https://doi.org/10.20965/jaciii.2022.p0206

Paper:

High-Precision and Fast LiDAR Odometry and Mapping Algorithm

Qingshan Wang[*,**], **Jun Zhang**[**,†], **Yuansheng Liu**[**], **and Xinchen Zhang**[**]

[*]CATARC (Tianjin) Automotive Engineering Research Institute Co., Ltd.
No.68 Xianfeng East Road, Dongli District, Tianjin 300300, China
E-mail: ldwangqingshan@163.com
[**]College of Robotics, Beijing Union University
No.97 Beisihuan East Road, Chao Yang District, Bejing 100101, China
[†]Corresponding author

[Received October 7, 2019; accepted February 3, 2022]

LiDAR SLAM technology is an important method for the accurate navigation of automatic vehicles and is a prerequisite for the safe driving of automatic vehicles in the unstructured road environment of complex parks. This paper proposes a LiDAR fast point cloud registration algorithm that can realize fast and accurate localization and mapping of automatic vehicle point clouds through a combination of a normal distribution transform (NDT) and point-to-line iterative closest point (PLICP). First, the NDT point cloud registration algorithm is applied for the rough registration of point clouds between adjacent frames to achieve a rough estimate of the pose of automatic vehicles. Then, the PLICP point cloud registration algorithm is adopted to correct the rough registration result of the point cloud. This step completes the precise registration of the point cloud and achieves an accurate estimate of the pose of the automatic vehicle. Finally, cloud registration is accumulated over time, and the point cloud information is continuously updated to construct the point cloud map. Through numerous experiments, we compared the proposed algorithm with PLICP. The average number of iterations of the point cloud registration between adjacent frames was reduced by 6.046. The average running time of the point cloud registration between adjacent frames decreased by 43.05156 ms. The efficiency of the point cloud registration calculation increased by approximately 51.7%. By applying the KITTI dataset, the computational efficiency of NDT-ICP was approximately 60% higher than that of LeGO-LOAM. The proposed method realizes the accurate localization and mapping of automatic vehicles relying on vehicle LiDAR in a complex park environment and was applied to a Small Cyclone automatic vehicle. The results indicate that the proposed algorithm is reliable and effective.

Keywords: LiDAR SLAM, NDT, PLICP, localization, mapping

1. Introduction

Simultaneous localization and mapping (SLAM) technology involves placing an automatic vehicle in a completely unknown environment, where it begins moving from an unspecified location while incrementally creating a map of the environment and then uses this created map to locate and navigate itself. The complexity of the SLAM problem lies in the interdependence of the localization and composition. The precise localization of automatic vehicles depends on the consistency of the map of the surrounding environment. Similarly, the construction of highly consistent environmental maps is based on accurate localization [1, 2].

Owing to the development of automatic vehicle technology, LiDAR is an important sensor in the field of automatic vehicle sensing. It has significant research value in the real-time localization and mapping technology of automatic vehicles. Usually, the precise localization of an automatic vehicle relies on the global positioning system (GPS).

However, owing to the weak GPS signal caused by high-rise buildings, trees, and flyovers, automatic vehicles must deal with uncontrollable factors. Therefore, this study uses vehicle LiDAR to realize the precise localization and map construction of an automatic vehicle.

Three-dimensional (3D) mapping is a popular technology [3–9]. In addition, LiDAR obtains obstacle information by scanning the surrounding environment information [10, 11]. Mapping with LiDAR is common, as LiDAR can provide high-frequency range measurements where errors are relatively constant, irrespective of the distances measured. In the case where the only motion of the LiDAR amounts to rotating a laser beam, the registration of the point cloud is simple. However, if the LiDAR equipment itself is moving, as in several applications of interest, accurate mapping requires the knowledge of LiDAR poses during continuous laser ranging. A common way to solve this problem is to use independent position estimation (e.g., using a GPS/INS) to register the laser points into a fixed coordinate system. Another set of methods uses odometry measurements, such as those

© Fuji Technology Press Ltd. Creative Commons CC BY-ND: This is an Open Access article distributed under the terms of the Creative Commons Attribution-NoDerivatives 4.0 International License (http://creativecommons.org/licenses/by-nd/4.0/)

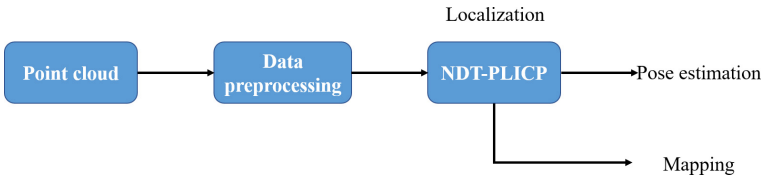


Fig. 1. Block diagram of the LiDAR pose estimation and mapping software system.

from wheel encoders or visual odometry systems [12, 13], to register the laser points. Because odometry integrates small incremental motions over time, it is prone to drift, and considerable attention has been devoted to the reduction of drift. Therefore, LiDAR is used to improve drift-related problems in ranging estimation. This study focused on the use of LiDAR to realize real-time pose estimation and mapping.

Analyzing the original problem of adjacent frame point cloud registration, we realized that it is not ideal to calculate adjacent frames and obtain perfect coordinate transformation; however, in practice, only the perfect state can be achieved. Therefore, this paper proposes a fast two-step point cloud registration algorithm to realize precise localization and map construction by automatic vehicles. First, the normal distribution transform (NDT) point cloud registration algorithm is used to estimate roughly the pose of an automatic vehicle. Then, the point cloud of the rough registration is corrected by the point-to-line iterative closest point (PLICP) point cloud registration algorithm to realize an accurate estimation of the automatic vehicle position and complete the map update. **Fig. 1** shows a diagram of the software system. This algorithm was applied to a Small Cyclone automatic vehicle and verified in a campus environment. The results demonstrate that the proposed algorithm is reliable and effective.

