International Journal of Control, Automation, and Systems 21(8) (2023) 2648-2663
http://dx.doi.org/10.1007/s12555-021-1088-7
ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

Towards Optimal Dynamic Localization for Autonomous Mobile Robot via Integrating Sensors Fusion

Jing Li*, Keyan Guo, Junzheng Wang, and Jiehao Li*

Abstract: When it comes to optimal dynamic localization, high accuracy and robustness localization is the main challenge for the autonomous mobile robot. In this paper, an optimal dynamic localization framework with integrating sensors fusion is considered. The global point map is utilized to provide absolute pose observation information, and the multi-sensor information is applied to realize robust localization in complex outdoor environments. The multi-sensor technique, including 3D-Lidar, global positioning system (GPS), and inertial measurement unit (IMU), is adopted to construct the global point map by pose optimization so that the absolute position and attitude observation information can still be provided when the outdoor GPS signal fails. Meanwhile, in the case of optimal localization, the system kinematics equation is constructed by the IMU error model, and the map pose is matched by map scanning. Moreover, the GPS position information participates in multi-source fusion when the GPS signal is reliable. Finally, the experimental results show that the average localization error is within 0.05 meters, reflecting the flexibility of dynamic localization.

Keywords: Autonomous mobile robot, multi-sensor fusion, optimal dynamic localization, point cloud.

1. INTRODUCTION

When it comes to optimal dynamic localization, high accuracy, and robustness localization is the main challenging for the autonomous mobile robot [1-6]. Many localization methods have been applied to the dynamic optimal localization of autonomous mobile robots such as GPS-IMU, SLAM, a priori map-based localization [7-13]. However, these methods often result in unstable localization or even divergence in some scenes, such as satellite signal loss, sparse scene features and dynamic targets in the field of view. Therefore, we propose a multi-sensor fusion localization method to realize the optimal dynamic localization for autonomous mobile robots.

There are many kinds of localization methods applied to the autonomous mobile robot. Global positioning system (GPS) [14] is the most commonly used localization system in autonomous mobile robots. Still, the accuracy of GPS can not be guaranteed in areas with signal occlusion (such as under trees, urban canyons or tunnels) and is affected by the multipath effect [15]. Inertial measurement unit (IMU) uses dead reckoning to estimate the position and attitude. This method does not rely on external signals, and the localization is more accurate in a short time, but it is easy to produce large cumulative error in a long time, resulting in localization failure [16]. Simultaneous localization and mapping (SLAM) has been used in autonomous mobile robot dynamic optimal localization in recent years: it is mainly divided into lidar slam and visual slam. Oriented FAST and rotated BRIEF-SLAM (ORB-SLAM) [17] is a visual slam technology with high performance in pose estimation, but this method will lead to localization failure when affected by illumination changes. Lidar odometry and mapping (LOAM) [18] is a lidar slam localization method. This method is not affected by illumination change and showed high performance in KITTI [19] but has the cumulative error. In some large environments, the test track drift is serious, and the amount of calculation is large. The core idea of map matching localization based on a priori is matching [20-23]. The real-time scanned information of lidar and camera is matched with the pre-constructed map through lidar or camera for pose estimation. The calculation speed is better than slam [20], but it is easy to cause localization failure in some scenes with sparse features or an insufficient number of features [24,25]. Therefore, comple-

Manuscript received December 23, 2021; revised July 6, 2022; accepted August 10, 2022. Recommended by Associate Editor Jongho Lee under the direction of Senior Editor Chan Gook Park. This work was supported by the National Key Research and Development Program of China under Grant 2019YFC1511401, and the National Natural Science Foundation of China under Grant 62173038 and 62203176.

Jing Li, Keyan Guo, and Junzheng Wang are with the State Key Laboratory of Intelligent Control and Decision of Complex Systems, School of Automation, Beijing Institute of Technology, Beijing 100081, China (e-mails: {bitljing, 3220190680}@bit.edu.cn, zjwang221@gmail.com). Jiehao Li is with the Key Laboratory of Key Technology on Agricultural Machine and Equipment, Ministry of Education, College of Engineering, South China Agricultural University, Guangzhou 510642, China, and also with the State Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing Institute of Technology, Beijing 100081, China (e-mail: jiehao.li@ieee.org).
* Corresponding authors.

menting the advantages of multiple sensor information to realize the optimal dynamic localization of autonomous mobile robots is particularly important.

Kalman filter [26,27] is usually used to fuse multi-sensor information for dynamic optimal estimation. This method [16] is used to fuse GPS and IMU information for localization. The main principle is to correct the cumulative interval error of dead reckoning by using absolute position reading [16]. Still, this method can only last for about one minute when the GPS signal is bad. Nevertheless, the result of fusing two sensors far exceeds the performance of independent GPS or IMU. Lightweight and ground-optimized lidar odometry and mapping (LEGO-LOAM) [28] first uses IMU motion information to correct point cloud distortion, thereby calculating the distance between the current frame and historical keyframes for closed-loop detection, and then performing pose estimation. However, it is easy to cause localization failure in some scenes with sparse features. Tightly-coupled lidar inertial odometry via smoothing and mapping (LIO-SAM) [29] proposed to add GPS, IMU and other sensor information as factor constraints to optimize localization. However, in some large-scale scenarios, the amount of calculation is complex, and the real-time positioning is poor. A lidar-inertial state estimator for robust and efficient navigation (LINS) [30] used error state Kalman filter (ESKF) [31] to estimate the pose of the robot, and used the error state as the state variable to avoid the risk of over parameterization. At the same time, the error state quantity was small, far away from parameter singularity and universal lock problems, and maintained the effective linearization state. For the optimal dynamic localization of autonomous mobile robots, it is a challenging problem to fuse sufficient multi-sensor information while considering the calculation speed and accuracy.

For the optimal dynamic localization of autonomous mobile robots, the main contributions of this paper are summarized as follows:

1) An optimal dynamic localization is proposed, which uses the global point map to provide absolute pose observation information, integrating multi-sensor information to realize robust localization in complex outdoor environments.
2) The multi-sensor technique, including 3D-Lidar, GPS and IMU, is utilized to construct the global point map by pose optimization so that the absolute position and attitude observation information can still be provided when the outdoor GPS signal fails.
3) In order to achieve optimal localization, the system kinematics equation is constructed by IMU error model, and the map pose is matched by map scanning. At the same time, the GPS position information is reliable when the signal is reliable are fused.

The remaining structure of this paper is as follows: Section 2 introduces the dynamic optimal positioning method of mobile robot based on multi-sensor fusion. Section 3 presents the hardware platform used in this experiment and analyzes the result under KITTI and actual test scenarios. Finally, Section 4 summarizes the results of this paper and summarizes the future work.

2. OPTIMAL DYNAMIC LOCALIZATION FRAMEWORK

The optimal dynamic localization flow chart for autonomous mobile robot via integrating sensors fusion proposed in this paper is shown in Fig. 1. Firstly, the global map is constructed through lidar slam, and then the observation pose is obtained based on the scanning matching between lidar and global map. Finally, the output trajectory is optimized by integrating the IMU error model, map

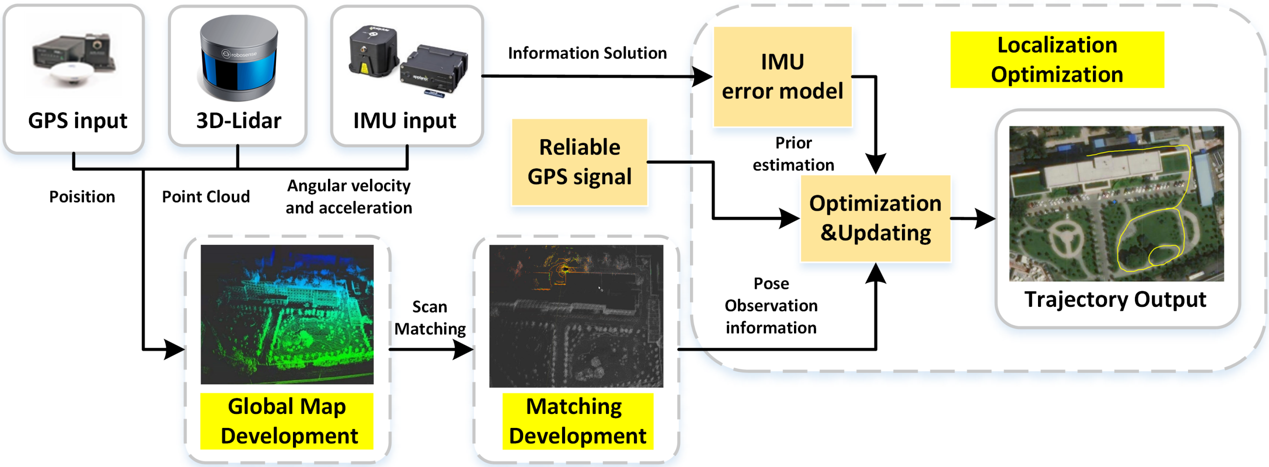


Fig. 1. Localization framework for autonomous mobile robot via integrating sensors fusion.

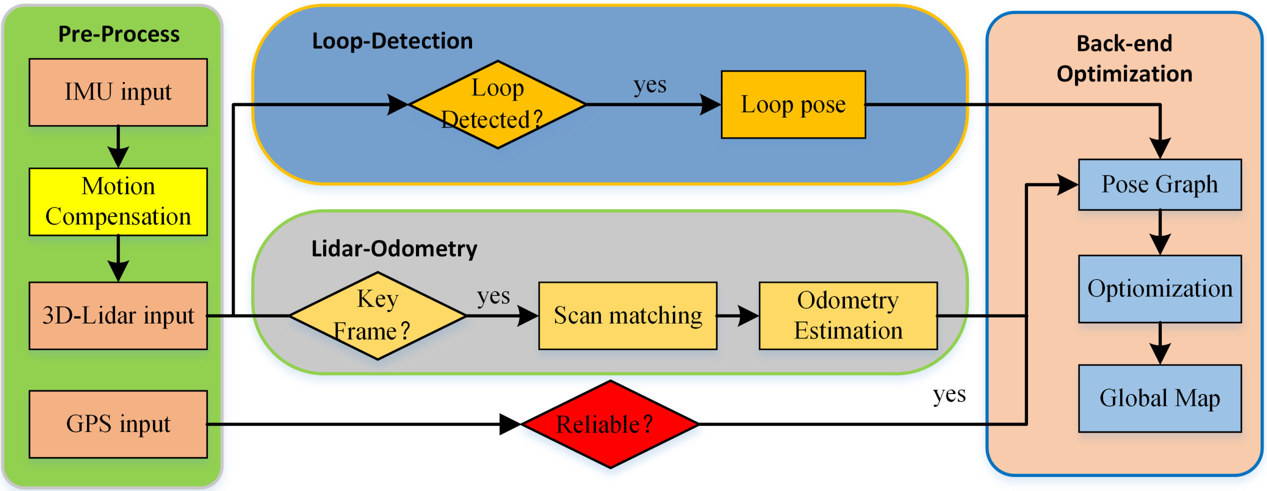


Fig. 2. Optimal dynamic localization process.

