

Article

LiDAR-Based Sensor Fusion SLAM and Localization for Autonomous Driving Vehicles in Complex Scenarios

Kai Dai ¹, Bohua Sun ^{1,*} , Guanpu Wu ¹, Shuai Zhao ², Fangwu Ma ¹ , Yufei Zhang ¹ and Jian Wu ¹¹ State Key Laboratory of Automotive Simulation and Control, Jilin University, Changchun 130025, China² Automotive Data Center, CATARC, Tianjin 300000, China

* Correspondence: bohua.sun@jlu.edu.cn

Abstract: LiDAR-based simultaneous localization and mapping (SLAM) and online localization methods are widely used in autonomous driving, and are key parts of intelligent vehicles. However, current SLAM algorithms have limitations in map drift and localization algorithms based on a single sensor have poor adaptability to complex scenarios. A SLAM and online localization method based on multi-sensor fusion is proposed and integrated into a general framework in this paper. In the mapping process, constraints consisting of normal distributions transform (NDT) registration, loop closure detection and real time kinematic (RTK) global navigation satellite system (GNSS) position for the front-end and the pose graph optimization algorithm for the back-end, which are applied to achieve an optimized map without drift. In the localization process, the error state Kalman filter (ESKF) fuses LiDAR-based localization position and vehicle states to realize more robust and precise localization. The open-source KITTI dataset and field tests are used to test the proposed method. The method effectiveness shown in the test results achieves 5–10 cm mapping accuracy and 20–30 cm localization accuracy, and it realizes online autonomous driving in complex scenarios.

Keywords: LiDAR SLAM; autonomous vehicle; localization; multi-sensor fusion



Citation: Dai, K.; Sun, B.; Wu, G.; Zhao, S.; Ma, F.; Zhang, Y.; Wu, J. LiDAR-Based Sensor Fusion SLAM and Localization for Autonomous Driving Vehicles in Complex Scenarios. *J. Imaging* **2023**, *9*, 52. <https://doi.org/10.3390/jimaging9020052>

Academic Editor: Pierre Gouton

Received: 17 November 2022

Revised: 21 January 2023

Accepted: 13 February 2023

Published: 20 February 2023



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

1. Introduction

With the development of intelligent and connected vehicle technology, the intelligent transportation system with autonomous driving passenger cars, commercial vehicles and taxis has undergone tremendous changes in the perception of the complex scenarios. Vehicle localization is the key issue that should be solved in autonomous driving and how to realize high-precise vehicle localization under the condition of unavailable satellites or unstructured roads is one of the technical problems to be solved urgently. The localization technique based on the vision [1] and satellites observations can achieve centimeter-level localization but heavily rely on satellite signals, traffic signs, and initialization [2]. LiDAR-based localization techniques are largely invariant to illumination and satellite signal changes. Therefore, high precision maps with denser point clouds are required, and the map-based multi-sensor fusion localization should be widely used to cover different driving conditions [3].

LiDAR SLAM is widely used in the construction of 3D point cloud maps [4]. The architecture of a simultaneous localization and mapping (SLAM) system consists of the front-end and the back-end. The front-end seeks to interpret the sensor data to obtain constraints as the basis for optimization approaches, such as point cloud registration, loop closure detection, or Global Navigation Satellite System (GNSS) pose. The back-end focuses on computing the best map result based on optimization techniques with the given constraints [5]. Many registration methods have been proposed for the front-end, such as the iterative closest point (ICP) [6], normal distribution transformation (NDT) [7], and feature-based [8]. However, typical registration methods suffer from drift in large-scale tests, due to the poor performance in the loop closure detection and the position correction

with absolute measurements. The back-end optimization process can reduce the drift based on the typical back-end algorithms, such as the early-used extended Kalman filter (EKF) [9] or the current commonly pose graph optimization [10]. Besides the accuracy and efficiency performance advantages, the back-end optimization process provides a framework that is more amenable to analysis as well.

Multi-sensor fusion localization for autonomous vehicles is mainly based on the GNSS, inertial measurement unit (IMU), camera, LiDAR, and vehicle states [11,12]. LiDAR-based methods can provide precise localization under the condition of weak satellite signals [13]. However, global localization and environmental degradation are still important issues for LiDAR-based methods. Complementary sensor fusion is an effective method to solve these issues. LiDAR shows good performance in scenarios with full 3D or texture features, real time kinematic (RTK) GNSS provides a precise absolute position, and IMU and vehicle states provide the position and orientation of the vehicle getting rid of the external scenarios.

Taking the above-mentioned into consideration, LiDAR-based SLAM and localization still have problems to be solved. A SLAM and localization method based on multi-sensor fusion is proposed and integrated into a general framework in this paper, to solve the map drift and localization failure and meet the demand of high-precision localization under the condition of unavailable satellites, extreme climate, or road structure changes. A pose graph considering the loop closure and RTK-GNSS position is used to optimize the map. The LiDAR-based localization result and vehicle states are integrated into an error state Kalman filter (ESKF) to obtain robust and precise localization.

Figure 1 is the framework of this article. In the process of offline mapping, a pose-graph optimization LiDAR SLAM is proposed based on NDT registration, loop closure detection and RTK-GNSS position constraints to generate an optimized 3D point cloud map. In the online localization process, the inertial navigation system (INS) is used as a prediction model in the Kalman filter propagation phase, LiDAR localization and vehicle velocity are used by an error-state Kalman filter as the measurements.

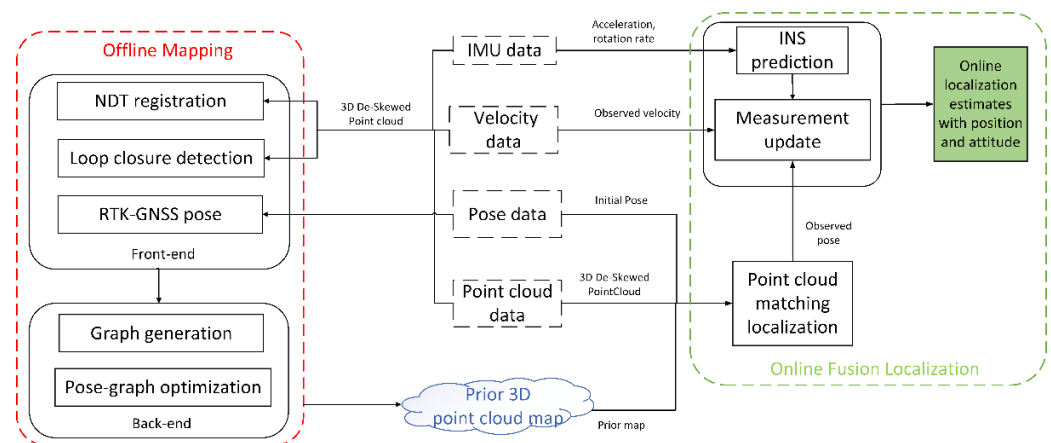


Figure 1. The framework that estimates the optimal position and attitude (PA) for the autonomous vehicle by combining offline mapping and online localization. NDT: normal distributions transform, RTK-GNSS: real time kinematic-global navigation satellite system, IMU: inertial measurement unit, INS: inertial navigation system.

The main contributions of this paper are summarized as follows.

- The NDT registration, scan context-based loop closure detection and RTK-GNSS are integrated into a LiDAR SLAM framework and innovative use of pose graph to combine multiple methods to optimize position and reduce map drift.
- LiDAR matching localization position and vehicle states are fused by ESKF, which takes full advantage of the vehicle velocity constraints of ground autonomous vehicles to optimize localization results and provide robust and accurate localization results.

- A general framework with mapping and localization is proposed, which is tested on the KITTI dataset [14] and real scenarios. Results demonstrate the effectiveness of the proposed framework.

The rest of the paper is structured as follows. The related work about mapping and localization is presented in Section 2. The offline mapping process is introduced in Section 3 and the online localization method introduced in Section 4. The experiment evaluation is given in Section 5. The discussion is given in Section 6. Finally, the conclusion and future work are presented in Section 7.

2. Related Work

In this section, a brief overview of algorithms related to LiDAR SLAM and multi-sensor fusion localization methods are introduced, including the point cloud registration algorithms, loop closure detection algorithms, pose graph algorithms, filter-based sensor fusion algorithms, and their interaction.

With the development of LiDAR SLAM, various registration algorithms have been proposed. The ICP algorithm is widely used in the registration of point cloud. Due to the improvement of computational efficiency and accuracy requirements, a variety of variant ICP algorithms have been derived [15]. However, the ICP is very sensitive to the initial guess. Different from the ICP, the NDT registration algorithm builds a statistical probability model of the point cloud, which is more efficient and accurate. Study [16] proposed a 3D-NDT registration algorithm as the improvement of the 2D-NDT algorithm [17] and compares qualitatively and quantitatively with the standard ICP algorithm. Results show that the method is faster and more reliable. Study [7] proposed an NDT-based SLAM method, which can achieve long-range high-precision map establishment and localization in dynamic scenarios. Li et al. [18] improved the accuracy of stereo visual SLAM by removing dynamic obstacles. Wen et al. [19] compared the performance of NDT-based graph optimization SLAM under complex urban conditions; the results show that the performance of the NDT SLAM algorithm is positively related to the traffic environment.

Loop closure is essential for correcting drift error and building a globally consistent map [5]. Visual-based loop closure detection is often limited by illumination variance and environment changes. The early LiDAR-based methods for place recognition focus on descriptors from structural information [20]. However, the descriptor method is limited by rotational invariance and poor point cloud resolution. Study [21] proposed a histogram method to address these problems but still causes false recognition. To address the aforementioned issues, studies [22,23] proposed the scan context method, which proposed a more efficient bin encoding function for place recognition and is widely used in LiDAR SLAM currently; in addition, the loop closure detection method based on deep learning has also been gradually applied to SLAM [24].