2. Related Work

Based on the LiDAR scanning of adjacent frames, a large amount of repeated point cloud information is used to determine the conversion relationship between frames, such that the distance between the point clouds of two frames is infinitely close. The typical approach for determining the transformation between two LiDAR scans is the iterative closest point (ICP) algorithm [14]. By finding correspondence at a point-wise level, ICP iteratively aligns two sets of points until the stopping criteria are satisfied. When the scans include a large number of points, ICP may suffer from prohibitive computational costs. Several ICP variants have been proposed to improve efficiency and accuracy [15]. Censi proposed a point-to-line-based ICP algorithm [16] that calculates the minimum distance between a point and a line. Chen and Medioni proposed a point-to-surface-based ICP algorithm [17] that calculates the minimum distance between a point and a surface. Segal et al. proposed a surface-to-surface ICP algorithm [18] that calculates the shortest distance between two surfaces. In addition, several ICP variants have leveraged parallel computing for improving efficiency [19–22]. An analysis of the above algorithm indicated that, although the efficiency of point cloud registration is improved, a part of the registration accuracy is lost, and the initial pose of the point cloud is stricter. Therefore, this study focused on providing a good initial state of the point cloud. Because LiDAR scans contain a large amount of point cloud data, to improve the efficiency of point cloud registration, Biber and Strasser proposed a 2D-NDT point cloud registration algorithm [23], which is a fast space model technology. Analogous to an occupancy grid, 2D-NDT subdivides the 2D plane into cells. For each cell, 2D-NDT assigns a normal distribution that locally models the probability of measuring a point. The result of the transform is a piecewise continuous and differentiable probability density function that can be used to match another scan by using Newton's algorithm. Thereby, no explicit correspondence needs to be established. Because the current LiDAR scan data form a 3D point cloud, Magnusson et al. proposed an improved NDT algorithm [24] and applied it to 3D space. 3D-NDT allows for accurate registration using a memory-efficient representation of the scan surface. The NDT algorithm uses a point cloud density function for point cloud registration. Although this method reduces the time consumption of the algorithm, it also reduces the accuracy of the point cloud registration.

Feature-based matching methods are attracting increasing attention because they require fewer computational resources, which is accomplished by extracting representative features from the environment. These features should be suitable for effective matching and invariance of the point-of-views. Several detectors, such as point feature histograms [25] and viewpoint feature histograms [26], have been proposed for extracting such features from point clouds using simple and efficient techniques. Numerous algorithms that use features for point cloud registration have been proposed. Bosse and Zlot [27, 28] have presented a keypoint selection algorithm that performs point curvature calculations in a local cluster. The selected keypoints are then used to perform matching and place recognition. Zhang and Singh proposed a low-drift real-time LiDAR odometry and mapping (LOAM) [29, 30] method. LOAM performs point feature-to-edge/plane scan matching to determine the correspondence between scans. Features are extracted by calculating the roughness of a point in its local region. Points with high roughness values are selected as edge features. Dube et al. proposed

a segmentation-based registration algorithm [31], wherein a feature vector is calculated for each segment based on its eigenvalues and shape histograms. A random forest is used to match the segments from the two scans. Shan and Englot proposed lightweight and ground-optimized LiDAR odometry and mapping on variable terrain (LeGO-LOAM) [32]. The above algorithm can extract valid feature information in certain scenarios; however, in an unstructured complex scene or a single scene, there are no apparent features, resulting in inaccurate point cloud registration and a large drift.

3. Fast Point Cloud Registration and Map Construction

LiDAR point cloud data are crucial for the localization and mapping of automatic vehicles. The initial pose state of the point cloud determines the efficiency and accuracy of the point cloud registration algorithm. When LiDAR performs environmental data acquisition, owing to the interference of the environment and LiDAR itself, the data returned by LiDAR have a large volume and uneven density and contain noise. To improve the real-time performance and accuracy of the point cloud registration algorithm, the original sensor point cloud data are preprocessed. First, LiDAR is used to perform data collection using an automatic vehicle. Then, outlier removal, voxel grid [33] filtering, and other types of processing are performed. Finally, the processed data are input into the point cloud registration algorithm for subsequent processing. The proposed algorithm combines the advantages of NDT and PLICP and proposes a fast two-step point cloud registration algorithm to realize the precise localization and map construction of automatic vehicles. First, the NDT point cloud registration algorithm is used to estimate the approximate pose of an automatic vehicle. Then, the PLICP point cloud registration algorithm is used to correct the coarsely registered point cloud to realize the accurate estimation of the automatic vehicle poses and complete the map update.

3.1. Rough Estimate of Automatic Vehicle Poses

The NDT algorithm represents a dataset of a large number of discrete points in a cube as a piecewise continuous differentiable probability density function. First, the point cloud space is divided into several identical cubes, and there are at least five points in each cube; the mean value q and covariance matrix of each point in each cube are calculated as

$$q = \frac{1}{n}\sum_i X_i, \quad \ldots \quad (1)$$

$$\Sigma = \frac{1}{n}\sum_i (X_i - q)(X_i - q)^T, \quad \ldots \quad (2)$$

where $X_{i=1,2,\ldots,n}$ represents the cluster of points and n represents the number of point clouds.

Then, the discrete point cloud is represented by a piecewise continuous differentiable representation in the form of probability density, and the probability density of each point in the cube is represented by the NDT algorithm:

$$p(X) \sim \exp\left(-\frac{(X-q)^T\Sigma^{-1}(X-q)}{2}\right). \quad \ldots \quad (3)$$

The Hessian matrix method is used to calculate the registration between adjacent frames of point cloud data scanned by the vehicle LiDAR. The NDT point cloud registration algorithm is as follows:

(1) Calculate the NDT of the first frame of the LiDAR scanning point cloud.

(2) Initialize the coordinate transformation parameter T:

$$T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \quad \ldots \quad (4)$$

where R is a rotation matrix of size 3×3 and t is a translation matrix of size 3×1.

(3) Map the point cluster of the second frame of laser scanning to the coordinate system of the first frame according to the coordinate transformation parameters and obtain the point cluster of the mapped X_i'.

(4) Calculate the NDT after each point map transformation as follows:

$$p(X_i') \sim \exp\left(-\frac{(X_i' - q_i)^T\Sigma_i^{-1}(X_i' - q_i)}{2}\right). \quad (5)$$

(5) Add the probability density of each point, and evaluate the transformation parameters of coordinates as follows:

$$s(p) = \sum_i \exp\left[-\frac{(X_i' - q_i)^T\sum_i^{-1}(X_i' - q_i)}{2}\right]. \quad (6)$$

(6) Use the Hessian matrix method to optimize $s(p)$.

(7) Repeat steps 3–6 until the convergence condition is met.

At this point, the NDT algorithm completes the coarse registration process of the point cloud between adjacent frames and returns the coordinate transformation parameter T' of the optimal solution.