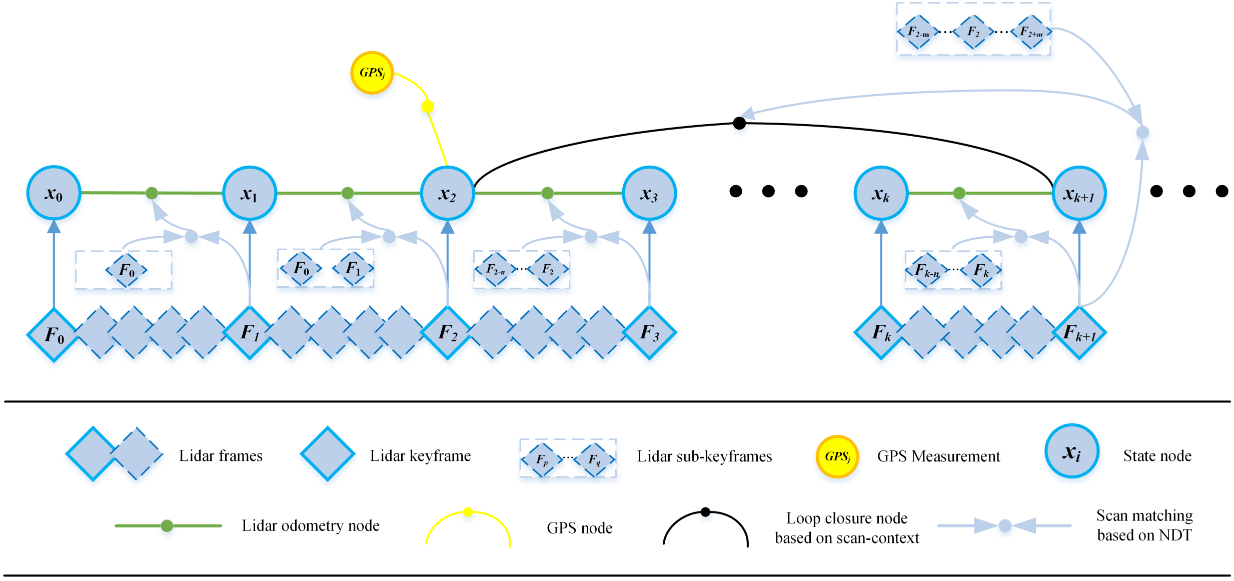


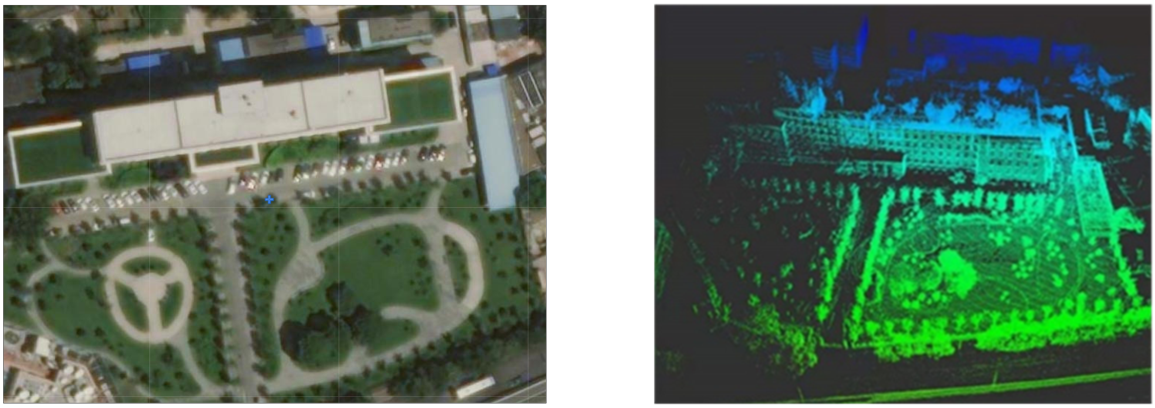
Fig. 3. Mapping optimization flow chart.

matching posture and position information under the reliability of some GPS signals.

2.1. Global map development

Mapping construction: This section introduces the construction process of the global point cloud map based on the optimization of the pose map. It mainly includes four modules: sensor data preprocessing, lidar odometry, loop detection, and back-end optimization. Sensor data preprocessing mainly synchronizes the information of each sensor in time and space. Lidar odometry is used primarily to splice the point clouds of each frame into the same coordinate system according to the position and attitude obtained by real-time scanning and matching of lidar. Loop detection mainly judges the similarity between point cloud frames and then constructs loop constraints to reduce the cumulative error. Finally, the back-end optimization constructs the pose map according to each constraint and optimizes the global map. The overall process is shown in Fig. 2.

Mapping optimization: In order to eliminate accumulated errors and map ghosting, loop detection [32] constraints are added, and GPS information is added as the observation information of the pose map for optimization when the GPS information is reliable. A general framework for graph optimization (G2O) [33] is used as the basic framework for graph optimization of the pose graph. The graph comprises nodes and edges, and the structure is shown in Fig. 3. When the lidar odometry is scanned and matched, we select a keyframe every five frames, splice



(a)

(b)

Fig. 4. Global point cloud map construction results. (a) Google Earth. (b) Global point cloud map.

the keyframes into the same coordinate system, and combine the scanned keyframes with local lidar sub-keyframes to reduce the amount of calculation during map construction. In loop detection, in order to consider speed and accuracy, loop detection is carried out every 50 m. The current keyframe is matched with the local map spliced by local lidar sub-keyframes, and the splicing range of sub-keyframes is set to 5 frames. A residual function is constructed during optimization by combining each observation information and state for optimization. The residual function is shown in

$$Z(x) = \sum_{x=0}^{N} e_k\left(x_k, G_k\right)^T \Omega_k e_k\left(x_k, G_k\right). \tag{1}$$

The goal of this figure is to minimize the error of all measurements. Then, the optimization problem can be expressed as

$$x^* = \arg\min_{x} Z(x), \tag{2}$$

where x_k represents the state node. G_k and Ω_k respectively are the mean value of the constraint of node x_k and the information matrix of the current constraint of x_k, and $e_k\left(x_k, G_k\right)$ denotes the error function of x_k and the observation information G_k.The Gauss-Newton (G-N) optimization method is used for optimization to minimize the sum of residual functions,which mainly divided into binary edge optimization and unary edge optimization. The global point cloud map construction results are shown in Fig. 4.

2.1.1 Binary edge optimization

The binary edge means that the loop closure detects the relative pose between two frames as the difference between the observation information and the laser odometry matching pose between two consecutive frames to construct a residual function for optimization.

Assuming that there is a loop between the i-th frame and the j-th frame, the relative pose can be expressed as

$$T_{ij} = T_i^{-1} T_j, \tag{3}$$

where T_i and T_j represent the absolute pose observations of the i-th frame and the j-th frame, respectively, and T_{ij} represents the relative pose observations of the i-th frame and the j-th frame. In Lie algebra, it is expressed as

$$\begin{aligned} \xi_{ij} &= \ln\left(T_i^{-1} T_j\right)^{\vee} \\ &= \ln\left(\exp((-\xi_i)^{\wedge}) \exp(\zeta_j^{\wedge})\right)^{\vee}, \end{aligned} \tag{4}$$

$$\begin{cases} \xi = \begin{bmatrix} \rho \\ \phi \end{bmatrix} \in R^6, \\ \xi^{\wedge} = \begin{bmatrix} \phi^{\wedge} & \rho \\ 0^T & 0 \end{bmatrix} \in R^{4\times4}, \\ \phi^{\wedge} = \begin{bmatrix} 0 & -\phi_3 & \phi_2 \\ \phi_3 & 0 & -\phi_1 \\ -\phi_2 & \phi_1 & 0 \end{bmatrix} \in R^{3\times3}, \end{cases} \tag{5}$$

where ξ denotes an element in each lie algebra, which indicates a six-dimensional vector, the first three dimensions are translation, denoted by ρ, and the latter three dimensions are rotation, denoted by $\phi.\xi^{\wedge} \in R^{4\times4}$ represents the four-dimensional matrix corresponding to the six-dimensional vector.ϕ represebts the vector of lie algebra corresponding to lie group SO(3) defined on R^3, $\phi^{\wedge} \in R^{3\times3}$ denotes the vector corresponds to the antisymmetric matrix.

