national chemical science and technology, a member of the National Teaching Steering Committee for the ”Process Equipment and Control Engineering” major of the Ministry of Education, a member of the Chemical Machinery Professional Committee of the Chinese Chemical Society, the person in charge of the national characteristic major ”Process Equipment and Control Engineering”, the person in charge of the provincial quality course ”Pressure Vessels and Process Equipment”, a famous teacher of Wuhan institute of Technology, and the person in charge of the ”Process Equipment and Control Engineering Professional Teaching Team” of Hubei Province. His contact email address is yjy@wit.edu.cn

Pan Liu is a master’s student at Wuhan Institute of Technology. His main research areas are computer image processing and weld defect detection. His contact email address is 3060480194@qq.com.

Wenfeng Xia is a doctoral student at Wuhan Institute of Technology and has successfully published two papers. His research areas include deep learning, image processing, high-temperature structural integrity principles and online monitoring technology. Her email contact address is 865388537@qq.com.

Yaonan Dai has a PhD degree and is an associate professor at the School of Mechanical and Electrical Engineering of Wuhan institute of Technology. His main research directions are oil and gas pipeline robots, artificial intelligence and high-temperature structural integrity. In recent years, he has presided over a 2023 Hubei Natural Science Foundation project. As the applicant and technical leader, he has been responsible for the ”2018 High-end CNC Machine Tools and Basic Manufacturing Equipment Science and Technology Major Project” and the ”2020 Hubei Provincial Key R&D Plan Project”, and participated in 5 related projects such as aerospace equipment, high-temperature and high-efficiency petrochemical equipment, and high-temperature nuclear power equipment. His email contact information is dyn1121758919@163.com.