3.2. Accurate Estimate of Automatic Vehicle Poses

The NDT algorithm completes a rough registration of the point cloud between adjacent frames. In this study, we used the PLICP algorithm to perform an accurate correction of the rough registration of point clouds to realize the accurate localization of the automatic vehicle. PLICP solves the surface matching problem, which can be expressed as follows: provide a reference S^{ref} and a set of points $\{p_i\}$, determine the roto-translation $q = (t, \theta)$ that

minimizes the distance of the points p_i, and roto-translate by q to their projection on surface S^{ref}. This is accomplished using the following formula:

$$\min_{q_{k+1}} \sum_i \left(n_i^T \left[p_i \oplus q_{k+1} - \prod \{ S^{ref}, p_i \oplus q_k \} \right] \right)^2, \quad (7)$$

where n_i is the normal to the surface at the projected point. The symbol $\oplus$ denotes the roto-translation operator. At this time, q_k is the initial transformation value obtained by the rough registration of the point cloud through NDT, which provides a good initial transformation of the point cloud for PLICP point cloud registration. Continuing from the course registration of the point cloud:

(8) Project the data of the current frame according to the initial pose of the reference frame coordinates.

(9) For point i of the current frame, find the nearest two points (j_i, j_{i+1}) in the reference frame.

(10) Calculate the error and remove the point where the error is relatively large.

(11) Calculate the minimized error function.

At this point, the PLICP algorithm has completed the correction of the point cloud registration of the NDT algorithm and achieved an accurate registration of the point cloud between frames.

3.3. Point Cloud Registration Algorithm

This paper proposes a two-step method for point cloud registration between adjacent frames. First, the point cloud data after preprocessing are used to perform point cloud registration using the NDT algorithm, and the pose of the automatic vehicle is roughly estimated. Then, using the PLICP algorithm to correct the error of the point cloud registration of the NDT algorithm, the interframe error is minimized, and an accurate estimation of the automatic vehicle pose is achieved. Let the surrounding point clouds at times t and $t+1$ after point cloud pretreatment be X_t and X_{t+1}, respectively. The algorithm is presented in **Fig. 2**.

NDT point cloud registration is performed when the initial relative position of the X_t and X_{t+1} point clouds is not known. The main purpose is to estimate quickly a rough point cloud registration matrix T using the NDT algorithm if the initial conditions are unknown. The calculation of NDT point cloud registration requires high efficiency, and the accuracy of the calculation results is not excessively high. The point cloud outputs after registration by the NDT algorithm are X_t and X_{t+1}, respectively, and the output transform parameter of the optimal solution coordinate system is T'.

Point cloud registration using the PLICP algorithm has strict requirements for the initial pose and posture of the point cloud, and the NDT algorithm can provide a good pose and posture of the point cloud for the PLICP algorithm. Therefore, the PLICP accurate registration algorithm uses the transformation matrix T' obtained using the NDT algorithm as a pose transformation relationship and then uses the PLICP algorithm to calculate a more accurate pose transformation. The PLICP algorithm calculates the distance from X'_{t+1} to X and constructs a new rotation translation matrix, T''. The minimum value of the transformation is obtained by calculating the point cloud $X(t,t+1)$ through the rotation translation matrix T''. If the minimum value satisfies the threshold condition, the algorithm ends. Otherwise, the iteration is repeated until the error meets the threshold condition or the number of iterations is completed.

Using this rough registration of the interframe point cloud, followed by the accurate registration of the interframe point cloud, an accurate estimation of the automatic parking pose is realized, and the accurate localization of the automatic vehicle is completed.

NDT-PLICP point cloud fast and accurate registration algorithm

Input adjacent frame point cloud: X_t, X_{t+1}
Output optimal transformation matrix T
Point cloud rough registration:
    for all X_t do
        $q \leftarrow \frac{1}{n}\sum_n X_t$
        $\Sigma \leftarrow \sum_n (X_t - q)(X_t - q)^T$
    end for
    find $T' \leftarrow X_t\, X_{t+1}$
    for all X_{t+1} do
        $PDF \leftarrow P(X'_{t+1})$
        $S(P) \leftarrow PDF$
    end for
Point cloud accurate registration: X'_{t+1}, X_t
    begin:
        $p \oplus (t,\theta) \triangleq R(\theta)p + t$
        $\min d \leftarrow (j_i, j_{i+1}) \sim i$
        $\min \sum_i \left(n_i^T \left[R(\theta_{k+1})p_i + t_{k+1} - p_{j_1^i} \right] \right)^2$
    end

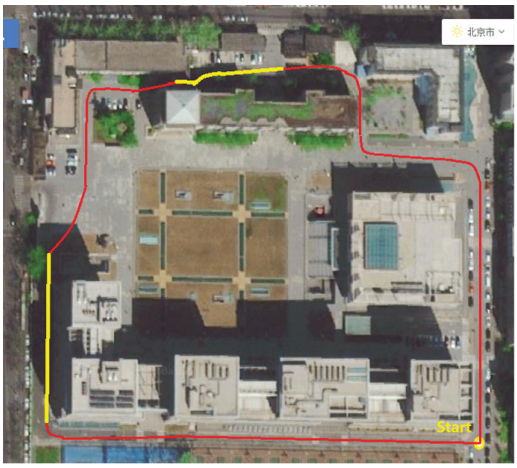
Fig. 2. NDT-PLICP point cloud registration algorithm.

3.4. Mapping

The map is constructed using the point cloud returned by the onboard sensor; the adjacent frame is transformed at 10 Hz, and the point cloud map is constructed at a low frequency of 1 Hz. In addition, in this study, we used a preprocessed point cloud for map construction, including the removal of noise points and point cloud after voxel grid filtering, providing effective point cloud data for map construction.

After data preprocessing, the point cloud of the surrounding environment is obtained. Let the point clouds obtained by scanning the surrounding environment at times t and $t+1$ be X_t and X_{t+1}, respectively. The valid map saved at time t is denoted by Q_t. In this study, the NDT algorithm was used to perform point cloud registration of X_t and X_{t+1}. After translation and rotation, the

(a) Beijing United University Small Cyclone automatic vehicle



(b) Campus environment

Fig. 3. Experimental vehicle and experimental environment.

coordinate transformation is obtained. The point cloud coordinates of X_{t+1} and the point cloud coordinates of X_t are converted into the same coordinate system to obtain a valid map at time $t+1$ as Q_{t+1}. The PLICP algorithm is then used to correct the distance between the point clouds of X_t and X_{t+1}. After the correction, the effective map at time $t+1$ is Q'_{t+1}. Once all the received point clouds have undergone coordinate transformations, the point cloud map is finally obtained as Q.