Ideally, (3) = (4), but when there is an error in the pose, use the left and right ends of the equation to calculate the residual term

$$\begin{aligned} e_{ij} &= \ln\left(T_{ij}^{-1} T_i^{-1} T_j\right)^{\vee} \\ &= \ln\left(\exp((-\xi_{ij})^{\wedge}) \exp((-\xi_i)^{\wedge}) \exp(\xi_j^{\wedge})\right)^{\vee}. \end{aligned} \tag{6}$$

Add perturbations $\delta\xi_i$ and $\delta\xi_j$ to the poses of the i-th frame and the j-th frame respectively to solve the Jacobian matrix, and use the adjoint property and the BCH formula to simplify. The difference is expressed as

$$\begin{aligned}\hat{e}_{ij} &= \ln(T_{ij}^{-1}T_i^{-1}\exp((-\delta\xi_i)^\wedge)\exp(\delta\xi_j^\wedge)T_j)^\vee\\ &= \ln(T_{ij}^{-1}T_i^{-1}\exp((-\delta\xi_i)^\wedge)T_j\\ &\quad\times\exp((Ad(T_j^{-1})\delta\xi_j)^\wedge))^\vee\\ &= \ln(\exp(e_{ij})\exp((-Ad(T_j^{-1})\delta\xi_i)^\wedge\\ &\quad+(Ad(T_j^{-1})\delta\xi_j)^\wedge)^\vee\\ &\approx e_{ij}-J_r^{-1}(e_{ij})Ad(T_j^{-1})\delta\xi_i\\ &\quad+J_r^{-1}(e_{ij})Ad(T_j^{-1})\delta\xi_j. \end{aligned}\tag{7}$$

The Jacobian matrix of the residuals about T_i T_j are as follows:

$$A_{ij} = \frac{\partial e_{ij}}{\partial\delta\xi_i} = -J_r^{-1}(e_{ij})Ad(T_j^{-1}),\tag{8}$$

$$\begin{cases} B_{ij} = \dfrac{\partial e_{ij}}{\partial\delta\xi_j} = J_r^{-1}(e_{ij})Ad\left(T_j^{-1}\right),\\ J_r^{-1}(e_{ij}) \approx I+\dfrac{1}{2}\begin{bmatrix}\phi_e^\wedge & \rho_e^\wedge\\ 0 & \phi_e^\wedge\end{bmatrix}.\end{cases}\tag{9}$$

Performing first-order taylor expansion on the residual to get (10) as follows:

$$\begin{aligned}e_{ij}(x_i+\Delta x, x_j+\Delta x) &= e_{ij}(x+\Delta x)\\ &\approx e_{ij}+J_{ij}\Delta x,\end{aligned}\tag{10}$$

where J_{ij} represents the Jacobian matrix of the residual with respect to the pose

$$J_{ij} = (0\cdots0\ \underbrace{A_{ij}}_{\text{node } i}\ 0\cdots0\ \underbrace{B_{ij}}_{\text{node } j}\ 0\cdots0).\tag{11}$$

For each residual term there is the following relationship

$$\begin{aligned}F_{ij}(x+\Delta x) &= e_{ij}(x+\Delta x)^T\Omega_{ij}e_{ij}(x+\Delta x)\\ &\approx (e_{ij}+J_{ij}\Delta x)^T\Omega_{ij}(e_{ij}+J_{ij}\Delta x)\\ &= \underbrace{e_{ij}^T\Omega_{ij}e_{ij}}_{c_{ij}}+2\underbrace{e_{ij}^T\Omega_{ij}J_{ij}}_{b_{ij}^T}\Delta x\\ &\quad+\Delta x^T\underbrace{J_{ij}^T\Omega_{ij}J_{ij}}_{H_{ij}}\Delta x\\ &= c_{ij}+2b_{ij}^T\Delta x+\Delta x^T H_{ij}\Delta x.\end{aligned}\tag{12}$$

The total residual can be expressed as

$$\begin{aligned}F(x+\Delta x) &= \sum_{<i,j>\in c} F_{ij}(x+\Delta x)\\ &= \sum_{<i,j>\in c}\left(c_{ij}+2b_{ij}^T\Delta x+\Delta x^T H_{ij}\Delta x\right)\\ &= c+2b^T\Delta x+\Delta x^T H\Delta x.\end{aligned}\tag{13}$$

The above optimization problem is transformed into finding Δx so that $\Delta F(x)$ achieves a minimum value. Let its derivative be zero

$$\frac{d\Delta F(x)}{d\Delta x} = 2b+2H\Delta x = 0,\tag{14}$$

$$H_i\Delta x = -b.\tag{15}$$

Therefore, it only needs to get J_{ij} and can get Δx, as in (10). Then, according to the correction amount, the value of x is corrected. That is, one iteration is completed, and the iteration is terminated for multiple iterations until the residual meets the convergence condition, and the optimization is completed.

2.1.2 Unary edge optimization

Unary edge refers to GPS prior position observation information, which do not connect two pose states like pose nodes between keyframes, but only connect an observation of a pose state quantity. Therefore, its corresponding residual is the difference between the observed value and the state quantity, namely

$$\begin{aligned}e_i &= \ln(Z_i^{-1}T_i)^\vee\\ &= \ln(\exp((-\xi_{zi})^\wedge)\exp(\xi_i^\wedge))^\vee,\end{aligned}\tag{16}$$

where Z_i represents the prior observations, and e_i represents the residuals corresponding to the prior observations. Add perturbation $\delta\xi_i$ to the residual, and use the adjoint property and the BCH formula to simplify

$$\begin{aligned}\hat{e}_i &= \ln(Z_i^{-1}\exp(\delta\xi_i^\wedge)T_i)^\vee\\ &= \ln(Z_i^{-1}T_i\exp((Ad(T_i^{-1})\delta\xi_i)^\wedge))^\vee\\ &= \ln(\exp(e_i)\exp((Ad(T_i^{-1})\delta\xi_i)^\wedge))^\vee\\ &\approx e_i+J_r^{-1}(e_i)Ad(T_i^{-1})\delta\xi_i.\end{aligned}\tag{17}$$

The Jacobian matrix of the residuals about T_i is

$$\begin{cases} \dfrac{\partial e_{ij}}{\partial\delta\xi_i} = J_r^{-1}(e_{ij})Ad\left(T_j^{-1}\right),\\ J_r^{-1}(e_{ij}) \approx I+\dfrac{1}{2}\begin{bmatrix}\phi_e^\wedge & \rho_e^\wedge\\ 0 & \phi_e^\wedge\end{bmatrix}.\end{cases}\tag{18}$$

The subsequent derivation can be analogous to the binary edge optimization correction derivation process.

2.2. Matching development

This section mainly introduces the pose obtained by matching the real-time scanning information of lidar with the global point cloud map. First, we make a sparsity of global map files to improve the speed of loading maps. Then the initial pose is obtained by GPS or the feature descriptor of the key frame saved during mapping. In order to improve the speed of map matching, the global map is split into the local map by crop box filter according

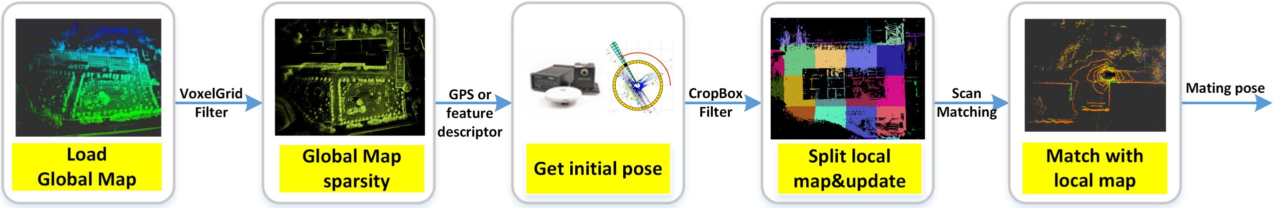


Fig. 5. Map matching flow chart.

to the real-time pose. Finally, the point cloud information scanned by lidar in real-time is matched with the local map to output the map matching pose. The scanning matching algorithm adopts normal distribution transform (NDT) [34,35], which can save complete information better than LOAM [18,28].The map matching flow chart is shown in Fig. 5.

The matching localization algorithm:

1) First, voxel the point cloud map. The point cloud space is divided into a set of voxels with a specified size, and then the mean and covariance of each voxel point cloud in the point cloud map are calculated to construct a Gaussian distribution.

$$\mu=\frac{1}{N_x}\sum_{i=1}^{N_x}x_i, \tag{19}$$

$$\Sigma=\frac{1}{N_x}\sum_{i=1}^{N_x}(x_i-\mu)(x_i-\mu)^T, \tag{20}$$

where $X=\{x_1, x_2, \cdots, x_{N_x}\}$ is the point set of the point cloud map. x_i $(i=1, \cdots, N_x)$ denotes the coordinates of the points in the point cloud map, and N_x represents the number of points in the point cloud map.

2) According to the initial pose between the current frame point cloud and the map, the current frame point cloud is transformed into the map coordinate system, and the joint probability of all points is calculated.

$$y_i'=T(p,y_i)=Ry_i+t, \tag{21}$$

$$f(X,y_i')=\frac{1}{\sqrt{2\pi}\sqrt{|\Sigma|}}\times\exp\left(-\frac{(y_i'-\mu)^T\Sigma^{-1}(y_i'-\mu)}{2}\right), \tag{22}$$

$$\Psi=\prod_{i=1}^{N_y}f(X,T(p,y_i))=\prod_{i=1}^{N_y}\frac{1}{\sqrt{2\pi}\sqrt{|\Sigma|}}\exp\left(-\frac{(y_i'-\mu)^T\Sigma^{-1}(y_i'-\mu)}{2}\right), \tag{23}$$

where $Y=\{y_1, y_2, \cdots, y_{N_y}\}$ represents the point cloud point set of the current frame. y_i $(i=1, \cdots, N_y)$ denotes the coordinates of the points in the point cloud of the current frame, and N_y is the number of points in the point cloud of the current frame. y_i' represents the coordinate conversion function of the current frame point cloud to the point cloud map,and $T(p,y_i)$ is the coordinate of the point converted from the current frame point cloud to the map coordinate system, $p=[t_x, t_y, t_z, \phi_x, \phi_y, \phi_z]^T$ and $t_x, t_y, t_z, \phi_x, \phi_y, \phi_z$ are the translation and rotation in the three directions of xyz, respectively, R and t represent the rotation matrix and the translation vector respectively. $f(X,y_i')$ is the joint probability of a single point, and $\Psi=\prod_{i=1}^{N_y}f(X,T(p,y_i))$ indicates the joint probability of all points.