Graph-based optimization [25], which optimizes the full trajectory and map of the vehicle from the full set of measurements, has received attention in many studies in recent years. Some general frameworks and open-source implementation of a pose-graph method are proposed by [26,27]. Study [28] proposed a tutorial for the reader to implement graph-based SLAM. To improve the robustness of pose-graph SLAM, study [29] proposed a novel formulation that allows the back-end to change parts of the topological structure of the graph during the optimization process and progress experiments by loop closure constraints. To obtain accurate positions for mapping in large-scale environment, study [30] proposed global positioning system (GPS) and LiDAR odometry (GLO)-SLAM, which uses LiDAR to verify the reliability of GNSS, and the LiDAR odometry also will be optimized by means of reliable GPS data. In addition, study [31] added IMU/odometry pre-integration results under the framework of graph optimization, which effectively reduced navigation drift. With the development of deep learning, related technologies have also been applied to the field of SLAM [32,33].

The multi-sensor fusion method is usually used in SLAM and localization areas. Fusing multiple sensors and making the whole system accurate, robust, and applicable

for various scenes is a challenging task. Study [34] integrated 2D LiDAR/IMU/GPS into a localization system for urban and indoor scenarios, IMU and RTK-GNSS for full scene localization, and vehicle velocity is good complementary information for localization, especially in satellites denied and environmental degradation conditions.

3. The Offline Mapping

The online LiDAR localization module relies on a pre-build map. The offline mapping aims to obtain a 3D point cloud map representation of the scenario. The NDT-based point cloud registration and scan context-based loop closure detection are innovatively combined into the front-end and the pose-graph is used in the back-end to optimize the map.

3.1. LiDAR SLAM Front-End

3.1.1. NDT Based Registration

Comparing with the ICP algorithm, the NDT divides the point cloud space into cells and each cell is continuously modeled as a Gaussian distribution. Taking the better calculation efficiency and registration accuracy of NDT into account, the NDT is chosen as the point cloud registration method. The process of NDT can be expressed as follows.

In the point cloud, point sets X and Y are the consecutive frames, X is the frame at the previous moment, Y is the frame at the next moment:

$$X = \{x_1, x_2, \dots, x_n\} \quad (1)$$

$$Y = \{y_1, y_2, \dots, y_n\} \quad (2)$$

Assuming that the transformation between X, Y can be expressed as follow:

$$p = [T_x \ T_y \ T_z \ R_x \ R_y \ R_z]^T \quad (3)$$

where T is the translation vector, R is the rotation vector.

The mean of all points in X can be expressed as:

$$\mu = \frac{1}{N_x} \sum_{i=1}^{N_x} x_i \quad (4)$$

where N_x is the number of points in the X . The covariance of X can be expressed as follow:

$$\Sigma = \frac{1}{N_x - 1} \sum_{i=1}^{N_x} (x_i - \mu)(x_i - \mu)^T \quad (5)$$

Assuming that the transformation p makes point y_i transform to y_i' , the transformation process can be expressed as the followed:

$$y_i' = T(p, y_i) = Ry_i + T \quad (6)$$

After transformation, the point y_i' is in the same coordinate system as the target point set X , and its coincidence degree can be expressed as a Gaussian distribution:

$$f(X, y_i') = f(X, T(p, y_i)) = \frac{1}{\sqrt{2\pi} \sqrt{|\Sigma|}} \exp\left(-\frac{(y_i' - \mu)^T \Sigma^{-1} (y_i' - \mu)}{2}\right) \quad (7)$$

The joint probability distribution of Y and X can be expressed as follows:

$$\psi = \prod_{i=1}^{N_y} f(X, T(p, y_i)) \quad (8)$$

where N_y is the number of points in the Y .

The objective function can be expressed as follow:

$$\max \psi = \max \prod_{i=1}^{N_y} f(X, T(p, y_i)) \quad (9)$$

Therefore, the maximize of the joint probability ψ means that the transformation has the highest degree of coincidence and the optimization variables T and R represent the translation and rotation between two consecutive frames, respectively.

3.1.2. Scan Context Based Loop Closure Detection

Comparing with the feature descriptors of the environment, few studies focus on the structural information to describe scenes. Scan context proposes a non-histogram method of global descriptors, which directly records the 3D structure of the visible space and can be deployed in LiDAR-based place recognition effectively. The lightweight and efficient encoding method, which can improve the accuracy of loop closure detection, is conducive to storing point cloud information. The scan context method is applied for the offline mapping process. Firstly, scan context is used to detect the loop closure frame. After detecting the candidate frame in the historical frame, NDT is used to register the loop closure frame with the current point cloud frame to obtain the precise loop pose.

Figure 2 shows the flow chart of scan context and loop closure detection. In the point cloud segmentation process, the point cloud space is cut into N_r rings along the increasing radius and the rings are cut into N_s sectors:

$$d_r = \frac{L_{\max}}{N_r} \quad (10)$$

where d_r represents the width of the ring, $L_{\max}$ represents the maximum measurement distance of LiDAR, N_r is numbers of rings.

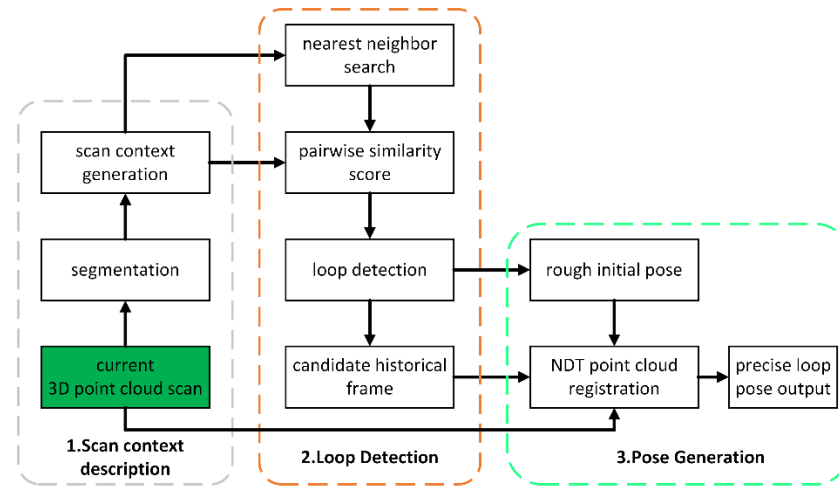


Figure 2. Scan context-based loop closure detection, current 3D point cloud scan is the start of this process. Which can provide loop closure detection position for offline mapping position optimization.

After segmentation, the segmented bin cells can be represented as a set P :

$$P = \bigcup_{i \in [N_r], j \in [N_s]} p_{ij} \quad (11)$$

where p_{ij} represents the set of midpoints of the i^{th} circle segmentation unit of the j^{th} sector.

In the generation of scan context process, the scan context I is represented as a $N_r \times N_s$ matrix, each element in the matrix represents the maximum value of all 3D points in the z-direction.

The distance function between two frames of point cloud scan context is defined as:

$$d(I^q, I^c) = \frac{1}{N_s} \sum_{j=1}^{N_s} \left(1 - \frac{c_j^q \cdot c_j^c}{\|c_j^q\| \|c_j^c\|} \right) \quad (12)$$

which can be used for similarity score, where I^q is the current frame scan context, c_j^q is the j^{th} of I^q , I^c is the historical frame scan context, and c_j^c is the j^{th} of I^c .

To solve the problem that the current frame is rotated relative to the historical frame, the order of the column vectors in the scan context obtained at the current time is changed and causes a large-distance function between the two frames, the historical frame I^c is translated by column, which will obtain N_s scan contexts, calculate the distance from the scan context of the current frame in turn, and select the one with the smallest distance, which can be expressed as follows:

$$D(I^q, I^c) = \min_{n \in [N_s]} d(I^q, I_n^c) \quad (13)$$

The loop frame can be found by comparing the similarity of scan contexts between the current frame and the historical frame point cloud; when the distance function is less than a certain threshold, the historical frame is considered to be a loop frame.

In the precise position calculation process, scan context is used to calculate the rotation angle φ between the current frame and the loop frame to improve the calculation efficiency and accuracy of the NDT, and φ is used as the initial position for the NDT registration process:

$$n^* = \underset{n \in [N_s]}{\operatorname{argmin}} d(I^q, I_n^c) \quad (14)$$

$$\varphi = \frac{2\pi}{N_s} \cdot n^* \quad (15)$$

3.1.3. RTK-GNSS Based Localization

Real time kinematic localization is a satellite navigation technique used to enhance the precision of localization data derived from satellite-based navigation systems. RTK relies on a single reference station to provide real-time corrections for the rover providing up to centimeter-level accuracy. There are indeed many situations with severe multipath and signal blockage under urban buildings or in forests. A stable and precise position and attitude can be provided for autonomous vehicles by combining RTK-GNSS and IMU.

3.2. Back-End Optimization

After interframe association and submap matching, there are inevitable cumulative errors in the point cloud map. The method of pose-graph optimization is used to further eliminate the cumulative errors, and the loop closure position and RTK-GNSS position will be used as constraints, the back-end optimization step is summarized in Algorithm 1.