4. Experiment

Several actual vehicle tests were conducted to verify the effectiveness of the algorithm. During the experiments, algorithms processing LiDAR data were run on a laptop computer with 2.5 GHz quad cores and 16 GB memory, using a robot operating system [34] with Linux. The method consumes two cores; the pose and mapping programs run on separate cores. **Fig. 3(a)** shows the experimental platform for the Beijing United University Small Cyclone automatic vehicle, and **Fig. 3(b)** shows the experimental environment within the campus; the red track is the driving route, and yellow indicates the parts where the GPS signal is weak or absent. Through a series of experiments, qualitative and quantitative analyses of the NDT-PLICP, PLICP, and NDT algorithms were conducted.

4.1. Point Cloud Registration Experiments

In this experiment, the automatic vehicle operated at a constant speed of 2.0 m/s. To validate the data, the point cloud registration between adjacent frames was selected to obtain an average transformation matrix. Although the automatic vehicle did not rely on the inertial measurement unit (IMU) for localization, it was compared with the NDT-PLICP, PLICP, and NDT algorithms as reference data. The average matrix of transformation between adjacent frames is as follows:

$$\text{IMU} = \begin{bmatrix} 1.002471 & 0.0241066 & 0.0262786 & 0.0326907 \\ -0.04723 & 1.019864 & 0.017862 & 0.19805 \\ -0.024876 & -0.020504 & 1.0270994 & 2.40136 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\text{PLICP} = \begin{bmatrix} 0.843709 & 0.00204072 & 0.018274 & 0.021079 \\ -0.02645 & 0.84631 & 0.01467 & 0.127684 \\ -0.013677 & -0.012946 & 0.87492 & 1.43784 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\text{NDT} = \begin{bmatrix} 0.674029 & 0.010127 & 0.0172964 & 0.018094 \\ -0.017284 & 0.65372 & 0.012876 & 0.097647 \\ -0.012865 & -0.0114926 & 0.629481 & 1.24861 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\text{NDT-PLICP} = \begin{bmatrix} 0.999574 & 0.0221143 & 0.0190379 & 0.0207708 \\ -0.0224011 & 0.999637 & 0.0149853 & 0.136805 \\ -0.0186996 & -0.0154054 & 0.999706 & 2.10202 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Presented above are the average transformation matrices obtained by IMU, PLICP, NDT, and NDT-PLICP for interframe point cloud registration, including the rotation matrix R and translation matrix T. In this study, based on the valid data provided by the IMU and compared with the value of the IMU matrix, both matrices obtained by the NDT-PLICP algorithm had values closer to those of the IMU matrices.

As summarized in **Table 1**, the average transformation parameters between adjacent frames were calculated: the rotation angle Roll on the X-axis, the rotation angle Pitch on the Y-axis, the rotation angle Yaw on the Z-axis, the translation on the X- and Y-axes, and the error between the translation on the Z-axis and the IMU. The average error between the NDT-PLICP algorithm and the refer-

Table 1. Transform parameter error.

Parameter	PLICP	NDT	NDT-PLICP
Roll [°]	0.004628	0.008417	0.003431
Pitch [°]	0.002006	0.006392	0.001862
Yaw [°]	0.001976	0.004314	0.001643
X [m]	0.000296	0.001205	0.000172
Y [m]	0.000962	0.002746	0.000413
Z [m]	0.001082	0.003589	0.000676

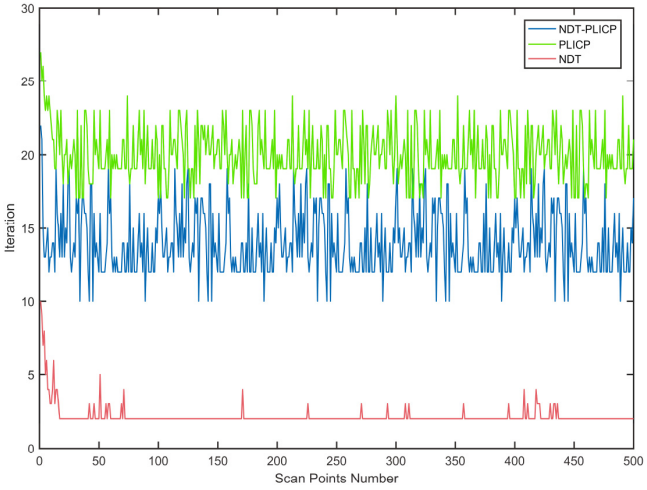


Fig. 4. Point cloud registration iterations.

Table 2. Average number of iterations of point cloud registration.

Algorithm	Average number of iterations
PLICP	20.026
NDT	2.166
NDT-PLICP	13.98

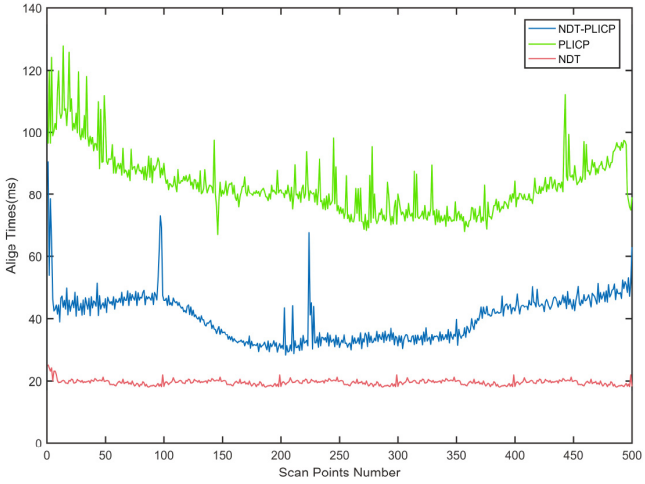


Fig. 5. Runtime of point cloud registration.

ence value IMU were as follows: for Roll [°], 0.003431; for Pitch [°], 0.001862; for Yaw [°], 0.001643; for X [m], 0.000172; for Y [m], 0.000413; and for Z [m], 0.000676. The average errors of the NDT-PLICP algorithm were the smallest. Therefore, it can be concluded that the relative accuracy of the NDT-PLICP algorithm is high.

In this study, the experimental data of the first 500 frames of point cloud registration in the campus environment, the number of iterations of interframe point cloud registration, and the running time of the interframe point cloud registration were analyzed. The number of iterations in the NDT algorithm and the optimized $s(p)$ function were considered as termination conditions for the interframe point cloud registration. The purpose was to register the point cloud between frames coarsely to provide a good initial point cloud pose state for the PLICP algorithm. Then, the PLICP point cloud registration algorithm was used to complete the final point cloud pose correction and achieve the accurate registration of the point cloud between frames.