3) When the joint probability of all points is the largest, it can be considered that the map matching is quasi-successful, and the optimal pose information is obtained at this time. Take the logarithm of the joint probability function to transform the original problem into an optimization problem

$$\ln\Psi=\prod_{i=1}^{N_y}\left(\ln\left(\frac{1}{\sqrt{2\pi}\sqrt{|\Sigma|}}\right)+\left(-\frac{(y_i'-\mu)^T\Sigma^{-1}(y_i'-\mu)}{2}\right)\right), \tag{24}$$

$$\max\Psi\sim\max\ln\Psi\sim\min\Psi_1=\min\sum_{i=1}^{N_y}(y_i'-\mu)^T\Sigma^{-1}(y_i'-\mu). \tag{25}$$

Let $e_i(p)=y_i'-u$, $F_i(p)=e_i^T(p)\Sigma^{-1}e_i(p)$, and the objective function is

$$\min\sum_{i=1}^{N_y}(y_i'-\mu)^T\Sigma^{-1}(y_i'-\mu)=\min\sum_{i=1}^{N_y}F_i(p). \tag{26}$$

Through Gauss-Newton method iterative optimization, find Δp to minimize the value of (26), Taylor expand $e_i(p+\Delta p)$ and $F_i(p+\Delta p)$ to get (27) and (28)

$$e_i(p+\Delta p)\approx e_i(p)+\frac{de_i}{dp}\Delta p=e_i(p)+J_i(p)\Delta p, \tag{27}$$

$$\begin{aligned}F_i(p+\Delta p)&=e_i(p+\Delta p)^T\Sigma^{-1}e_i(p+\Delta p)\\&\approx(e_i(p)+J_i(p)\Delta p)^T\Sigma^{-1}(e_i(p)+J_i(p)\Delta p)\\&=e_i^T\Sigma^{-1}e_i+2e_i^T\Sigma^{-1}J_i(p)\Delta p\\&\quad+\Delta p^TJ_i^T(p)\Sigma^{-1}J_i(p)\Delta p\end{aligned}$$

$$= F_i(p) + 2b_i^T \Delta p + \Delta p^T H_i \Delta p, \tag{28}$$

$$\begin{cases} b_i^T = e_i^T \Sigma^{-1} J_i(p), \\ H_i = J_i^T(p) \Sigma^{-1} J_i(p), \end{cases} \tag{29}$$

$$\begin{aligned} \Delta F_i(p) &= F_i(p + \Delta p) - F_i(p) \\ &= 2b_i^T \Delta p + \Delta p^T H_i \Delta p. \end{aligned} \tag{30}$$

The above optimization problem is transformed into finding Δp to make $\Delta F_i(p)$ obtain the minimum value, let $\Delta F_i(p)$ derivative be zero

$$\frac{d\Delta F_i(p)}{d\Delta p} = 2b_i + 2H_i \Delta p = 0, \tag{31}$$

$$H_i \Delta p = -b_i. \tag{32}$$

So just get $J_i(p)$ to get Δp, and the matching pose information is finally output.

$$J_i(p) = \frac{de_i}{dp}, \tag{33}$$

$$\begin{aligned} y_i^{'} &= T(p, y_i) \\ &= R_x R_y R_z y_i + t \\ &= \begin{bmatrix} c_y c_z & -c_y s_z & s_y \\ c_x s_z + s_x s_y c_z & c_x c_z - s_x s_y s_z & -s_x c_y \\ s_x s_z - c_x s_y c_z & s_x c_z + c_x s_y s_z & c_x c_y \end{bmatrix} y_i + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}, \end{aligned} \tag{34}$$

$$J_i(p) = \begin{bmatrix} 1 & 0 & 0 & 0 & c & f \\ 0 & 1 & 0 & a & d & g \\ 0 & 0 & 1 & b & e & h \end{bmatrix}, \tag{35}$$

$$\begin{cases} a = y_{i1}(-s_x s_z + c_x s_y c_z) + y_{i2}(-s_x c_z - c_x s_y s_z) \\ \quad + y_{i3}(-c_x c_y), \\ b = y_{i1}(c_x s_z + s_x s_y c_z) + y_{i2}(-s_x s_y s_z + c_x c_z) \\ \quad + y_{i3}(-s_x c_y), \\ c = y_{i1}(-s_y c_z) + y_{i2}(s_y s_z) + y_{i3}(c_y), \\ d = y_{i1}(s_x c_y c_z) + y_{i2}(-s_x c_y s_z) + y_{i3}(s_x s_y), \\ e = y_{i1}(-c_x s_y c_z) + y_{i2}(c_x c_y s_z) + y_{i3}(-c_x s_y), \\ f = y_{i1}(-c_y s_z) + y_{i2}(-c_y s_z), \\ g = y_{i1}(c_x c_z - s_x s_y s_z) + y_{i2}(-c_x s_z - s_x s_y c_z), \\ h = y_{i1}(s_x c_z + c_x s_y s_z) + y_{i2}(c_x s_y c_z - s_x s_z), \end{cases} \tag{36}$$

$$\begin{cases} c_x = \cos\phi_x, \\ s_x = \sin\phi_x, \\ c_y = \cos\phi_y, \\ s_y = \sin\phi_y, \\ c_z = \cos\phi_z, \\ s_z = \sin\phi_z. \end{cases} \tag{37}$$

2.3. Localization optimization

This section introduces how to fuse multi-sensor information to optimize dynamic localization, which is mainly divided into solving IMU information, constructing IMU error model, and generating dynamic optimal localization information through error state Kalman filter. The task of the 3D-LiDAR , GPS and IMU fusion is to fuse and optimize the localization information of IMU, 3D-LiDAR and point cloud map matching, and GPS to generate new positioning information and enhance the robustness of the localization information. First, the IMU measurement information is solved to construct an error model, and the system error state equation is constructed based on the IMU error model. Second, the position and attitude observation information obtained by matching the 3D-LiDAR with the global point cloud map and the position observation information of GPS in the reliable signal area are used to construct the observation equation. In order to reduce the nonlinearization error of the extended Kalman filter, the error state quantity is introduced to build the error state Kalman filter to optimize the fusion of the 3D-LiDAR, GPS and IMU information and output the positioning information, so as to realize the robust and efficient positioning of the unmanned vehicle in the outdoor complex environment. The overall flow chart is shown in Fig. 6.

IMU solution: Three-dimensional rigid body motion includes translation and rotation. Rotation mainly includes rotation matrix, rotation vector and quaternion. By solving the differential equation of three-dimensional rigid body motion, we can obtain the solution of the differential equation to solve the IMU measurement information, mainly including the rotation matrix differential equation, quaternion differential equation and rotation vector differential equation. The attitude, velocity and position of IMU are solved by solving differential equations.

The definition of quaternion operator $\otimes$ and pose transformation operator $\odot$ are as follows:

1) If we have two complex numbers $A = a + bi$ and $B = c + di$, then constructing $q = A + Bj$ yields a number in the space of quaternions H.

$$\begin{aligned} q &= q_w + q_v \\ &= q_w + q_w \boldsymbol{i} + q_w \boldsymbol{j} + q_w \boldsymbol{k}, \end{aligned} \tag{38}$$

where $\boldsymbol{k} = ij$, q_w represents the real part of the quaternion and q_v is the imaginary part of the quaternion. Quaternion multiplication operator $\otimes$ is defined as follows:

$$\boldsymbol{p} \otimes \boldsymbol{q} = \begin{bmatrix} p_w q_w - \boldsymbol{p}_v^T \boldsymbol{q}_v \\ p_w \boldsymbol{q}_v + q_w \boldsymbol{p}_v + \boldsymbol{p}_v \times \boldsymbol{q}_v \end{bmatrix}. \tag{39}$$

2) Suppose there is a vector a in the three-dimensional space, and its coordinates in the two coordinate systems before and after rotation (the coordinate system before rotation is the world coordinate system w, and the coordinate system after rotation is the carrier system b) are $[a_1, a_2, a_3]^T$ and $[a_1', a_2', a_3']^T$, then there are

$$[\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3] \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = [\boldsymbol{e}_1', \boldsymbol{e}_2', \boldsymbol{e}_3'] \begin{bmatrix} a_1' \\ a_2' \\ a_3' \end{bmatrix}, \tag{40}$$

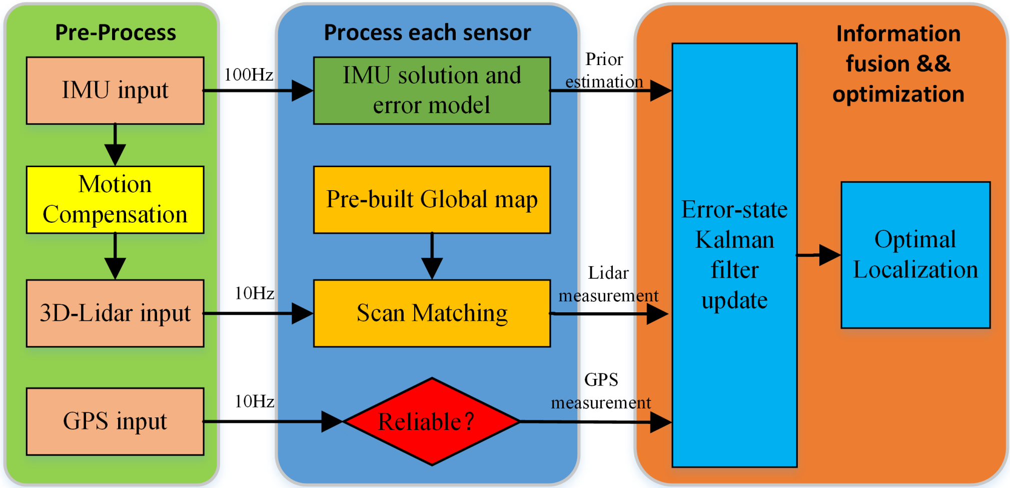


Fig. 6. Map matching flow chart.

where $[e_1, e_2, e_3]^T$ and $[e'_1, e'_2, e'_3]^T$ are the unit orthonormal basis (coordinate system) before and after the rotation, which can be obtained

$$a = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} \boldsymbol{e}_1^T e'_1 & \boldsymbol{e}_1^T \boldsymbol{e}'_2 & \boldsymbol{e}_1^T \boldsymbol{e}'_3 \\ \boldsymbol{e}_2^T \boldsymbol{e}'_1 & \boldsymbol{e}_2^T \boldsymbol{e}'_2 & \boldsymbol{e}_2^T \boldsymbol{e}'_3 \\ \boldsymbol{e}_3^T e'_1 & \boldsymbol{e}_1^T \boldsymbol{e}'_2 & \boldsymbol{e}_3^T \boldsymbol{e}'_3 \end{bmatrix} = \begin{bmatrix} a'_1 \\ a'_2 \\ a'_3 \end{bmatrix} = R_{wb} a', \tag{41}$$

where R_{wb} denotes the rotation matrix, the determinant is 1. The pose transformation operator $\odot$ is explained as above.