Algorithm 1. The process of back-end optimization**Input:**LiDAR odometry position x_i, x_j RTK-GNSS position z_i Loop closure position z_{ij}' **Output:**Optimized vehicle position x_{opt}

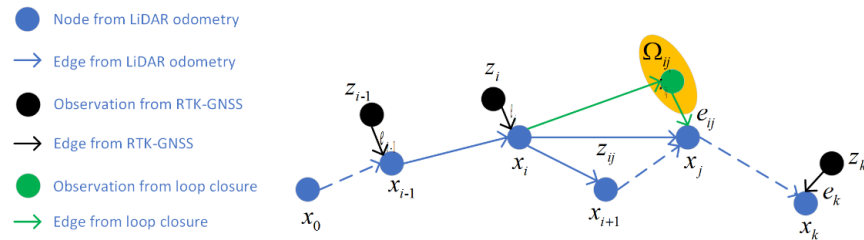
```

1: Trajectory alignment for  $x_i, z_i$  and  $z_{ij}'$ 
2: for each position  $x_i$  do
3:   if meet optimization cycle times  $h$  then
4:     execute optimization process:
5:      $x_{opt} = \arg \min F(x_i, x_j, z_i, z_{ij}')$ 
6:   else
7:     add RTK-GNSS position  $z_i$  constraint
8:     if loop closure position detected then
9:       add loop closure position  $z_{ij}'$  constraint
10:    end if
11:  end if
12: end for
13: return optimized vehicle position  $x_{opt}$ 

```

3.2.1. Graph Generation

The graph consists of edges and nodes, as shown in Figure 3; x_i represents nodes from LiDAR odometry, z_i represents prior position from RTK-GNSS, e_i represents the edge between x_i and z_i . z_{ij} represents the transformation of x_j and x_i , z_{ij}' represents expected observation from loop closure, and e_{ij} represents the edge between z_{ij} and z_{ij}' .

**Figure 3.** Graph structure for back-end optimization.**3.2.2. Graph Optimization**

Graph optimization takes all the constraints into a non-linear optimization problem, which will consider all the observation measurements:

$$F(x) = \sum_{i,j} e(x_i, x_j, \hat{z}_{ij})^T \Omega_{ij} e(x_i, x_j, \hat{z}_{ij}) \quad (16)$$

where $F(x)$ represents errors between all edges. Ω_{ij} is the matrix indicating the importance of each constraint in the global graph optimization. To adjust the state quantity x to minimize the value of the residual, it is necessary to obtain the Jacobian of the residual relative to state quantity, and then use the gradient descent method to optimize. The solution of this optimization is the x_{opt} which satisfying the following function:

$$x_{opt} = \operatorname{argmin} F(x) \quad (17)$$

To integrate the RTK-GNSS into the graph optimization, the error between LiDAR odometry x_i and RTK-GNSS position z_i can be represented as follows:

$$e_i = \ln(z_i^{-1} x_i)^V \quad (18)$$

The residual e_i after adding disturbance term to the x_i can be expressed as follows:

$$\hat{e}_i = \ln(z_i^{-1} \exp(\delta \tilde{\xi}_i^\wedge) x_i)^\vee \quad (19)$$

The error between z_{ij} and z_{ij}' can be represented as follows:

$$e_{ij} = \ln(z_{ij}^{-1} x_i^{-1} x_j)^\vee \quad (20)$$

The residual e_{ij} after adding disturbance to the x_i and x_j can be expressed as follows:

$$\hat{e}_{ij} = \ln(z_{ij}^{-1} x_i^{-1} \exp((- \delta \tilde{\xi}_i)^\wedge) \exp(\delta \tilde{\xi}_j^\wedge) x_j)^\vee \quad (21)$$

The residual is expanded after adding disturbance term, and the Jacobian matrix J of the residual with respect to the state quantity can be obtained.

A first-order Taylor expansion on the residuals can be expressed as follows:

$$e(x + \Delta x) \approx e + J\Delta x \quad (22)$$

$$F(x + \Delta x) \approx (e + J\Delta x)^T \Omega_{ij} (e + J\Delta x) \quad (23)$$

The state quantity x_{opt} after correction can be expressed as follows:

$$x_{opt} = x + \Delta x \quad (24)$$

4. The Online Localization

A multi-sensor fusion localization method based on the ESKF is proposed, which will estimate the vehicle position and attitude (PA) jointly by fusing vehicle states and LiDAR localization pose.

4.1. LiDAR Localization Based on 3D Point Cloud Map

The LiDAR localization based on a 3D point cloud map estimates the position of the vehicle in real-time, and the position can be used for the Kalman filter observation update. In this process, the RTK-GNSS position is used as the initial position for LiDAR localization to improve matching accuracy and efficiency. The NDT algorithm is used as registration method to match the real-time point cloud with the local map, the LiDAR localization step is summarized in Algorithm 2.

Algorithm 2. LiDAR localization in prior 3D point cloud map

Input:

RTK-GNSS position z_i

Point cloud p_i

Prior 3D point cloud global map M

Output:

LiDAR localization position x_{lidar}

- 1: Load 3D point cloud map M
 - 2: **if** get the initial position z_i **then**
 - 3: load local submap M_{sub} from global map M
 - 4: **if** need update submap M_{sub} **then**
 - 5: update submap M_{sub}
 - 6: **else**
 - 7: calculate position between p_i and M_{sub} :
 - 8: NDT registration $x_{lidar} = p_i \propto M_{sub}$
 - 9: **end if**
 - 10: **else**
 - 11: wait for initial position z_i
 - 12: **end if**
 - 13: **return** LiDAR localization position x_{lidar}
-

4.2. Filter State Equation

In the filter, the state variables error is expressed as follows:

$$X = [\delta P^T, \delta V^T, \phi^T, \varepsilon^T, \nabla^T]^T \quad (25)$$

where δP is the position error, δV is the velocity error, ϕ is the attitude error, ε is the gyroscope bias error, and ∇ is the accelerometer bias error. The state transition equation in continuous time can be expressed as follows:

$$X = F_t X + B_t W \quad (26)$$

According to the derivation of the IMU kinematics model, where

$$F_t = \begin{bmatrix} 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & F_{23} & 0_{3 \times 3} & C_b^n \\ 0_{3 \times 3} & 0_{3 \times 3} & F_{33} & C_b^n & 0_{3 \times 3} \\ & & 0_{3 \times 15} & & \\ & & 0_{3 \times 15} & & \end{bmatrix} \quad (27)$$

$$F_{23} = \begin{bmatrix} 0 & -f_U & -f_N \\ f_U & 0 & -f_E \\ -f_N & f_E & 0 \end{bmatrix} \quad (28)$$

The east-north-up (ENU) and right-forward-up (RFU) are chosen as the navigation reference frame n , and the body frame b , respectively, where f_E is the acceleration in the east direction, f_N is the acceleration in the north direction, f_U is the acceleration in the up direction, and C_b^n is the direction cosine matrix from b frame to n frame:

$$F_{33} = \begin{bmatrix} 0 & \omega \sin L & -\omega \cos L \\ -\omega \sin L & 0 & 0 \\ \omega \cos L & 0 & 0 \end{bmatrix} \quad (29)$$

where ω is the angular velocity of the earth's rotation, L is the latitude, W includes the gyroscope noise ω_g and accelerometer noise ω_a :

$$W = [w_{gx} \ w_{gy} \ w_{gz} \ w_{ax} \ w_{ay} \ w_{az}]^T \quad (30)$$

$$B_t = \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & C_b^n \\ -C_b^n & 0_{3 \times 3} \\ 0_{6 \times 3} & 0_{6 \times 3} \end{bmatrix} \quad (31)$$

4.3. Filter Measurement Update Equation

To compensate the loss of localization signal under complex driving scenarios and enhance the robustness of the localization system, LiDAR localization position and vehicle velocity are used as observation inputs:

$$Y = [\delta P_L^T \ \phi_L^T \ \delta V_v^T] \quad (32)$$

where δP_L is the LiDAR localization position error, ϕ_L is the attitude error, and δV_v is the vehicle velocity error.

The observation equation is as follows:

$$Y = G_t X + C_t N \quad (33)$$

where

$$G_t = \begin{bmatrix} I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 6} \\ 0_{3 \times 3} & 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 6} \\ 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 6} \end{bmatrix} \quad (34)$$

N is the observation noise and can be expressed as follows:

$$N = \begin{bmatrix} n_{p_L^E} & n_{p_L^N} & n_{p_L^U} & n_{\phi_L^E} & n_{\phi_L^N} & n_{\phi_L^U} & n_{V_v^E} & n_{V_v^N} & n_{V_v^U} \end{bmatrix}^T \quad (35)$$

$$C_t = \begin{bmatrix} I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix} \quad (36)$$

5. Experimental Verification and Performance Analysis

This section introduces experiments with the KITTI dataset and field tests based on the proposed method.

5.1. The Experiment Based on KITTI Dataset

The KITTI dataset was jointly founded by the Karlsruhe Institute of Technology in Germany and the Toyota American Institute of Technology. It is currently the world's largest autonomous driving localization and computer vision algorithm evaluation dataset. It contains LiDAR data, IMU data, RTK-GNSS data, velocity data, and the localization groundtruth, which is used to evaluate the mapping and localization accuracy. KITTI has multiple sequence datasets for various scenarios; sequence 00 was used in this study. The mapping and localization result is shown in Figure 4.

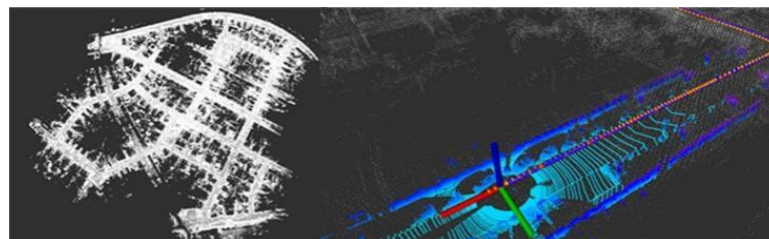


Figure 4. Optimized map and LiDAR localization result based on KITTI dataset.

5.1.1. Mapping Performance Analysis Based on KITTI Dataset

Back-end optimization plays an important role in the process of mapping; Figures 5 and 6 show the results of optimization performance, the abscissa in the figure represents the index of the data frame where the position is saved; in the following, the abscissa is represented by Index, which has the same meaning. In Figure 6, it can be seen that the optimized longitudinal, lateral and altitude error are reduced to centimeter-level, which effectively eliminates the cumulative error of the front-end odometry and improves the mapping accuracy.