As shown in **Fig. 4**, in this study, we compared the PLICP point cloud registration algorithm and the point cloud registration algorithm proposed in this paper. The maximum number of iterations set in the NDT algorithm was 10, and the optimization function $s(p)$ was the empirical value in the experimental environment as termination condition. The statistical experimental data help conclude that, in the initial stage of point cloud registration, the calculation efficiency of the initial position of the point cloud is low, and the iteration time of the three algorithms is relatively long. However, as the number of scanning point cloud frames increases, the computational efficiency levels out. The number of iterations of the PLICP algorithm was greater than 16. When the NDT algorithm performed coarse registration of the inter-point cloud, the number of iterations was less than 10. After registration using the NDT algorithm, the number of iterations of the NDT-PLICP was less than 16.

Table 2 summarizes the average number of iterations of the 500-frame point cloud registration of the PLICP, NDT, and NDT-PLICP algorithms. The average number of iterations of the PLICP algorithm was 20.026. The average number of iterations of the NDT algorithm was 2.166. The average number of iterations of the NDT-PLICP algorithm was 13.98. After using the NDT algorithm for the coarse registration of the interframe point cloud, the number of iterations of the interframe point cloud registration by the NDT-PLICP algorithm was reduced by 6.046 compared with that of the PLICP algorithm, and the computational efficiency was improved by approximately 30.19%.

Figure 5 shows the results of the runtime of the point cloud registration (ms). In the initial stage of the point cloud registration, the three algorithms indicated that the initial pose transformation is positively correlated with the number of iterations. The PLICP point cloud registration algorithm had a runtime longer than 80 ms. The runtime of the point cloud coarse registration using the NDT algorithm averaged 20 ms. After the point cloud was coarsely registered by the NDT algorithm, the runtime of the NDT-PLICP point cloud registration was reduced to

Table 3. Average runtime.

Algorithm	Average runtime [ms]
PLICP	83.27074
NDT	19.55676
NDT-PLICP	40.21918

less than 50 ms.

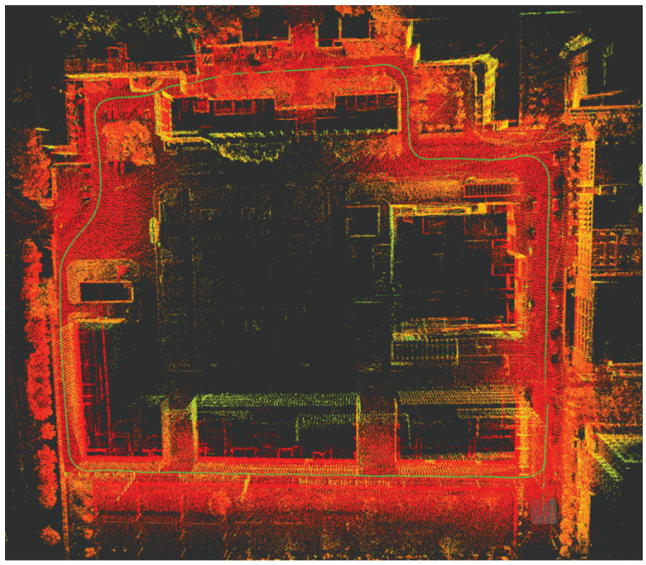
As summarized in **Table 3**, the PLICP, NDT, and NDT-PLICP algorithms continuously registered the average runtime of the 500-frame point cloud. The average runtime of the PLICP algorithm was 83.27074 ms. The average runtime of the NDT algorithm was 19.55676 ms. The average runtime of the NDT-PLICP algorithm was 40.21918 ms. After the point cloud coarse registration was completed using the NDT algorithm, the running time of the NDT-PLICP algorithm was reduced by 43.05156 ms compared with that of the PLICP algorithm, and the calculation efficiency was improved by approximately 51.7%.

The above experimental data analysis indicates that the proposed algorithm improves not only the computational efficiency of the interframe point cloud registration algorithm but also the accuracy of the interframe point cloud registration.

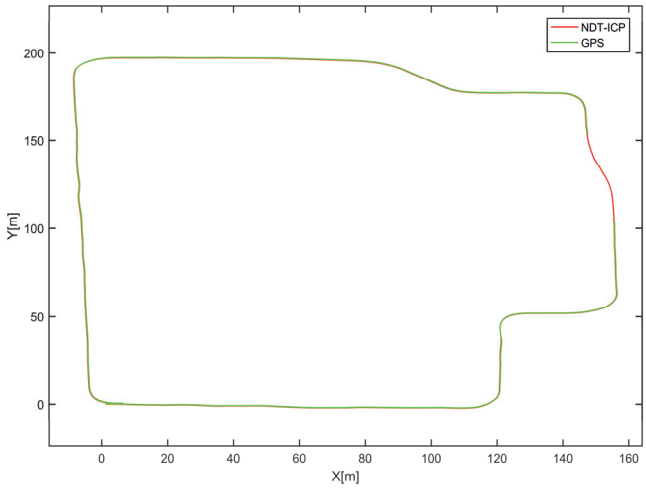
4.2. Localization and Mapping

The experimental environment was a complex campus scene with a total length of 0.75 km. It receives approximately 4500 frames of LiDAR point cloud data and 4500 frames of GPS calibration data. The route includes tall buildings, trees, and overpasses, resulting in weak GPS signals, particularly under the flyover, and an unmanned vehicle cannot receive complete GPS signals. If the unmanned vehicle relies on GPS signals for localization and navigation, an uncontrollable disaster may occur. **Fig. 6(a)** shows the point cloud map formed by the point cloud registration of the algorithm (blue is the localization trajectory), and **Fig. 6(b)** shows the route trajectory of the algorithm after point cloud registration and GPS localization after the calibration track.

As summarized in **Table 4**, the accuracy of the localization of the algorithm was compared according to the real-time kinematics (RTK) state of the GPS. In this experiment, approximately 4500 frames of LiDAR localization data and calibrated GPS localization information were received, and RTK indicated the state of the current GPS information. RTK $= 0$ indicated that no GPS signal was present, and the unmanned vehicle could not trust the result of the current GPS localization, for 430 frames, which was 9.56% of the total distance. $0 <$ RTK ≤ 3 indicated that the GPS signal was relatively weak, for 680 frames or 15.11% of the total distance, and the average localization error of the algorithm was 0.25 m. RTK $= 4$ indicated that the GPS signal ratio was good, for 3390 frames or 75.33% of the total distance, and the average error of the algorithm localization was 0.1 m. This algorithm not only supplemented the GPS-free localization result but also provided more accurate localization results for the unmanned car.