Let there be a fixed vector r^w in the world coordinate system (W coordinate system) and B in the robot coordinate system (b coordinate system), then

$$r^w = R_{wb} \cdot r^b, \tag{42}$$

$$r^w = q_{wb} \otimes r^b \otimes q^*_{wb}, \tag{43}$$

where R_{wb} and q_{wb} are the rotation matrix and quaternion of the robot coordinate system relative to the world coordinate system. q^*_{wb} denotes the conjugate quaternion of q_{wb} , and $\otimes$ is the quaternion multiplication. Equation (42) can be differentiated on both sides

$$\dot{r}^w = R_{wb} \dot{r}^b + \dot{R}_{wb} r^b. \tag{44}$$

Because of $\dot{r}^w = 0$, $\dot{r}^b = -w^b_{wb} \times r^b$, where w^b_{wb} is the angular velocity of the robot rotation. Deduced the differential equation of the rotation matrix

$$\dot{R}_{wb} = R_{wb} \left[w^b_{wb} \right]_\times, \tag{45}$$

where $\left[w^b_{wb} \right]_\times$ represents the antisymmetric matrix corresponding to the vector rotation angular velocity of the robot. After multiplying both sides of (43) by q_{wb}, it can be obtained through differentiation

$$\dot{r}^w \otimes q_{wb} + r^w \otimes \dot{q}_{wb} = \dot{q}_{wb} \otimes r^b + q_{wb} \otimes \dot{r}^b. \tag{46}$$

Because of $\dot{r}^w = 0$ and $\dot{r}^b = -w^b_{wb} \times r^b = -w^b_{wb} \otimes r^b$, the quaternion differential equation can be obtained by sorting and deriving, as in (47)

$$\dot{q}_{wb} = q_{wb} \otimes \frac{1}{2} \begin{bmatrix} 0 \\ w^b_{wb} \end{bmatrix}, \tag{47}$$

$$R_{wb} = I + \frac{\sin \phi}{\phi} \left(\phi^\wedge \right) + \frac{1 - \cos \phi}{\phi^2} \left(\phi^\wedge \right)^2. \tag{48}$$

According to Rodriguez formula as (48), substituting (13) and simplification can obtain the rotation vector differential equation as

$$\phi = w^b_{wb} + \frac{1}{2} \phi \times w^b_{wb}, \tag{49}$$

where w^b_{wb} and a^b_{wb} are the three-axis angular velocity and three-axis acceleration measured in the robot coordinate system of the IMU relative to the world coordinate system W. The attitude is solved by solving the quaternion differential equation and the rotation vector differential equation.

According to (45), we can obtain (35) for the rotation matrix at t_{k-1} and t_k

$$R_{wb_k} = R_{wb_{k-1}} e^{\int_{t_{k-1}}^{t_k} [w]_\times d\sigma} = R_{wb_{k-1}} e^{\phi_\times}. \tag{50}$$

According to (50), when the time interval is very small, the two vector directions can be considered to be coincident, so the cross product part is 0, that is, $\dot{\phi} = w^b_{wb}$, $\phi = \int_{t_{k-1}}^{t_k} w(\sigma) d\sigma$, The increment of the rotation vector

t_{k-1} and t_k can be obtained by the median method, as shown in (36)

$$\phi = \frac{\omega_{k-1} + \omega_k}{2}\left(t_k - t_{k-1}\right), \tag{51}$$

where ω_{k-1} and ω_k are the measured angular velocities of the IMU at t_{k-1} and t_k respectively. By solving the rotation vector, the quaternion update equation, rotation matrix update equation, speed update equation, and position update equation at t_{k-1} and t_k can be obtained, as shown in (52)-(55)

$$q_{wb_k} = q_{wb_{k-1}} \otimes \begin{bmatrix} \cos\frac{\phi}{2} \\ \frac{\phi}{|\phi|}\sin\frac{\phi}{2} \end{bmatrix}, \tag{52}$$

$$R_{wb_k} = R_{wb_{k-1}}\left(I + \frac{\sin\phi}{\phi}\left(\phi^{\wedge}\right) + \frac{1-\cos\phi}{\phi^2}\left(\phi^{\wedge}\right)^2\right), \tag{53}$$

$$v_k = v_{k-1} + \left(\frac{R_{wb_k}a_k + R_{wb_{k-1}}a_{k-1}}{2} - g\right)\left(t_k - t_{k-1}\right), \tag{54}$$

$$\begin{aligned} p_k = {} & p_{k-1} + v_{k-1}\left(t_k - t_{k-1}\right) \\ & + \frac{1}{2}\left(\frac{R_{wb_k}a_k + R_{wb_k-1}a_{k-1}}{2} - g\right)\left(t_k - t_{k-1}\right)^2, \end{aligned} \tag{55}$$

where $q_{wb_{k-1}}$ and q_{wb_k} , $R_{wb_{k-1}}$ and R_{wb_k}, v_{k-1} and v_k, p_{k-1} and p_k are the quaternion, rotation matrix, velocity and position of the robot at t_{k-1} and t_k, respectively. $|\phi|$ denotes the module of the rotation vector. $\phi^{\wedge}$ represents the antisymmetric matrix corresponding to the rotation vector. g represents the gravitational acceleration.

IMU error model construction: In practical application, IMU inevitably has errors, and these errors will be updated and propagated through the IMU solution equation, resulting in the continuous accumulation of errors. By studying the error propagation law of IMU, the error model of IMU is constructed, including the attitude error equation, velocity error equation, position error equation and bias error equation.

In the case of considering the error, the system variables are divided into error value δx, true value x_t and nominal value x. They meet $x_t = x + \delta x$.

Let the attitude update equation of nominal value be

$$q = q \otimes \frac{1}{2}\begin{bmatrix} 0 \\ w - w_b \end{bmatrix}, \tag{56}$$

where w, w_b and q denote the nominal value of the angular velocity of the robot, the bias of the gyroscope, and quaternion, respectively.

The attitude update equation under real value is

$$\dot{q}_t = q_t \otimes \frac{1}{2}\begin{bmatrix} 0 \\ w_t - w_{bt} \end{bmatrix}, \tag{57}$$

where w_t, w_{bt} and q_t are the true values of the angular velocity of robot, the bias of the gyroscope and quaternion, respectively.

The relationship between the true value and the nominal value are

$$q_t = q \otimes \delta q, \tag{58}$$

$$w_t = w + w_n, \tag{59}$$

$$w_{bt} = w_b + \delta w_b, \tag{60}$$

$$\delta q = \begin{bmatrix} \cos\frac{|\delta\theta|}{2} \\ \frac{\delta\theta}{|\delta\theta|}\sin\frac{|\delta\theta|}{2} \end{bmatrix} \approx \begin{bmatrix} 1 \\ \frac{\delta\theta}{2} \end{bmatrix}, \tag{61}$$

where w_n, δw_b, δq and $\delta\theta$ represent the white noise of the gyroscope, the error of the gyroscope's bias, the quaternion error, and the misalignment angle respectively.

Equations (58)-(61) are combined, and finally the attitude error equation is derived as (58)

$$\delta\dot{\theta} = -\left[\omega_t - w_{bt}\right]_{\times}\delta\theta + w_n - \delta w_{bt}. \tag{62}$$

Let the velocity update equation under nominal value be

$$\dot{v} = R\left(a - a_b\right), \tag{63}$$

where a, a_b, v, and R denote the nominal value of acceleration, accelerometer bias, robot velocity, and robot rotation matrix respectively.

The velocity update equation at the true value is

$$\dot{v}_t = R_t\left(a_t - a_{bt}\right), \tag{64}$$

where a_t, a_{bt}, v_t, and R_t are the true value of acceleration, accelerometer bias, carrier velocity, and robot rotation matrix respectively.

The relationship between the actual value and the nominal value of the speed are

$$v_t = v + \delta v, \tag{65}$$

$$R_t = R \cdot e\left([\delta\theta]_{\times}\right) \approx R\left(I + [\delta\theta]_{\times}\right), \tag{66}$$

$$a_t = a + a_n, \tag{67}$$

$$a_{bt} = a_b + \delta a_b, \tag{68}$$

where δv, a_n, and δa_b denote the speed error, the white noise of the accelerometer, and the accelerometer bias error respectively. Equations (66)-(68) are combined, and finally the speed error equation, position error equation and bias error equation are derived as (69)-(71)

$$\delta\dot{v} = -R_t\left[a_t - a_{bt}\right]_{\times}\delta\theta + R_t\left(a_n - \delta a_b\right), \tag{69}$$

$$\delta\dot{p} = \delta v, \tag{70}$$

$$\begin{aligned} \delta\dot{w}_b &= b_{n_w}, \\ \delta\dot{a}_b &= b_{n_a}, \end{aligned} \tag{71}$$

where b_{n_a} and b_{n_w} are the white noise of accelerometer bias and gyroscope bias respectively.

Optimal localization: The optimal dynamic localization uses IMU error model to describe the kinematic state of the system. The ESKF [31] is used as the information fusion method to ensure that the system always maintains the effective linearization state and reduces the amount of calculation. Within the Kalman filtering paradigm, these are the most remarkable assets of the ESKF: (a) The error state degrees of freedom is equal to the actual pose degrees of freedom, which can avoid issues related to over-parametrization. (b) The error-state system always operates close to the origin, and therefore guarantees that the linearization validity holds at all times. (c) The error state is small, so the second-order term can be ignored directly, making the Jacobian calculation simpler and faster.