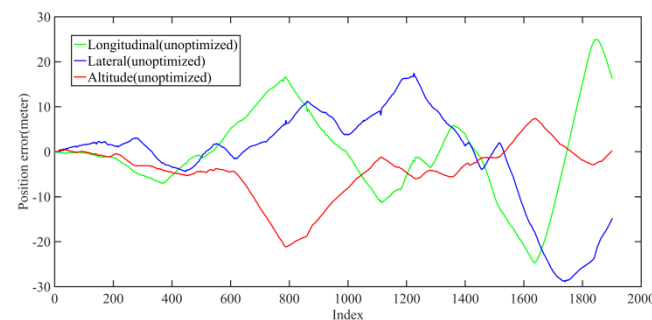


Figure 5. The unoptimized position error in longitudinal, lateral, and altitude directions of KITTI dataset, respectively.

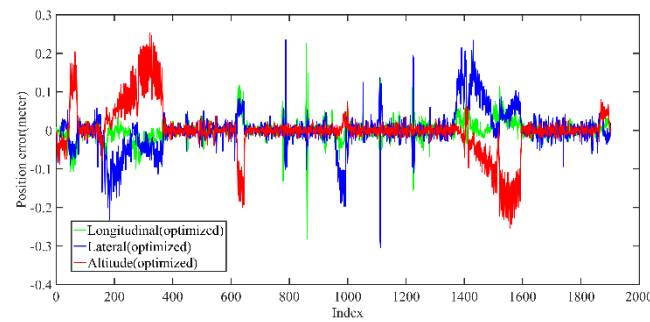


Figure 6. The optimized position error in longitudinal, lateral, and altitude directions of KITTI dataset, respectively; the error is significantly reduced after optimization.

Figure 7 shows the coincidence degree between the groundtruth estimated trajectory. It can be seen that there is a large deviation between the unoptimized trajectory and the groundtruth, the optimized trajectory error is significantly reduced. Table 1 shows the trajectory accuracy after optimization is significantly improved and basically coincides with the groundtruth, and the average error is about 10 cm.

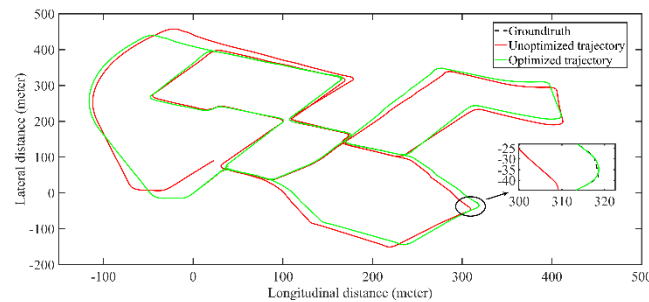


Figure 7. The comparison of trajectory before and after optimization with groundtruth.

Table 1. Mapping performance comparison before and after optimization [35] (units: meter).

	Max	Min	Mean	RMSE	STD
Before optimization	34.31	0.02	13.61	16.61	9.53
After optimization	0.23	0.01	0.11	0.13	0.09

5.1.2. Localization Performance Analysis Based on KITTI Dataset

The LiDAR localization result is fused with IMU and vehicle velocity to improve the localization accuracy in the case of scenario degradation. As shown in Figures 8 and 9, a localization performance test is conducted based on the prior KITTI point cloud map.



Figure 8. LiDAR localization based on prior KITTI point cloud map; the white points in the figure are the point cloud map built by the offline mapping process, the red line is the trajectory of LiDAR matching results, the orange line is groundtruth, and the blue line is the fused trajectory.

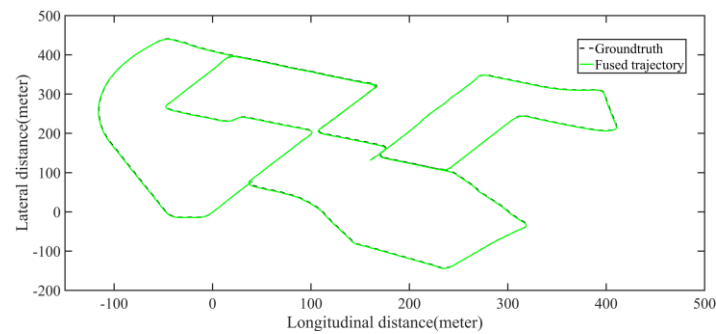


Figure 9. The comparison of fused trajectory and groundtruth. The fused localization data basically coincides with the groundtruth, indicating that the online localization results meet expectations.

As shown in Figure 10 and Table 2, the maximum position error on the KITTI dataset is within 35 cm, the average position error is within 20 cm, and a stable localization result is maintained.

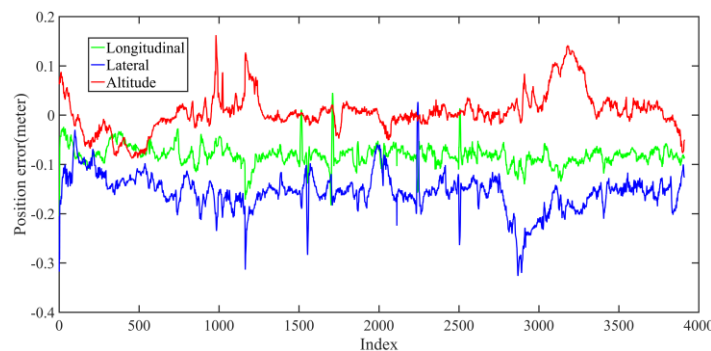


Figure 10. The position error of localization result in longitudinal, lateral, and altitude directions, respectively.

Table 2. Localization error compared with groundtruth (units: meter).

Max	Min	Mean	RMSE	STD
0.35	0.08	0.18	0.16	0.07

5.2. The Field Test Vehicle and Test Results

To further verify the effectiveness of the proposed method, a four-wheel steering and four-wheel hub motor drive vehicle is developed by our team. It is equipped with sensors for data collection and can feedback vehicle states information through the controller area network (CAN) bus.

5.2.1. Test Vehicle and Sensor Configuration

A 32 beams LiDAR, an RTK-GNSS system and an IMU are equipped on the testing intelligent electric vehicle. Sensor specifications of the test vehicle are shown in Figure 11 and Table 3. Gyro and accelerometer bias stability of IMU are 5 deg/h and 0.5 mg, respectively. In addition, the vehicle velocity can be obtained from the on-board CAN bus, and the groundtruth is provided by RTK-GNSS system.



Figure 11. Test vehicle and sensors configuration.

Table 3. Sensor specifications of test vehicle (Velodyne HDL-32E from Velodyne, San Jose, CA, USA. StarNeto, Newton-M2 from StarNeto Technology, Beijing, China).

Sensors	Specifications	No.	Frequency/Hz	Accuracy
3D LiDAR	Velodyne, HDL-32E, 32 beams	1	10	2 cm, 0.09 deg
RTK-GNSS system	StarNeto, Newton-M2, L1/L2 RTK	1	50	2 cm, 0.1 deg
IMU	Newton-M2	1	100	5 deg/h, 0.5 mg
Vehicle velocity	On-board CAN bus	1	100	0.1 m/s

5.2.2. Field Test Mapping Performance Analysis

The proposed mapping method was tested in a real scenario to verify its performance. In the field test, the dataset was collected in the industrial park with scenario change.

As shown in Figures 12 and 13, two point cloud maps were constructed by offline mapping process, and the map drift was effectively eliminated after optimization. As shown in Figures 14 and 15, due to the use of graph optimization algorithm, the optimized trajectory basically coincides with the groundtruth. It can be seen from Figures 16 and 17 that the position error of the three axes before optimization gradually increases over 1 m, and the average error before the optimization is 1.6 m. As shown in Figures 18 and 19, the error of the three axes is reduced to centimeter-level after optimization, and the average error is 8 cm. It can be seen from the above analysis that the proposed mapping method can reduce the position error significantly and construct a globally consistent map.

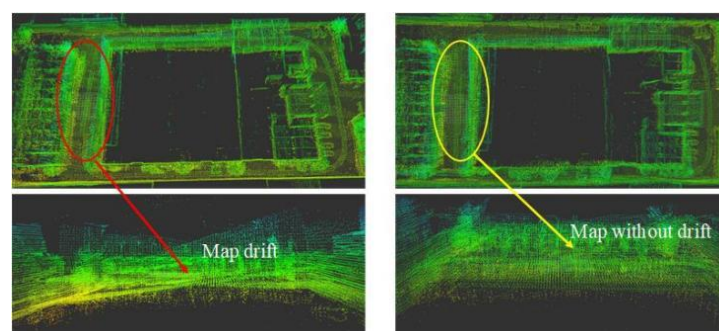


Figure 12. The map drift in small scale scenario is eliminated by back-end optimization. The figure on the left is unoptimized, and the right one is optimized. It can be seen that the map drift of the optimized map is significantly reduced.