(a) NDT-PLICP point mapping



(b) NDT-ICP and GPS localization trajectories

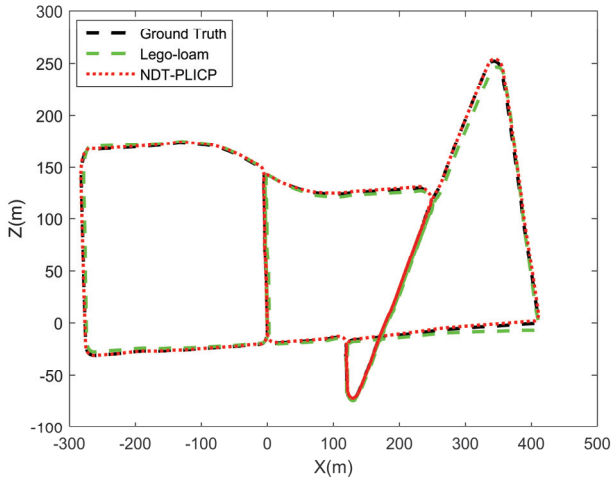
Fig. 6. Localization and mapping.

Table 4. NDT-PLICP and GPS error analysis.

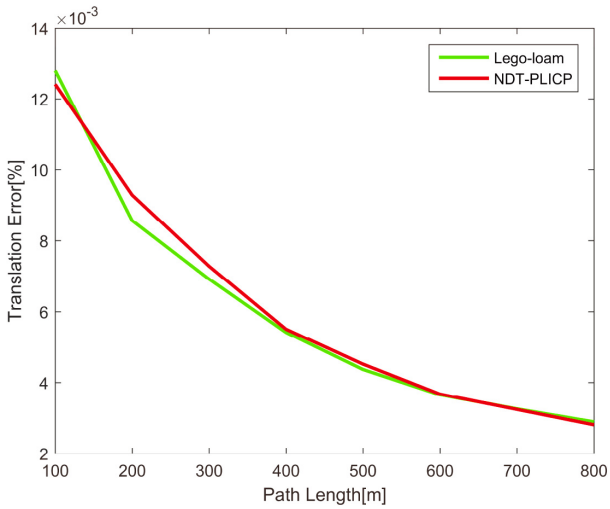
RTK status	Frames	Error	Rate
RTK $= 0$	430		9.56%
$0 <$ RTK ≤ 3	680	0.25 m	15.11%
RTK $= 4$	3390	0.1 m	75.53%

In this study, using the KITTI data, the average translation and rotation errors of the algorithm were analyzed and calculated using the ground-truth experimental results. The error percentage was used for the translation error and the rotation degree was used for the rotation error. The measurement was performed and compared with **Fig. 7(c)** the NDT-PLICP and LeGO-LOAM algorithms.

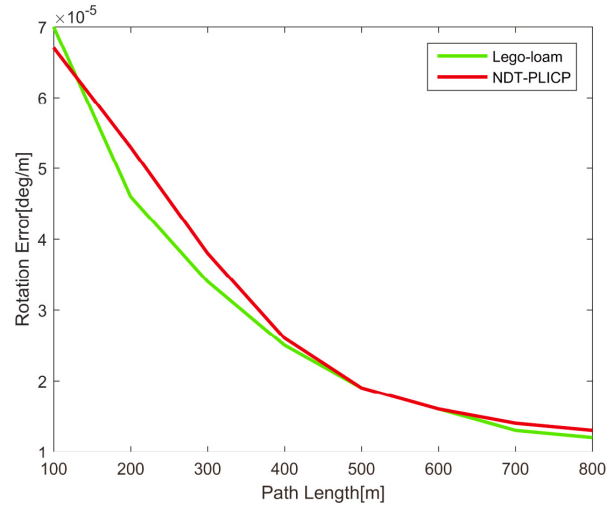
As shown in **Fig. 7(a)**, using the KITTI data for NDT-PLICP and LeGO-LOAM algorithms, and the ground-



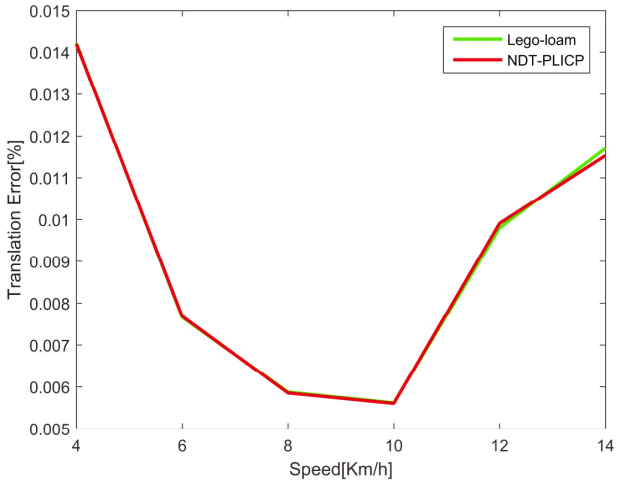
(a) NDT-PLICP, LeGO-LOAM, and ground-truth localization track



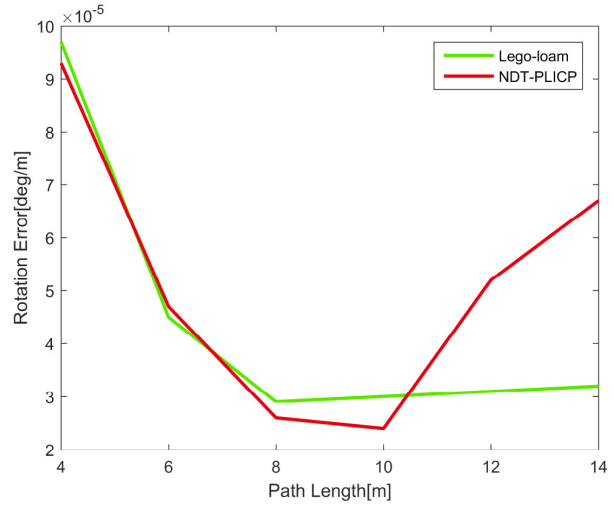
(b) NDT-PLICP and LeGO-LOAM translation errors



(c) NDT-PLICP and LeGO-LOAM rotation errors



(d) NDT-PLICP and LeGO-LOAM rotation errors at different speeds



(e) NDT-PLICP and LeGO-LOAM translation errors at different speeds

Fig. 7. NDT-PLICP and LeGO-LOAM experimental analyses based on KITTI data.