1) Construct the kinematic equation of the system

The system error state equation is constructed according to the IMU error model. The state equation is as shown in

$$\dot{\delta x} = F_t \delta x + B_t w, \tag{72}$$

where $\delta x = \begin{bmatrix} \delta p & \delta v & \delta\theta & \delta a_b & \delta w_b \end{bmatrix}^T$ and $w = \begin{bmatrix} a_n & a_w & b_{n_a} & b_{n_w} \end{bmatrix}^T$ are the state variables and white noise of the system respectively.

$$F_t = \begin{bmatrix} 0 & I_3 & 0 & 0 & 0 \\ 0 & 0 & -R_t\left[a_t - a_{bt}\right]_\times & -R_t & 0 \\ 0 & 0 & -\left[w_t - w_{bt}\right]_\times & 0 & -I_3 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{73}$$

$$B_t = \begin{bmatrix} 0 & 0 & 0 & 0 \\ R_t & 0 & 0 & 0 \\ 0 & I_3 & 0 & 0 \\ 0 & 0 & I_3 & 0 \\ 0 & 0 & 0 & I_3 \end{bmatrix}. \tag{74}$$

2) Construct the observation equation of the system

According to the pose information obtained by map matching and the position information of GPS in the reliable signal area, the observation equation is constructed as the observation information. The observation equation is as

$$y = G_t \delta x + C_t n, \tag{75}$$

where $y = \begin{bmatrix} \delta\bar{P} & \delta\bar{\theta} \end{bmatrix}^T$ represent the observation of the system including position $\delta\overline{P}$ and attitude $\delta\overline{\theta}$, $\overset{\vee}{P}$ and P denote the location information predicted by the IMU solution and map matching or GPS location observation information when the signal is reliable respectively. $\overset{\vee}{R}_t$ and R_t^T are the predicted rotation matrix calculated by IMU and the rotation matrix obtained from map matching respectively. $n = \begin{bmatrix} n_{\delta\overline{P}} & n_{\delta\overline{\theta}} \end{bmatrix}^T$ is the observation noise.

$$\delta\overline{P} = \overset{\vee}{P} - P, \tag{76}$$

$$\delta\overline{\theta} = \left(R_t^T \overset{\vee}{R}_t - I\right)^{\vee}, \tag{77}$$

$$G_t = \begin{bmatrix} I_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & I_3 & 0 & 0 \end{bmatrix}, \tag{78}$$

$$C_t = \begin{bmatrix} I_3 & 0 \\ 0 & I_3 \end{bmatrix}. \tag{79}$$

3) Construct Kalman filter

The Kalman filter is constructed based on the state equation and observation equation, and the equations are as follows:

$$\delta \overset{\vee}{x}_t = F_{t-1} \delta \hat{x}_{t-1} + B_{t-1} w_t, \tag{80}$$

$$\overset{\vee}{P}_t = F_{t-1} \hat{P}_{t-1} F_{t-1}{}^T + B_{t-1} Q_t B_{t-1}^T, \tag{81}$$

$$K_t = \overset{\vee}{P}_t G_t^T \left(G_t \overset{\vee}{P}_t G_t^T + C_t R_t C_t^T\right)^{-1}, \tag{82}$$

$$\hat{P}_t = (I - K_t G_t) \overset{\vee}{P}_t, \tag{83}$$

$$\delta \hat{x}_t = \overset{\vee}{\delta x}_t + K_t (y_t - G_t \delta \overset{\vee}{x}_t), \tag{84}$$

where Q_t and $\overset{\vee}{P}_t$ are the system process noise and the covariance matrix of the prediction process respectively. K_t denotes the Kalman gain and $\delta \hat{x}_t$ is the estimated error state quantity.

4) Calculate the posterior pose information

The posterior pose information is calculated according to the estimated error state, as shown in

$$\hat{P}_t = \overset{\vee}{P}_t - \delta \hat{P}_t, \tag{85}$$

$$\hat{R}_t = \overset{\vee}{R}_t (I - \left[\delta \overset{\wedge}{\theta}_t\right]_\times). \tag{86}$$

3. EXPERIMENTAL VALIDATION

In this section, some experiments are conducted to illustrate the developed method in terms of accuracy, stability, and real-time performance. Firstly, perform simulation experiments based on the KITTI data set, and then perform actual experiments based on the laboratory's unmanned vehicles and analyze the experimental results.

3.1. Autonomous robot system description

System description: Fig. 7 shows the unmanned vehicle "Beili-Youlong" used in the actual experiment. It is mainly divided into an environment perception system, motion control system, control system, power system and

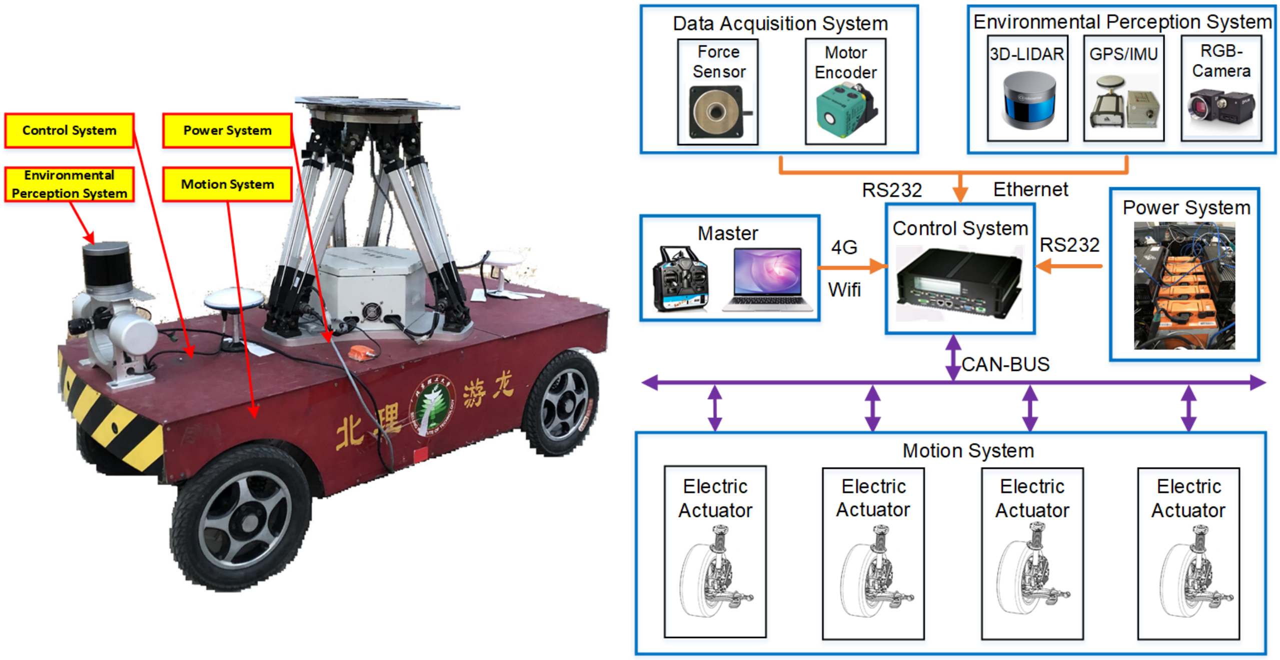


Fig. 7. Experimental platform.

master [36-40]. The environment perception system comprises RS-LIDAR-32 lidar, XW-GI7660 integrated navigation and camera. The motion control system is primarily composed of wheel motors. Besides, the power system is mainly composed of power supplements. The systems communicate through RS-232, Ethernet, or can bus. The RS-lidar-32 adopts a hybrid solid-state lidar method and integrates 32 laser transceiver components. The measurement distance is up to 200 meters, the measurement accuracy is within $+/-5$ cm, the output points are up to 640,000 points/sec, the horizontal angle is $360°$, and the vertical measurement is angle $-25° \sim 15°$. The XW-GI7660 integrated navigation has a 3-axis fiber optic gyroscope, 3-axis quartz accelerometer and dual GNSS receivers. When the signal is reliable, the heading accuracy is $0.04°$, the attitude accuracy is $0.02°$, and the position accuracy is 1 cm+1 ppm.

System setting: To prove the effectiveness of our method, we perform the KITTI dataset to evaluate our method quantitatively and qualitatively. The KITTI data set is the most extensive computer vision evaluation data set in autonomous driving, including actual image data, lidar data, 2D and 3D object annotation data collected from urban, rural and highway scenes. The data acquisition platform of the KITTI dataset has two grayscale cameras, two-color cameras, a Velodyne 64-line 3D lidar and a GPS navigation system. The experimental computer system is configured with Intel i5-4210H CPU (2.90 GHz) and 8 GB memory, using the ROS [41] under Ubuntu Linux as the software platform.

3.2. Evaluation performance on the KITTI dataset

We used KITTI00 dataset. The data set is collected from urban communities. The road environment is asphalt road, including a continuous bumpy road and light change scene caused by the artificial well cover and road fluctuation. There are many intersections, and the length of the driving track is about 3.7 km. The GPS trajectory in the data set is taken as the ground truth, and the experimental results are shown in Fig. 8 and Table 1.

We calculate the translation error and rotation error at different distances and speeds respectively [17]. The error metrics are defined as

$$E_{rot}(f) = \frac{1}{|f|} \sum_{(i,j)\in f} \angle \left[\left(\hat{p}_j \odot \hat{p}_i \right) \odot \left(p_j \odot p_i \right) \right], \quad (87)$$

$$E_{trans}(f) = \frac{1}{|f|} \sum_{(i,j)\in f} \left\| \left(\hat{p}_j \odot \hat{p}_i \right) \odot \left(p_j \odot p_i \right) \right\|_2, \quad (88)$$

where f is a set of frames (i, j), $\hat{p} \in SE(3)$ and $p \in SE(3)$ are estimated and true poses respectively. $\odot$ denotes Pose

Table 1. Accuracy and runtime comparison results between the proposed method and other methods on KITTI00.

Method	RMSE (m)	Runtime (s)
Proposed method	0.12	0.02
LIO-SAM [29]	0.96	0.04
LINS [30]	0.41	0.02
Point-Localization [42]	0.32	0.05

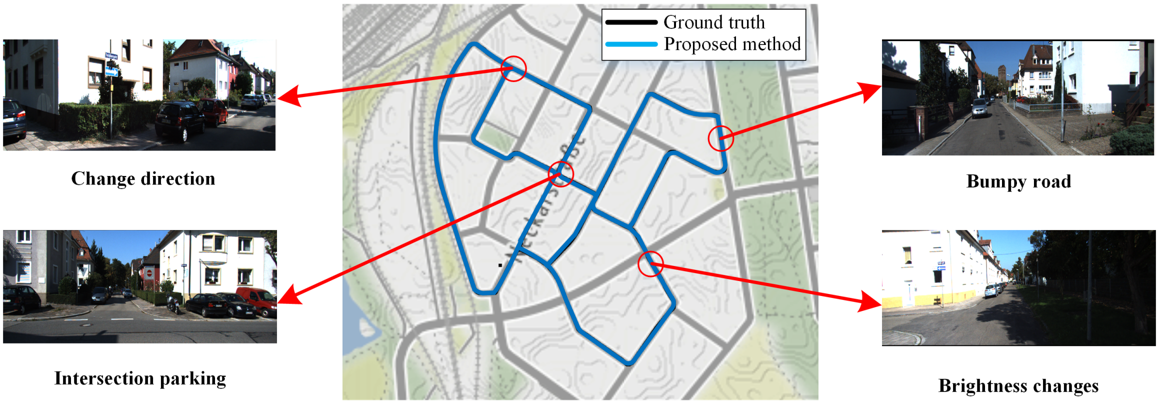
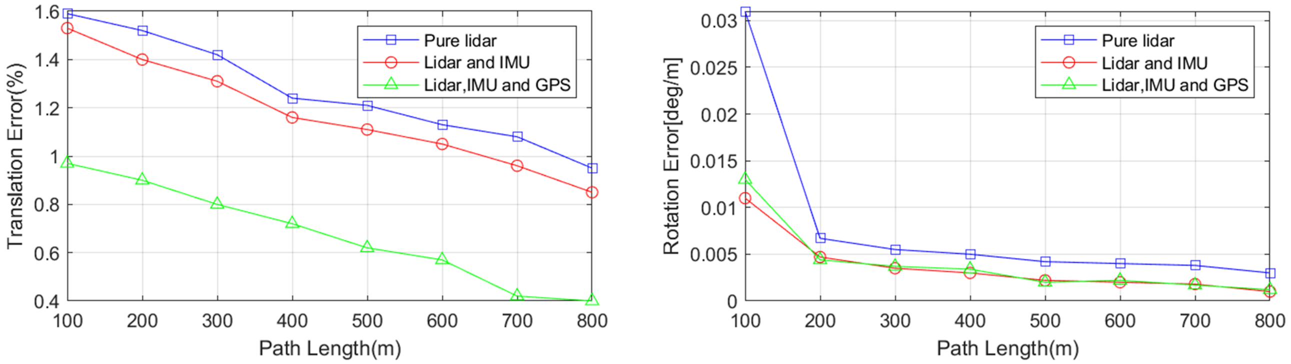


Fig. 8. Experimental results on KITTI00.