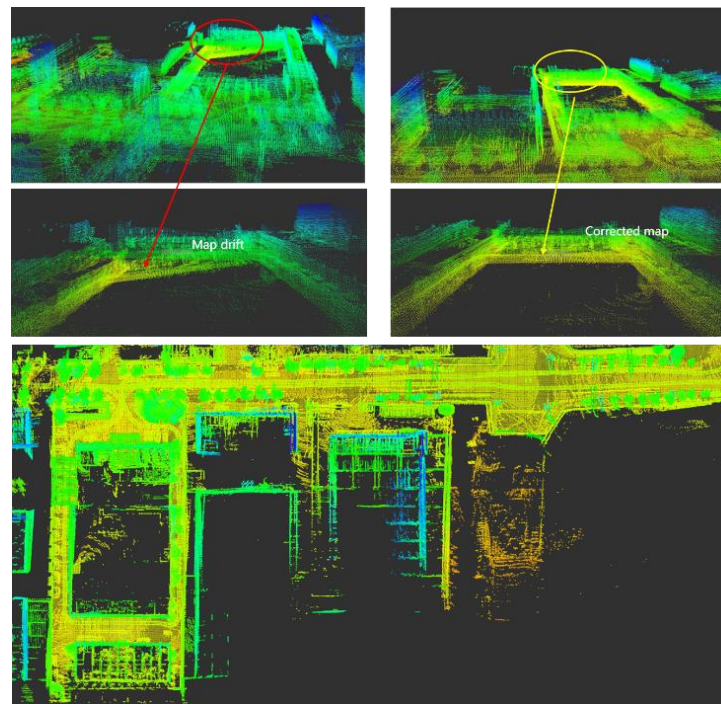


Figure 13. The map drift in large scale scenario is eliminated by back-end optimization. The figure on the upper left is unoptimized, and the upper right one is optimized. It can be seen that the map drift of the optimized map is significantly reduced. The bottom figure is the complete large-scale map after optimization.

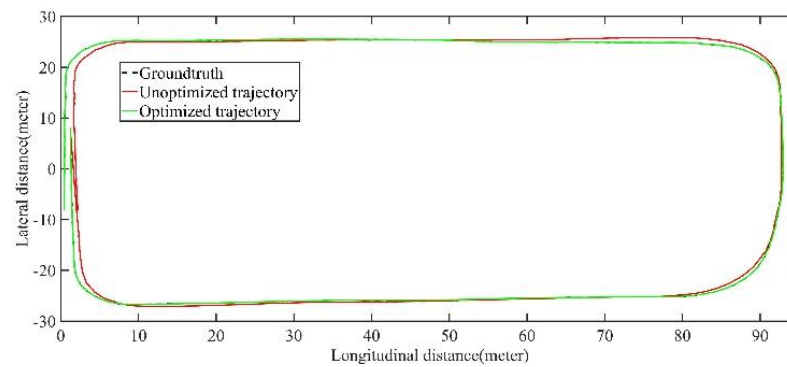


Figure 14. The comparison of trajectory before and after optimization with groundtruth of small scale scenario.

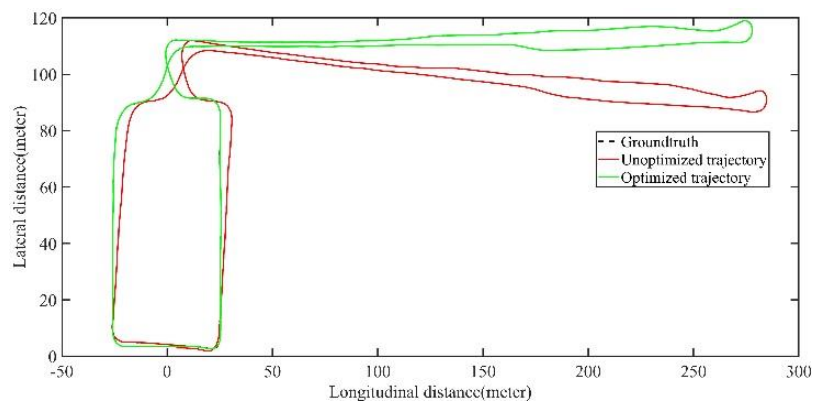


Figure 15. The comparison of trajectory before and after optimization with groundtruth of large scale scenario.

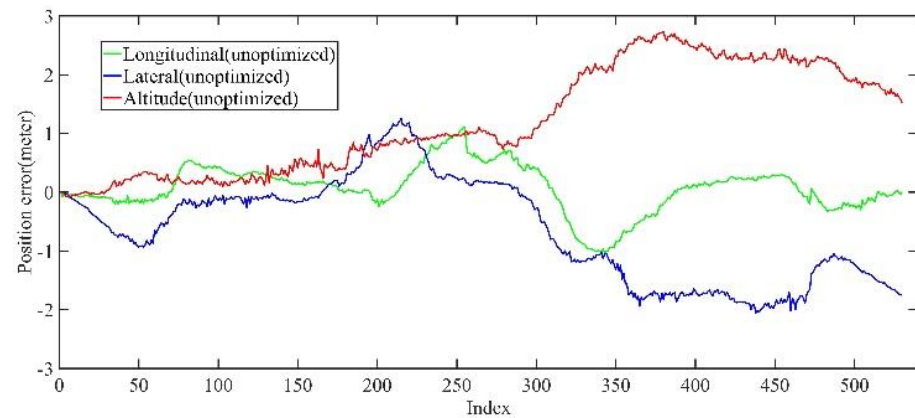


Figure 16. Small scale scenario unoptimized mapping trajectory position error in longitudinal, lateral, and altitude directions. Before optimization, the error is at the meter level.

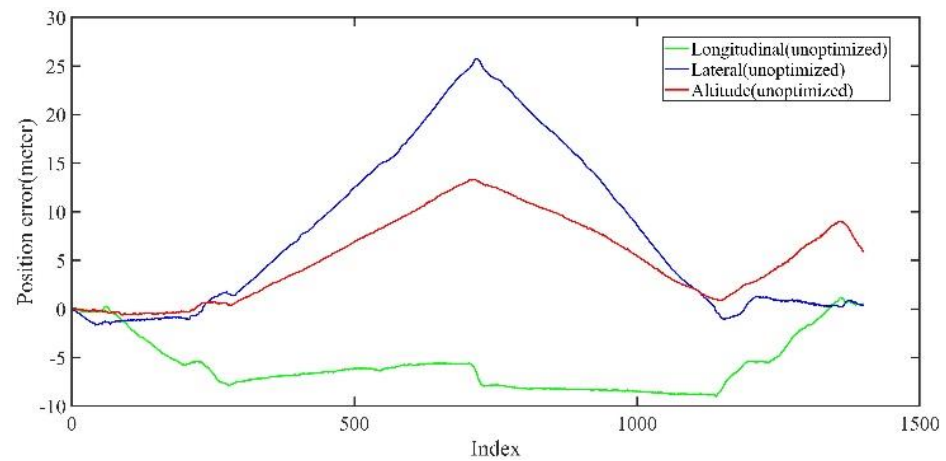


Figure 17. Large scale scenario unoptimized mapping trajectory position error in longitudinal, lateral, and altitude directions. Before optimization, the error is at the meter level.

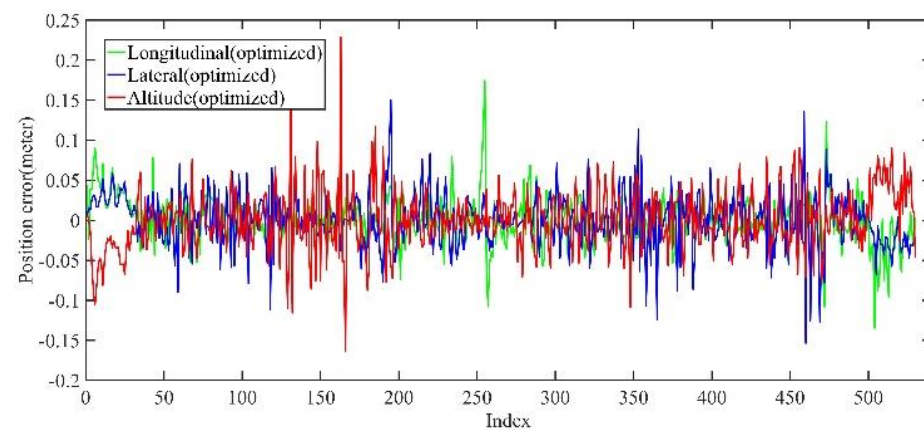


Figure 18. Small scale scenario optimized mapping trajectory position error in longitudinal, lateral, and altitude directions. After optimization, the error is at the centimeter level.

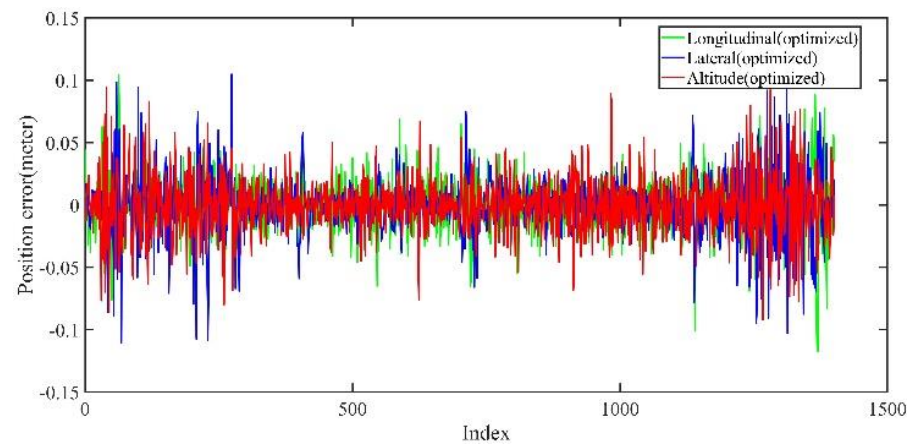


Figure 19. Large scale scenario optimized mapping trajectory position error in longitudinal, lateral, and altitude directions. After optimization, the error is at the centimeter level.

5.2.3. Field Test Localization Performance Analysis

Based on the prior point cloud map, five different field tests were implemented to verify the localization performance. The field tests included different driving conditions and scenarios.

Five sets of field tests represent different driving conditions and travel distances. As shown in Figures 20–25, due to the small changes in the scenario, the fused trajectory basically coincides with the groundtruth under the normal driving scenario and the curve driving scenario of one lap, and the maximum error does not exceed 45 cm. From Figures 26–31, we conducted another two sets of experiments under different scenarios, due to changes in the environment, the LiDAR localization position has drifted, but the maximum positioning error after fusing the IMU and the vehicle velocity can still be controlled within 55 cm. From Figures 32–34, in the large-scale scenario, the localization performance is still robust. It can be seen from Table 4 that the average position error is within 30 cm, which meets the autonomous driving lane-level localization requirements. Field tests scenarios results show that the localization algorithm based on the prior point cloud map can achieve good performance.