Table 5. Runtime parameter analysis.

Algorithm	Runtime [ms]
LeGO-LOAM	0.1 s
NDT-PLICP	0.04 s

truth localization analysis, the effectiveness of the proposed algorithm could be evaluated. **Figs. 7(b)** and **(c)** show the average translation error and average rotation error of the NDT-PLICP and LeGO-LOAM algorithms via comparison with the ground truth. The average translation errors of the NDT-PLICP and LeGO-LOAM algorithms were 0.005983 and 0.006001, respectively, whereas their average rotation errors were 0.00002937 and 0.00002905, respectively. As shown in **Figs. 7(d)** and **(e)**, the average translation and average rotation errors were experimentally analyzed at different vehicle speeds. The average translation errors of the NDT-PLICP and LeGO-LOAM algorithms at different vehicle speeds are presented. The average translation errors were 0.00912 and 0.00916, respectively; and the average rotation errors were 0.000044 and 0.000046, respectively.

As summarized in **Table 5**, the average running time of the interframe registration of the NDT-ICP algorithm was 0.04 s, and the average running time of LeGO-LOAM was 0.1 s. Under the same specifications of hardware equipment, the calculation efficiency of NDT-ICP was greater by approximately 60%.

Qualitative and quantitative analyses of the above experiments indicate that the NDT-PLICP algorithm improves the efficiency and accuracy of point cloud registration. In particular, when the GPS signal is absent or unreliable, the automatic vehicle can completely rely on LiDAR for precise pose estimation and map construction.

5. Conclusion

In this paper, we proposed a fast point cloud registration algorithm to realize real-time LiDAR localization and map construction and applied it to automatic vehicles in a large number of experiments. First, the computer receives the point cloud data scanned by the vehicle LiDAR, and voxel grid filtering is subsequently performed on the point cloud data to remove the noise points and a large number of redundant point clouds to obtain effective point cloud data for the point cloud registration algorithm. This paper describes a fast point cloud registration algorithm for interframe point cloud registration. The first step is to use the NDT algorithm for point cloud registration to obtain an approximate estimate of the automatic vehicle position. Then, the PLICP algorithm is used to correct the point cloud pose after point cloud registration, so that the interframe point cloud translation distance and rotation angle can reach the optimal registration parameters and an accurate estimation of the automatic vehicle position can be realized. Finally, the cloud registration is accumulated over time, the point cloud information is continuously updated, and the construction of the point cloud map is completed, which provides an a priori map for the autonomous navigation of the automatic vehicle and finally realizes the precise localization of the automatic vehicle.

This study verified the effectiveness of the proposed point cloud registration algorithm through a series of experiments. In the campus experimental environment, the registration accuracy of the proposed algorithm was better than that of other registration algorithms, indicating the precise localization of automatic vehicles. Compared with the PLICP algorithm, the NDT-PLICP algorithm reduced the average number of iterations of point cloud registration between adjacent frames by 6.046, the average running time by 43.05156 ms, and the efficiency of point cloud registration by approximately 51.7%. The computational efficiency of NDT-ICP was approximately 60% higher than that of LeGO-LOAM. Meanwhile, the NDT-PLICP algorithm was compared with GPS, and the accuracy was within 10 cm, which validates the fast and accurate point cloud registration algorithm proposed in this paper.

Acknowledgements

The author(s) disclose the receipt of the following financial support for the research, authorship, and/or publication of this article: This study was financially supported by "Demonstration and verification of high-precision map and fusion positioning (2021YFB2501105)," "Autonomous driving real-time urban road scene understanding based on visual computing (61871039)," "Key technology for multi-view video information acquisition and localization of autonomous vehicle (61871038)," and "Beijing Municipal High-level Innovative Team Construction Plan for High-level Teacher Team Construction Support Program (IDHT20170511)."

References:

[1] K. M. Wurm, C. Stachniss, and G. Grisetti, "Bridging the Gap Between Feature- and Grid-based SLAM," Robotics and Autonomous Systems, Vol.58, No.2, pp. 140-148, 2010.

[2] A. Eliazar and R. Parr, "DP-SLAM: Fast, Robust Simultainous Localization and Mapping Without Predetermined Landmarks," Proc. of the 18th Int. Joint Conf. on Artificial Intelligence (IJCAI-03), pp. 1135-1142, 2003.

[3] S. Thrun, W. Burgard, and D. Fox, "Probabilistic Robotics," MIT Press, 2005.

[4] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the bayes tree," The Int. J. of Robotics Research, Vol.31, pp. 217-236, 2012.

[5] R. Zlot and M. Bosse, "Efficient large-scale 3D mobile mapping and surface reconstruction of an underground mine," The 7th Int. Conf. on Field and Service Robots, 2012.

[6] L. Hengjie, B. Hong, and X. Cheng, "Fast Closed-Loop SLAM based on the fusion of IMU and Lidar," J. of Physics: Conf. Series, 2021 Int. Conf. on Electrical, Electronics and Computing Technology (EECT 2021), Vol.1914, Article No.012019, 2021.

[7] S. Yuan, H. Wang, and L. Xie, "Survey on Localization Systems and Algorithms for Unmanned Systems," Unmanned Systems, Vol.9, No.2, pp. 129-163, 2020.

[8] P. Jiang, L. Chen, H. Guo et al., "Novel indoor positioning algorithm based on Lidar/inertial measurement unit integrated system," Int. J. of Advanced Robotic Systems, Vol.18, No.2, doi: 10.1177/1729881421999923, 2021.

[9] J. Li, X. Zhang, J. Li et al., "Building and optimization of 3D semantic map based on Lidar and camera fusion," Neurocomputing, Vol.409, pp. 394-407, 2020.

[10] X. Guo, Y. Liu, Q. Zhong et al., "Research on Moving Target Tracking Algorithm Based on Lidar and Visual Fusion," J. Adv. Comput. Intell. Intell. Inform, Vol.22, No.5, pp. 593-601, doi: 10.20965/jaciii.2018.p0593, 2018.

[11] Q. Zhong, Y. Liu, X. Guo et al., "Dynamic Obstacle Detection and Tracking Based on 3D Lidar," J. Adv. Comput. Intell. Intell. Inform, Vol.22, No.5, pp. 602-610, doi: 10.20965/jaciii.2018.p0602, 2018.