(a)

(b)

Fig. 9. Ablation experiment. (a) Translation errors at different distances. (b) Rotation errors at different distances.

transformation between two frames, and ∠ represents the rotation angle.

Table 1 shows the accuracy and speed results of our method and other methods. It shows that our method can achieve centimeter-level localization accuracy. Fig. 9 shows the translation error and rotation error of localization at different distances. The rotation error is between 0.001 deg/m and 0.012 deg/m, and the translation error is between 0.35% and 0.95%. We also conducted ablation experiments and generated the results of lidar, lidar-IMU, and lidar-IMU-GPS. After adding IMU, the localization accuracy changes little, but the standard deviation decreases, indicating that the volatility of localization results decreases and the positioning information is more stable. After introducing the information of GPS in the reliable information area, the localization accuracy is greatly improved.

3.3. Evaluation performance on autonomous mobile robot

Fig. 10 shows the experimental results of evaluating the localization algorithm on the actual test scene data. The experimental environment is the school of automation at Beijing University of technology. The road environment is asphalt road, including a continuous bumpy road, brightness changes scene, empty scene and shaded by trees. The localization information of GPS when the signal is reliable is taken as the ground truth. The absolute error results of the tested localization are shown in Table 2. The experimental results show that the average localization error is about 0.05 m. In LIO-SAM [29], a method based on factor graph optimization is used to fuse multi-source information, that is, all information before the current moment is considered for fusion optimization, resulting in poor real-time performance. In the real vehicle experiment, it resulted in a large deviation in some road sections. GPS has high positioning accuracy when the signal is valid, but in LINS [30], the error state Kalman filter only integrates the information of lidar odometer and IMU, and does not add GPS information for optimization, so the performance is poor. In Point-Localization [42], the odometer made by

Table 2. Actual experiment localization error result.

Method	Error (m)
Proposed method	0.05
LIO-SAM [29]	0.62
LINS [30]	0.38
Point-Localization [42]	0.22

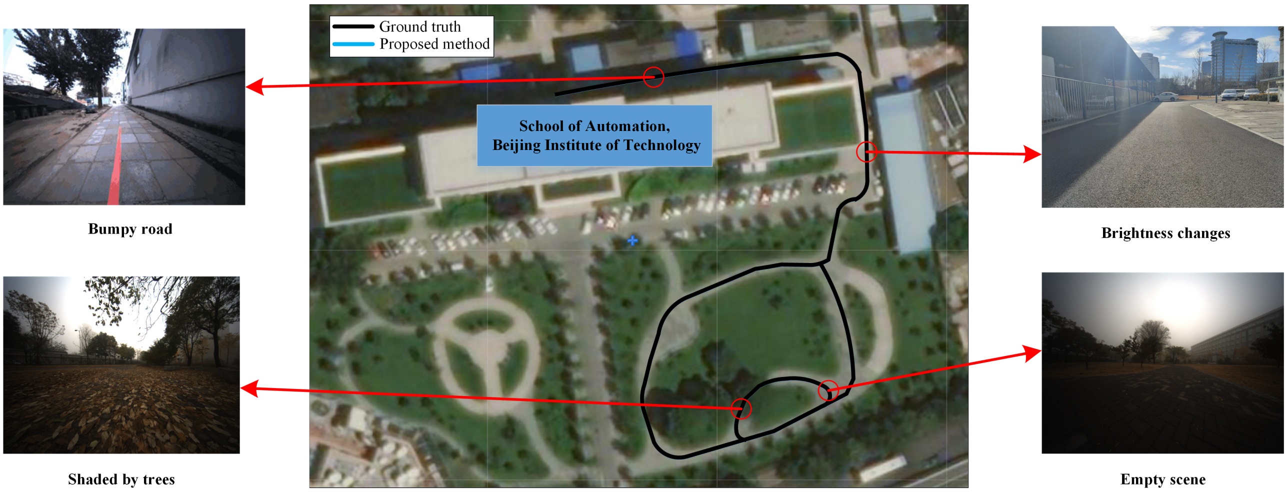


Fig. 10. Experimental results on actual test scene data.

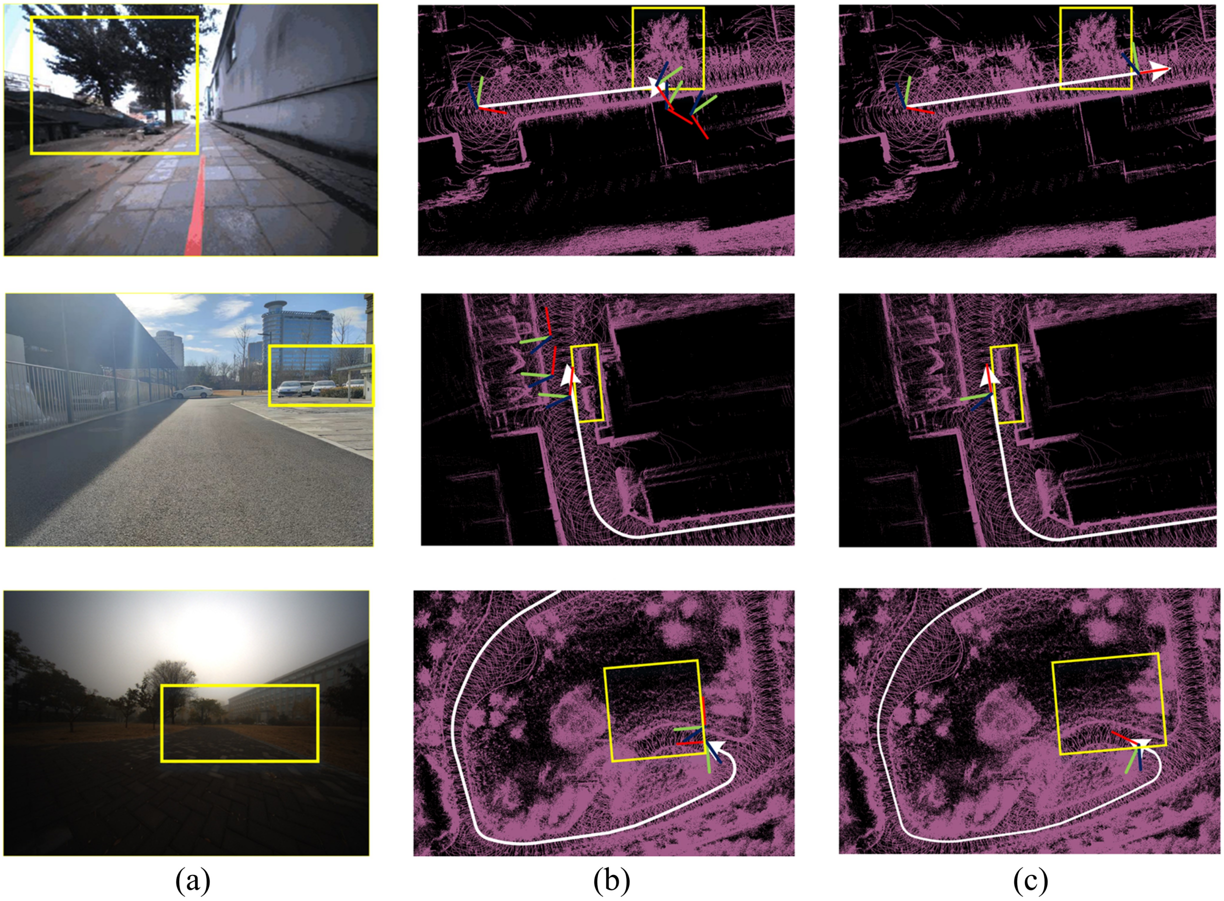


Fig. 11. Comparative test results under three scenarios. (a) GPS signal failure. (b) Lidar field of view is blocked. (c) Sparse scene features.

Hall sensor is fused with the point cloud map matching and positioning, and the GPS information is also not added for optimization, which leads to a decrease of the overall positioning accuracy.

Fig. 11 shows the localization results of the method proposed in this paper under three complex scenarios: GPS signal failure, lidar field of view is blocked and sparse scene features. As shown in Figs. 11(a) and 11(b), the scene is a tree that causes the GPS signal to be blocked and invalid, but the unmanned vehicle can still locate stably Fig. 11(c). As shown in Figs. 11(d) and 11(e), two vehicles block part of the field of view of the lidar, but the unmanned vehicle can rely on IMU solution for accurate positioning in a short time. As shown in Figs. 11(g) and 11(h), map matching can not accurately locate in sparse scenes, but reliable GPS positioning information in open areas can enable unmanned vehicles to locate stably. Experiments show that the proposed method can achieve stable positioning in these three complex scenes.