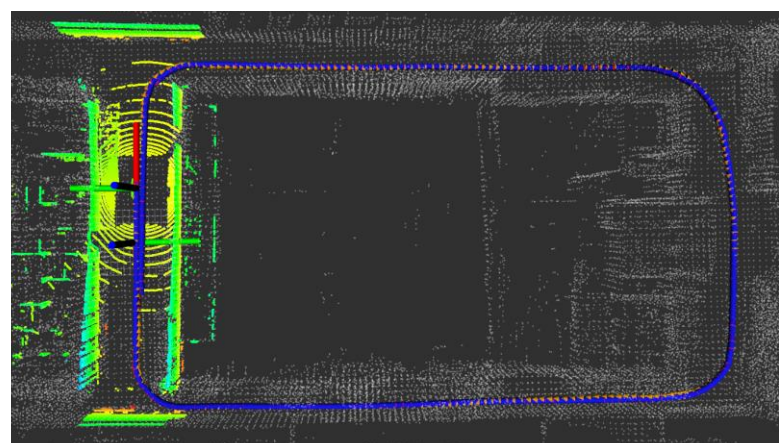


Figure 20. Field test NO.1. One lap under normal driving scenario localization test in prior point cloud map. The white points in the figure are the point cloud map built by the offline mapping process, the red line is the trajectory of LiDAR matching results, the orange line is groundtruth, and the blue line is the fused trajectory.

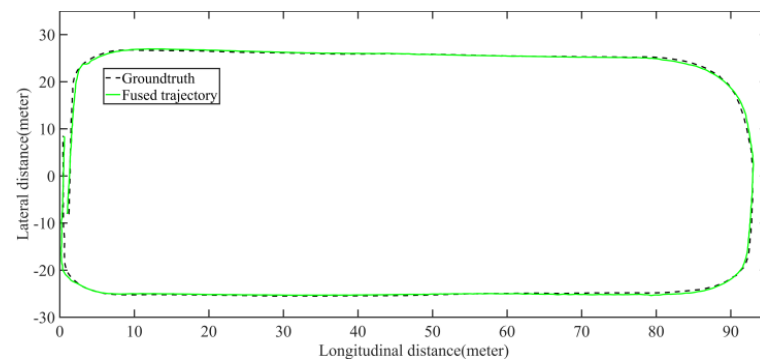


Figure 21. Field test NO.1. One lap under normal driving scenario localization test trajectory comparison. The comparison of fused trajectory with groundtruth.

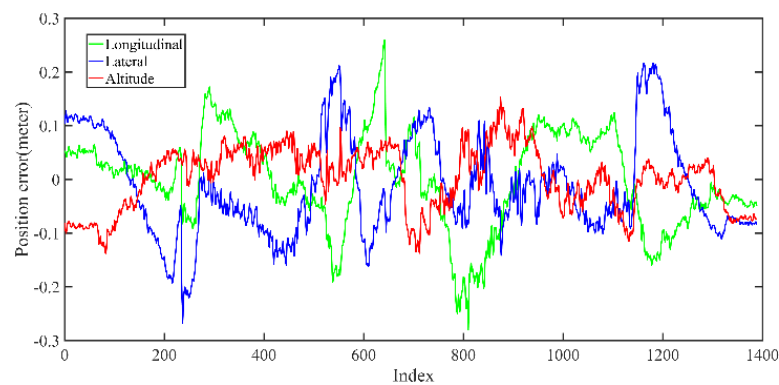


Figure 22. Field test NO.1. One lap under normal driving scenario localization error in longitudinal, lateral, and altitude directions, respectively; the localization error is at the centimeter level.

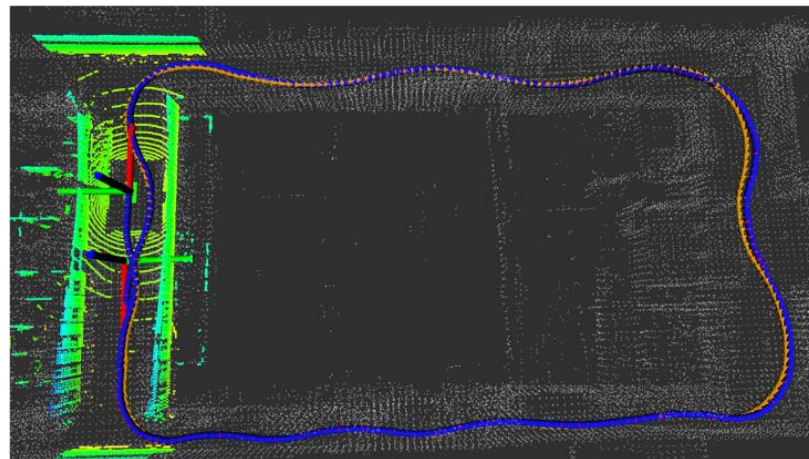


Figure 23. Field test NO.2. One lap under curve driving scenario localization test in prior point cloud map. The white points in the figure are the point cloud map built by the offline mapping process, the red line is the trajectory of LiDAR matching results, the orange line is groundtruth, and the blue line is the fused trajectory.

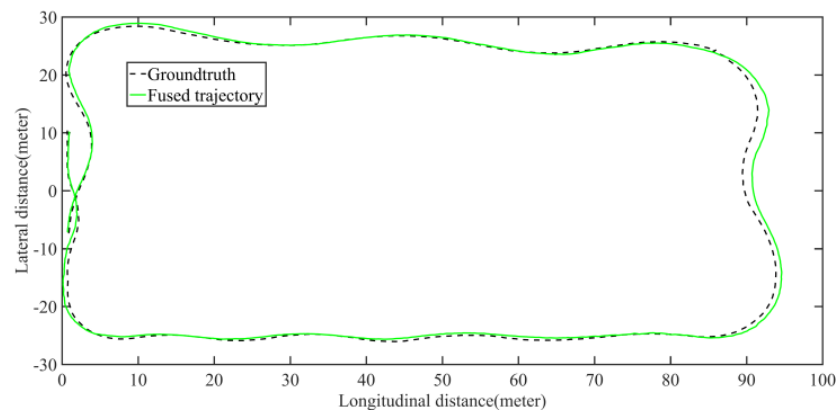


Figure 24. Field test NO.2. One lap under curve driving scenario localization test trajectory comparison. The comparison of fused trajectory with groundtruth.

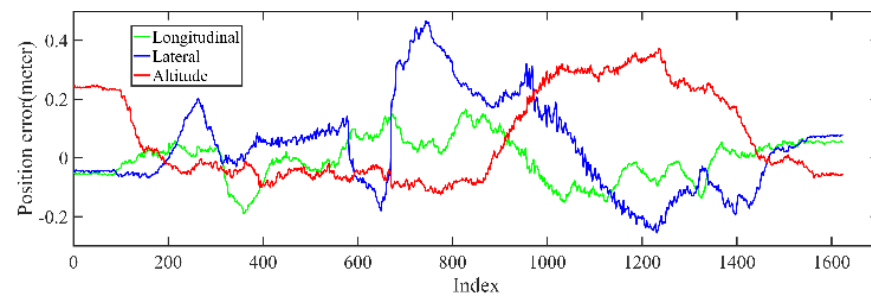


Figure 25. Field test NO.2. One lap under curve driving scenario localization error in longitudinal, lateral, and altitude directions, respectively; the localization error is at the centimeter level.

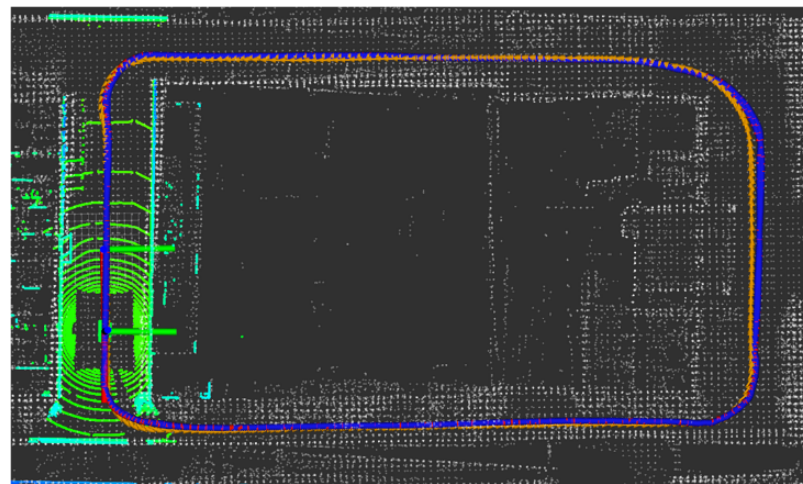


Figure 26. Field test NO.3. Two laps under normal driving scenario localization test in prior point cloud map. The white points in the figure are the point cloud map built by the offline mapping process, the red line is the trajectory of LiDAR matching results, the orange line is groundtruth, and the blue line is the fused trajectory.