[12] K. Konolige, M. Agrawal, and J. Solà, "Large-Scale Visual Odometry for Rough Terrain," The 13th Int. Symp. of Robotics Research (ISRR), 2010.

[13] David Nistér, O. Naroditsky, and J. R. Bergen, "Visual odometry for ground vehicle applications," J. of Field Robotics, Vol.23, No.1, pp. 3-20, 2006.

[14] P. J. Besl and N. D. McKay, "A Method for Registration of 3D Shapes," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.14, No.2, pp. 239-256, 1992.

[15] S. Rusinkiewicz and M. Levoy, "Efficient Variants of the ICP Algorithm," Proc. of the 3rd Int. Conf. on 3-D Digital Imaging and Modeling, pp. 145-152, 2001.

[16] A. Censi, "An ICP variant using a point-to-line metric," 2008 IEEE Int. Conf. on Robotics and Automation, doi: 10.1109/ROBOT.2008.4543181, 2008.

[17] Y. Chen and G. Medioni, "Object Modelling by Registration of Multiple Range Images," Image and Vision Computing, Vol.10, No.3, pp. 145-155, 1992.

[18] A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," Proc. of Robotics: Science and Systems, 2009.

[19] R. A. Newcombe, S. Izadi, O. Hilliges et al., "Kinect Fusion: Real-time Dense Surface Mapping and Tracking," Proc. of the IEEE Int. Symp. on Mixed and Augmented Reality, pp. 127-136, 2011.

[20] A. Nuchter, "Parallelization of Scan Matching for Robotic 3D Mapping," Proc. of the 3rd European Conf. on Mobile Robots, 2007.

[21] D. Qiu, S. May, and A. Nuchter, "GPU-Accelerated Nearest Neighbor Search for 3D Registration," Proc. of the Int. Conf. on Computer Vision Systems, pp. 194-203, 2009.

[22] D. Neumann, F. Lugauer, S. Bauer et al., "Real-time RGB-D mapping and 3-D modeling on the GPU using the random ball cover data structure," 2011 IEEE Int. Conf. on Computer Vision Workshops (ICCV Workshops), 2011.

[23] P. Biber and W. Strasser, "The Normal Distributions Transform: A New Approach to Laser Scan Matching," Proc. 2003 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2003), 2003.

[24] M. Magnusson, A. Lilienthal, and T. Duckett, "Scan registration for autonomous mining vehicles using 3D-NDT," J. of Field Robotics, Vol.24, No.10, pp. 803-827, 2007.

[25] R. B. Rusu, Z. C. Marton, N. Blodow, and M. Beetz, "Learning Informative Point Classes for the Acquisition of Object Model Maps," Proc. of the IEEE Int. Conf. on Control, Automation, Robotics and Vision, pp. 643-650, 2008.

[26] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3D Recognition and Pose Using the Viewpoint Feature Histogram," Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 2155-2162, 2010.

[27] M. Bosse and R. Zlot, "Keypoint Design and Evaluation for Place Recognition in 2D Lidar Maps," Robotics and Autonomous Systems, Vol.57, No.12, pp. 1211-1224, 2009.

[28] R. Zlot and M. Bosse, "Efficient Large-scale 3D Mobile Mapping and Surface Reconstruction of an Underground Mine," Proc. of the 8th Int. Conf. on Field and Service Robotics, 2012.

[29] J. Zhang and S. Singh, "LOAM: Lidar Odometry and Mapping in Real-time," Proc. of Robotics: Science and Systems, 2014.

[30] J. Zhang and S. Singh, "Low-drift and Real-time Lidar Odometry and Mapping," Autonomous Robots, Vol.41, No.2, pp. 401-416, 2017.

[31] R. Dube, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Cadena, "SegMatch: Segment Based Place Recognition in 3D Point Clouds," Proc. of the IEEE Int. Conf. on Robotics and Automation, pp. 5266-5272, 2017.

[32] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain," IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), pp. 4758-4765, 2018.

[33] D. Zhu, "Point Cloud Library PCL Learning Course," Beijing Aerospace University Press, 2012.

[34] M. Quigley, B. Gerkey, K. Conley et al., "ROS: An open-source robot operating system," Workshop on Open Source Software (Collocated with ICRA 2009), 2009.

Name:
Qingshan Wang

Affiliation:
Engineer, CATARC (Tianjin) Automotive Engineering Research Institute Co., Ltd.

Address:
No.68 Xianfeng East Road, Dongli District, Tianjin 300300, China

Brief Biographical History:
2017- Bachelor Student, Beijing Union University
2020- Master Student, Beijing Union University
2020- Engineer, CATARC (Tianjin) Automotive Engineering Research Institute Co., Ltd.

Name:
Jun Zhang

Affiliation:
Associate Professor, Wheeled Mobile Robot Department, College of Robotics, Beijing Union University

Address:
No.97 Beisihuan East Road, Chao Yang District, Beijing 100101, China

Brief Biographical History:
2006- Beijing Union University

Main Works:
• "Research on Adaptive Optimal Preview Model Based on Stanley Algorithm," Computer Engineering, Vol.44, No.7, pp. 42-46, 2018.

Membership in Academic Societies:
• Beijing Society of Image and Graphics

Name:
Yuansheng Liu

Affiliation:
Professor and Head, Wheeled Mobile Robot Department, College of Robotics, Beijing Union University

Address:
No.97 Beisihuan East Road, Chao Yang District, Beijing 100101, China

Brief Biographical History:
1996- Beijing Jiaotong University
2008- Beijing Union University

Main Works:
• "Research on moving target tracking algorithm based on lidar and visual fusion," J. Adv. Comput. Intell. Intell. Inform., Vol.22, No.5, pp. 593-601, 2018.
• "A Hybrid Loop Closure Detection Method Based on Lidar SLAM," 2019 Int. Conf. on Computational Intelligence and Security (CIS), 2019.

Membership in Academic Societies:
• Beijing Union University, Academic Committee Member
• China Computer Federation (CCF) Intelligent Vehicle Branch, Executive member

Name:
Xinchen Zhang

Affiliation:
Engineer, Institute of Automation, Chinese Academy of Sciences

Address:
No.95 Zhongguancun East Road, Haidian District, Beijing 100190, China

Brief Biographical History:
2018- Bachelor, Beijing Union University
2021- Master, Beijing Union University
2021- Engineer, Chinese Academy of Sciences