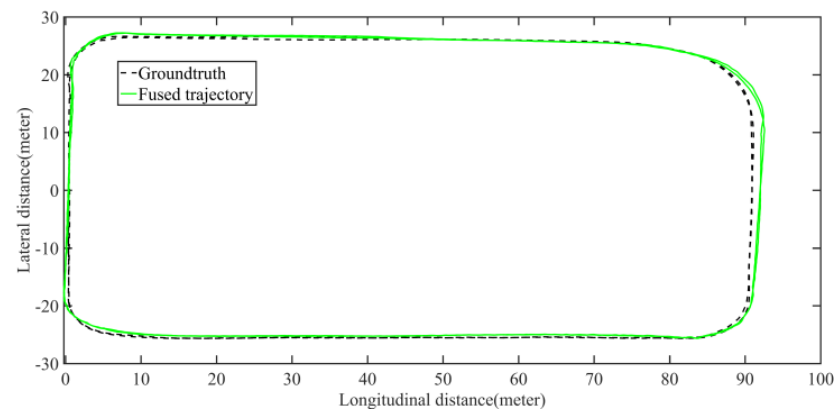


Figure 27. Field test NO.3. Two laps under normal driving scenario localization test trajectory comparison. The comparison of fused trajectory with groundtruth.

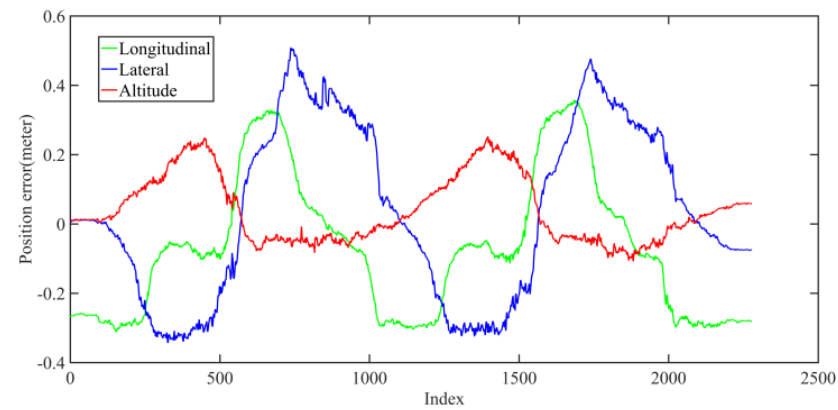


Figure 28. Field test NO.3. Two laps under normal driving scenario localization error in longitudinal, lateral, and altitude directions, respectively; the localization error is at the centimeter level.

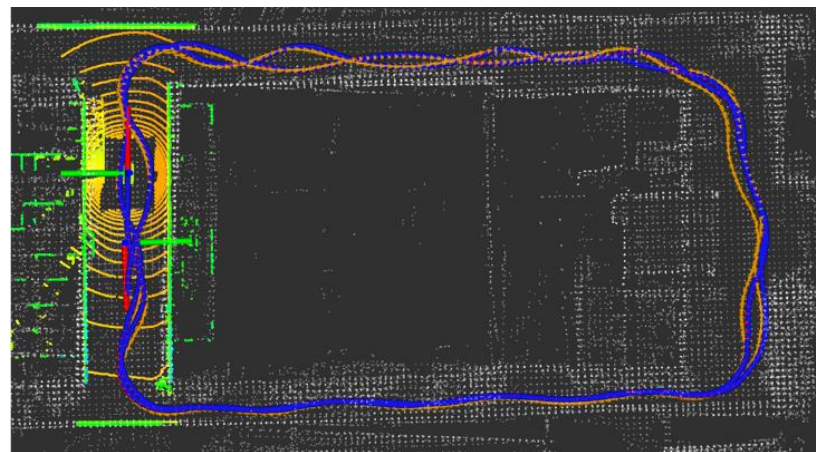


Figure 29. Field test NO.4. Two laps under curve driving scenario localization test in prior point cloud map. The white points in the figure are the point cloud map built by the offline mapping process, the red line is the trajectory of LiDAR matching results, the orange line is groundtruth, and the blue line is the fused trajectory.

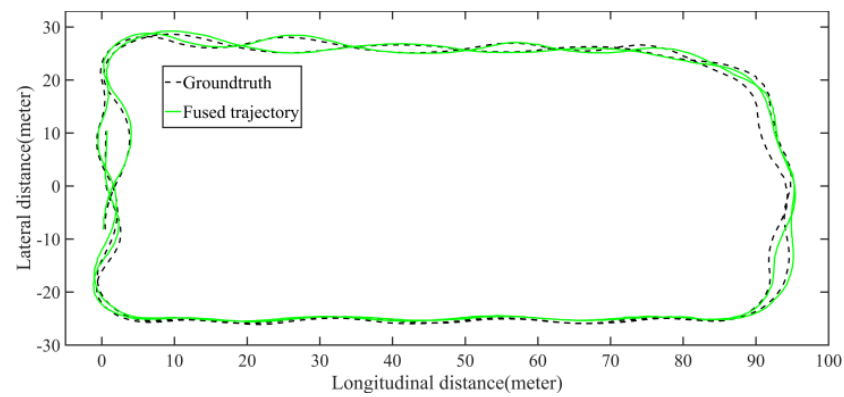


Figure 30. Field test NO.4. Two laps under curve driving scenario localization test trajectory comparison. The comparison of fused trajectory with groundtruth.

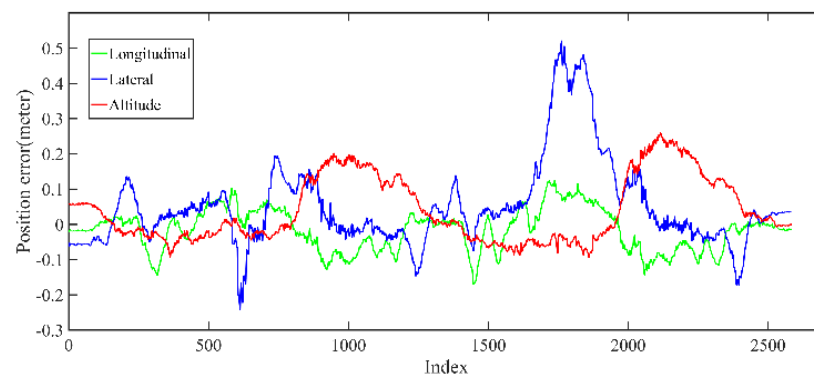


Figure 31. Field test NO.4. Two laps under curve driving scenario localization error in longitudinal, lateral, and altitude directions, respectively; the localization error is at the centimeter level.



Figure 32. Field test NO.5. Large-scale scenario localization test in prior point cloud map. The white points in the figure are the point cloud map built by the offline mapping process, the red line is the trajectory of LiDAR matching results, the orange line is groundtruth, and the blue line is the fused trajectory.

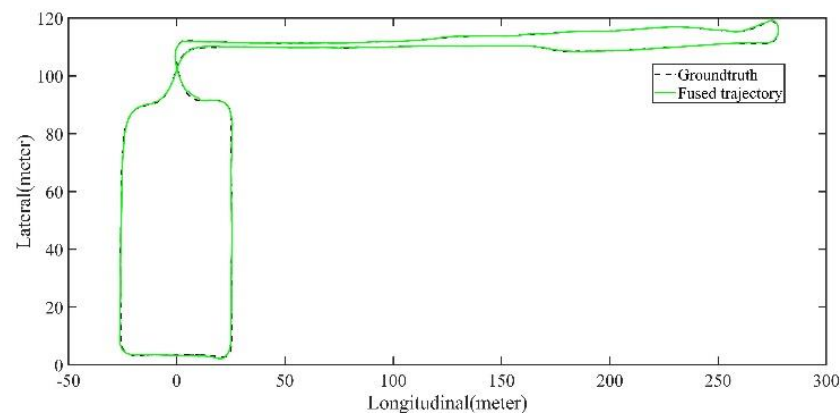


Figure 33. Field test NO.5. Large scale scenario localization test trajectory comparison. The comparison of fused trajectory with groundtruth.

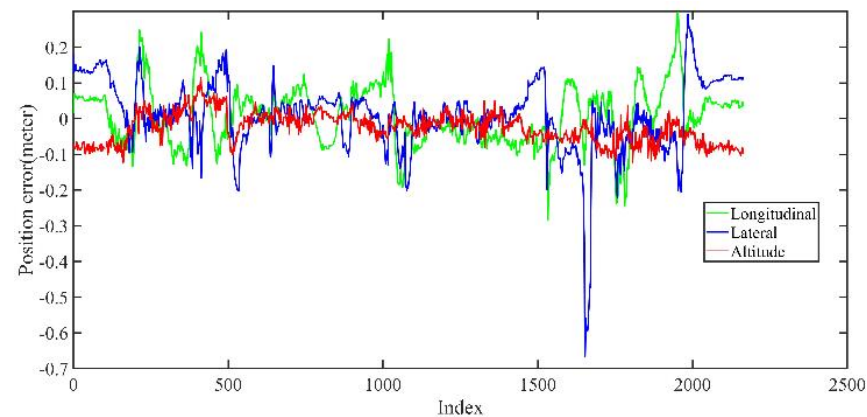


Figure 34. Field test NO.5. Large scale scenario localization error in longitudinal, lateral, and altitude directions, respectively; the localization error is at the centimeter level.

Table 4. Average errors from 5 field test scenarios.

Field Test Sequence	Test Scenario	Average Error	RMSE
01	Normal driving, 1 lap	21.6 cm	23.2 cm
02	Curve driving, 1 lap	22.5 cm	24.3 cm
03	Normal driving, 2 laps	28.3 cm	29.2 cm
04	Curve driving, 2 laps	25.5 cm	27.1 cm
05	Large scale, 1 laps	29.2 cm	29.6 cm

6. Discussion

From the experimental results, this can all be summarized as the following:

The offline mapping method proposed in this paper can effectively eliminate map drift, provide a mapping accuracy of 5–10 cm, and can be used in localization work to provide a stable and reliable map data source.

In the online localization process, we use the multi-sensor fusion method to achieve a positioning accuracy of 20–30 cm, which benefits from the good mapping accuracy and the design of the multi-sensor fusion model.

7. Conclusions and Future Work

This paper presented a LiDAR-based sensor fusion SLAM and localization method for autonomous vehicle offline mapping and online localization. In the mapping process, NDT registration, scan context loop closure detection and RTK-GNSS position are considered in front-end and back-end, innovative use of the pose graph to combine multiple methods

to optimize position and reduce map drift, which realizes 5–10 cm mapping accuracy, and the map drift is eliminated effectively. In the online localization process, the ESKF is used to fuse complementary sensor information, such as LiDAR, IMU and vehicle velocity to achieve good localization accuracy in various challenging scenarios, which takes full advantage of the vehicle velocity constraints of ground autonomous vehicles to optimize localization results, and reaches 20–30 cm localization accuracy and shows environmental robustness. Such good and stable mapping and localization results can assist autonomous vehicles to safely complete navigation tasks in the lane. Furthermore, the proposed method can be used to fuse more sensors for offline mapping and online localization, respectively, facing different applications. In the future work, we will try to tightly couple the IMU and LiDAR to the mapping system, which can reduce the drift of the front-end, and improve the quality of the mapping.

Author Contributions: Conceptualization, K.D. and B.S.; data curation, K.D. and S.Z.; software, K.D.; formal analysis, K.D., G.W. and B.S.; funding acquisition, B.S., J.W. and F.M.; writing—original draft preparation, K.D. and B.S.; writing—review and editing, G.W. and Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grants 52102457, in part by the China Postdoctoral Science Foundation under Grant 2021M691207 and in part by the Science and Technology Development Plan Project of Changchun under Grant 21QC09.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

- Aslan, M.F.; Durdu, A.; Yusefi, A.; Sabanci, K.; Sungur, C. A tutorial: Mobile robotics, SLAM, bayesian filter, keyframe bundle adjustment and ROS applications. *Robot Oper. Syst. (ROS)* **2021**, *6*, 227–269.
- Suhr, J.K.; Jang, J.; Min, D.; Jung, H.G. Sensor fusion-based low-cost vehicle localization system for complex urban environments. *IEEE Trans. Intell. Transp. Syst.* **2016**, *18*, 1078–1086. [\[CrossRef\]](#)
- Kuutti, S.; Fallah, S.; Katsaros, K.; Dianati, M.; McCullough, F.; Mouzakitis, A. A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications. *IEEE Internet Things J.* **2018**, *5*, 829–846. [\[CrossRef\]](#)
- Kim, C.; Cho, S.; Sunwoo, M.; Resende, P.; Bradai, B.; Jo, K. Updating Point Cloud Layer of High Definition (HD) Map Based on Crowd-Sourcing of Multiple Vehicles Installed LiDAR. *IEEE Access* **2021**, *9*, 8028–8046. [\[CrossRef\]](#)
- Cadena, C.; Carlone, L.; Carrillo, H.; Latif, Y.; Scaramuzza, D.; Neira, J.; Reid, I.; Leonard, J.J. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Trans. Robot.* **2016**, *32*, 1309–1332. [\[CrossRef\]](#)
- Pomerleau, F.; Colas, F.; Siegwart, R.; Magnenat, S. Comparing ICP variants on real-world data sets. *Auton. Robot.* **2013**, *34*, 133–148. [\[CrossRef\]](#)
- Einhorn, E.; Gross, H.-M. Generic NDT mapping in dynamic environments and its application for lifelong SLAM. *Robot. Auton. Syst.* **2015**, *69*, 28–39. [\[CrossRef\]](#)
- Zhang, J.; Singh, S. LOAM: Lidar Odometry and Mapping in Real-time. In *Robotics: Science and Systems*; MIT Press: Cambridge, MA, USA, 2007; Volume 2.
- Zhang, T.; Wu, K.; Song, J.; Huang, S.; Dissanayake, G. Convergence and consistency analysis for a 3-D invariant-EKF SLAM. *IEEE Robot. Autom. Lett.* **2017**, *2*, 733–740. [\[CrossRef\]](#)
- Ren, Z.; Wang, L.; Bi, L. Robust GICP-based 3D LiDAR SLAM for underground mining environment. *Sensors* **2019**, *19*, 2915. [\[CrossRef\]](#)
- Fayyad, J.; Jaradat, M.A.; Gruyer, D.; Najjaran, H. Deep learning sensor fusion for autonomous vehicle perception and localization: A review. *Sensors* **2020**, *20*, 4220. [\[CrossRef\]](#)
- Ahmed, H.; Tahir, M. Accurate attitude estimation of a moving land vehicle using low-cost MEMS IMU sensors. *IEEE Trans. Intell. Transp. Syst.* **2016**, *18*, 1723–1739. [\[CrossRef\]](#)
- Levinson, J.; Montemerlo, M.; Thrun, S. Map-based precision vehicle localization in urban environments. In *Robotics: Science and Systems*; MIT Press: Cambridge, MA, USA, 2009; Volume 4.
- Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Proceedings of the Computer Vision and Pattern Recognition*, Providence, RI, USA, 16–21 June 2012; pp. 3354–3361.

15. Junior, E.M.O.; Santos, D.R.; Miola, G.A.R. A New Variant of the ICP Algorithm for Pairwise 3D Point Cloud Registration. *Am. Acad. Sci. Res. J. Eng. Technol. Sci.* **2022**, *85*, 71–88.
16. Magnusson, M.; Lilienthal, A.; Duckett, T. Scan registration for autonomous mining vehicles using 3D-NDT. *J. Field Robot.* **2007**, *24*, 803–827. [\[CrossRef\]](#)
17. Ren, R.; Fu, H.; Wu, M. Large-scale outdoor slam based on 2d lidar. *Electronics* **2019**, *8*, 613. [\[CrossRef\]](#)
18. Li, G.; Liao, X.; Huang, H.; Song, S.; Zeng, Y. Robust Stereo Visual SLAM for Dynamic Environments with Moving Object. *IEEE Access* **2021**, *9*, 32310–32320. [\[CrossRef\]](#)
19. Wen, W.; Hsu, L.-T.; Zhang, G. Performance analysis of NDT-based graph SLAM for autonomous vehicle in diverse typical driving scenarios of Hong Kong. *Sensors* **2018**, *18*, 3928. [\[CrossRef\]](#)
20. Luo, L.; Cao, S.-Y.; Han, B.; Shen, H.-L.; Li, J. BVMATCH: LiDAR-Based place recognition using bird's-eye view images. *IEEE Robot. Autom. Lett.* **2021**, *6*, 6076–6083. [\[CrossRef\]](#)
21. Kasaei, S.H.; Tomé, A.M.; Lopes, L.S.; Oliveira, M. GOOD: A global orthographic object descriptor for 3D object recognition and manipulation. *Pattern Recognit. Lett.* **2016**, *83*, 312–320. [\[CrossRef\]](#)
22. Kim, G.; Park, B.; Kim, A. 1-day learning, 1-year localization: Long-term lidar localization using scan context image. *IEEE Robot. Autom. Lett.* **2019**, *4*, 1948–1955. [\[CrossRef\]](#)
23. 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. [\[CrossRef\]](#)
24. Arshad, S.; Kim, G.-W. Role of deep learning in loop closure detection for visual and lidar slam: A survey. *Sensors* **2021**, *21*, 1243. [\[CrossRef\]](#)
25. Fan, T.; Wang, H.; Rubenstein, M.; Murphey, T. CPL-SLAM: Efficient and certifiably correct planar graph-based SLAM using the complex number representation. *IEEE Trans. Robot.* **2020**, *36*, 1719–1737. [\[CrossRef\]](#)
26. Vallvé, J.; Solà, J.; Andrade-Cetto, J. Pose-graph SLAM sparsification using factor descent. *Robot. Auton. Syst.* **2019**, *119*, 108–118. [\[CrossRef\]](#)
27. Macenski, S.; Jambrecic, I. SLAM Toolbox: SLAM for the dynamic world. *J. Open Source Softw.* **2021**, *6*, 2783. [\[CrossRef\]](#)
28. Grisetti, G.; Kümmerle, R.; Stachniss, C.; Burgard, W. A tutorial on graph-based SLAM. *IEEE Intell. Transp. Syst. Mag.* **2010**, *2*, 31–43. [\[CrossRef\]](#)
29. Latif, Y.; Cadena, C.; Neira, J. Robust loop closing over time for pose graph SLAM. *Int. J. Robot. Res.* **2013**, *32*, 1611–1626. [\[CrossRef\]](#)
30. Lin, R.; Xu, J.; Zhang, J. GLO-SLAM: A slam system optimally combining GPS and LiDAR odometry. *Ind. Robot Int. J. Robot. Res. Appl.* **2021**; ahead-of-print. [\[CrossRef\]](#)
31. Chang, L.; Niu, X.; Liu, T. GNSS/IMU/ODO/LiDAR-SLAM Integrated Navigation System Using IMU/ODO Pre-Integration. *Sensors* **2020**, *20*, 4702. [\[CrossRef\]](#) [\[PubMed\]](#)
32. Chen, C.; Wang, B.; Lu, C.X.; Trigoni, N.; Markham, A. A survey on deep learning for localization and mapping: Towards the age of spatial machine intelligence. *arXiv* **2020**, arXiv:2006.12567.
33. Wang, K.; Ma, S.; Chen, J.; Ren, F.; Lu, J. Approaches challenges and applications for deep visual odometry toward to complicated and emerging areas. *IEEE Trans. Cogn. Dev. Syst.* **2020**, *14*, 35–49. [\[CrossRef\]](#)
34. Gao, Y.; Liu, S.; Atia, M.M.; Noureldin, A. INS/GPS/LiDAR integrated navigation system for urban and indoor environments using hybrid scan matching algorithm. *Sensors* **2015**, *15*, 23286–23302. [\[CrossRef\]](#) [\[PubMed\]](#)
35. Grupp, M. evo: Python Package for the Evaluation of Odometry and Slam. Available online: <https://github.com/MichaelGrupp/evo> (accessed on 16 November 2022).

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.