4. CONCLUSION

This paper proposes an optimal dynamic localization method for autonomous mobile robots via integration sensors. Firstly, the global point cloud map with low drift and small cumulative error established by 3D-Lidar is used as a priori information. Next, the error model is established based on the measurement information of IMU, and then the system state equation is constructed. Furthermore, the observation equation is constructed based on the position and attitude information obtained by global map matching and the position information of GPS in the signal reliable area. Finally, the error state Kalman filter optimizes the fusion localization. The experimental results show that the average locating error of this method is 0.08 m, the fastest single locating time is 0.01 s, and it can locate stably in complex scenes.

In the future, the cameras based on deep learning networks will be considered to increase the diversity of map construction information and improve the perception and planning of autonomous robots. Furthermore, in terms of optimal localization, more efficient multi-sensor fusion strategies will design to enhance the accuracy and robustness.

CONFLICT OF INTEREST

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

REFERENCES

[1] K.-S. Hong and P.-T. Pham, "Control of axially moving systems: A review," *International Journal of Control, Automation, and Systems*, vol. 17, no. 12, pp. 2983-3008, 2019.

[2] S. Muhammad and G.-W. Kim, "Simultaneous localization and mapping in the epoch of semantics: A survey," *International Journal of Control, Automation, and Systems*, vol. 17, no. 3, pp. 729-742, 2019.

[3] J. Li, J. Wang, S. Wang, W. Qi, L. Zhang, Y. Hu, and H. Su, "Neural approximation-based model predictive tracking control of non-holonomic wheel-legged robots," *International Journal of Control, Automation, and Systems*, vol. 19, no. 1, pp. 372-381, 2021.

[4] J. Li, J. Wang, S. Wang, and C. Yang, "Human-robot skill transmission for mobile robot via learning by demonstration," *Neural Computing and Applications*, pp. 1-11, 2021. DOI: 10.1007/s00521-021-06449-x

[5] Y. Dai, J. Wang, J. Li, and J. Li, "MDRNet: A lightweight network for real-time semantic segmentation in street scenes," *Assembly Automation*, vol. 41, no. 6, pp. 725-733, 2021.

[6] J. Li, J. Wang, H. Peng, L. Zhang, Y. Hu, and H. Su, "Neural fuzzy approximation enhanced autonomous tracking control of the wheel-legged robot under uncertain physical interaction," *Neurocomputing*, vol. 410, pp. 342-353, 2020.

[7] J. Li, Y. Dai, J. Wang, X. Su, and R. Ma, "Towards broad learning networks on unmanned mobile robot for semantic segmentation," *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, pp. 9228-9234, 2022.

[8] N. Kawabata, Y. Kuwabara, and T. Kawasaki, "Self-localization of autonomous car using autoware," *IEICE Technical Report*, vol. 120, no. 389, pp. 103-108, 2021.

[9] J. Li, H. Qin, J. Wang, and J. Li, "OpenStreetMap-based autonomous navigation for the four wheel-legged robot via 3d-lidar and CCD camera," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 3, pp. 2708-2717, 2022.

[10] S. Kuutti, S. Fallah, K. Katsaros, M. Dianati, F. Mccullough, and A. Mouzakitis, "A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 829-846, 2018.

[11] J. Li, X. Zhang, J. Li, Y. Liu, and J. Wang, "Building and optimization of 3D semantic map based on Lidar and camera fusion," *Neurocomputing*, vol. 409, pp. 394-407, 2020.

[12] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE Access*, vol. 8, pp. 58443-58469, 2020.

[13] Y. Dai, J. Li, J. Wang, and J. Li, "Towards extreme learning machine framework for lane detection on unmanned mobile robot," *Assembly Automation*, vol. 42, no. 3, pp. 361-371, 2022.

[14] E. Stenborg, *Long-term Localization for Self-driving Cars*, Ph.D. dissertation, Chalmers Univeristy of Technology, 2020.

[15] J. Liu and G. Guo, "Vehicle localization during gps outages with extended Kalman filter and deep learning," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1-10, 2021.

[16] J. Liu and G. Guo, "Vehicle localization during gps outages with extended kalman filter and deep learning," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1-10, 2021.

[17] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORS-SLAM: A versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147-1163, 2015.

[18] J. Zhang and S. Singh, "Low-drift and real-time lidar odometry and mapping," *Autonomous Robots*, vol. 41, no. 2, pp. 401-416, 2017.

[19] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, pp. 3354-3361, 2012.

[20] A. Ranganathan, D. Ilstrup, and T. Wu, "Light-weight localization for vehicles using road markings," *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, pp. 921-927, 2013.

[21] X. Li, S. Du, G. Li, and H. Li, "Integrate point-cloud segmentation with 3D lidar scan-matching for mobile robot localization and mapping," *Sensors*, vol. 20, no. 1, p. 237, 2020.

[22] J. K. Suhr, J. Jang, D. Min, and H. G. Jung, "Sensor fusion-based low-cost vehicle localization system for complex urban environments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1078-1086, 2016.

[23] X. Lin, F. Wang, B. Yang, and W. Zhang, "Autonomous vehicle localization with prior visual point cloud map constraints in gnss-challenged environments," *Remote Sensing*, vol. 13, no. 3, p. 506, 2021.

[24] R. W. Wolcott and R. M. Eustice, "Fast lidar localization using multiresolution gaussian mixture maps," *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, pp. 2814-2821, 2015.

[25] N. Akai, L. Y. Morales, E. Takeuchi, Y. Yoshihara, and Y. Ninomiya, "Robust localization using 3D NDT scan matching with experimentally determined uncertainty and road marker matching," *Proc. of IEEE Intelligent Vehicles Symposium (IV)*, IEEE, pp. 1356-1363, 2017.

[26] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35-45, 1960. 1960.

[27] M. Mehdikhani, *Integration of a Low-cost Gyro in the Localization of an Industrial Mobile Robot via an Rrror-state Extended Kalman Filter*, Master's thesis, ING, 2021.

[28] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, pp. 4758-4765, 2018.

[29] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "LIO-SAM: Tightly-coupled lidar inertial odometry via smoothing and mapping," *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, pp. 5135-5142, 2020.

[30] C. Qin, H. Ye, C. E. Pranata, J. Han, S. Zhang, and M. Liu, "LINS: A lidar-inertial state estimator for robust and efficient navigation," *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, pp. 8899-8906, 2020.

[31] J. Sola, "Quaternion kinematics for the error-state kalman filter," arXiv preprint arXiv:1711.02508, 2017.

[32] G. Kim and A. Kim, "Scan context: Egocentric spatial descriptor for place recognition within 3D point cloud map," *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, pp. 4802-4809, 2018.

[33] B. Sheng, S. Wenzhong, F. Wenzheng, C. Pengxin, N. Mingyan, and X. Haodong, "A tight coupling mapping method to integrate the ESKF, g2o, and point cloud alignment," *The Journal of Supercomputing*, vol. 78, pp. 1903-1922, 2022.

[34] H. Lim, S. Hwang, S. Shin, and H. Myung, "Normal distributions transform is enough: Real-time 3D scan matching for pose correction of mobile robot under large odometry uncertainties," *Proc. of 20th International Conference on Control, Automation and Systems (ICCAS)*, IEEE, pp. 1155-1161, 2020.

[35] S. Srinara, C.-M. Lee, S. Tsai, G.-J. Tsai, and K.-W. Chiang, "Performance analysis of 3D BDT scan matching for autonomous vehicles using INS/GNSS/3D LiDAR-SLAM integration scheme," *Proc. of IEEE International Symposium on Inertial Sensors and Systems (INERTIAL)*, pp. 1-4, 2021.

[36] J. Li, R. Li, J. Li, J. Wang, Q. Wu, and X. Liu, "Dual-view 3D object recognition and detection via Lidar point cloud and camera image," *Robotics and Autonomous Systems*, vol. 150, 103999, 2022.

[37] S. Wang, Z. Chen, J. Li, J. Wang, J. Li, and J. Zhao, "Flexible motion framework of the six wheel-legged robot: Experimental results," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 4, pp. 2246-2257, 2022.

[38] J. Li, J. Wang, H. Peng, Y. Hu, and H. Su, "Fuzzy-torque approximation-enhanced sliding mode control for lateral stability of mobile robot," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 4, pp. 2491-2500, 2022.

[39] Z. Chen, J. Li, S. Wang, J. Wang, and L. Ma, "Flexible gait transition for six wheel-legged robot with unstructured terrains," *Robotics and Autonomous Systems*, vol. 150, 103989, 2022.

[40] J. Li, Y. Dai, X. Su, and W. Wu, "Efficient dual-branch bottleneck networks of semantic segmentation based on CCD camera," *Remote Sensing*, vol. 14, no. 16, p. 3925, 2022.

[41] K. Zheng, "ROS navigation tuning guide," *Robot Operating System (ROS)*, pp. 197-226, Springer, 2021.

[42] Y. Zhu, B. Xue, L. Zheng, H. Huang, M. Liu, and R. Fan, "Real-time, environmentally-robust 3d lidar localization," *Proc. of IEEE International Conference on Imaging Systems and Techniques (IST)*, pp. 1-6, 2019.

Jing Li was born in 1982. She received her M.S. degree of engineering from Shandong University of Technology in 2007, and a Ph.D. degree in control science and engineering from Beijing Institute of Technology in 2011. She is now an associate professor of School of Automation, State Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing Institute of Technology. Her research interests include image detection technology, and object detection and tracking.

Keyan Guo was born in 1997. He received his B.S. degree in automation from Ocean University of China, Qingdao, China, in 2019. He is currently pursuing an M.S. degree as a member of State Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing Institute of Technology, China. His current research interests include simultaneous localization and mapping of mobile robot.

Junzheng Wang received his Ph.D. degree in control science and engineering from the Beijing Institute of Technology, Beijing, China, in 1994. He is the Deputy Director with the State Key Laboratory of Intelligent Control and Decision of Complex Systems, the Director of the Key Laboratory of Servo Motion System Drive and Control, and the Dean of the Graduate School of Beijing Institute of Technology, where he is a Professor and a Ph.D. Supervisor. His current research interests include motion control, electric hydraulic servo system, and dynamic target detection and tracking based on image technology. He received the Second Award from the National Scientific and Technological Progress (No.1) of China.

Jiehao Li received his M.Sc. degree in control engineering at South China University of Technology, Guangzhou, China, in 2017. He received a Ph.D. degree at the State Key Laboratory of Intelligent Control and Decision of Complex Systems, School of Automation, Beijing Institute of Technology, Beijing, China, in 2022. He is now an Associate Professor at College of Engineering, South China Agricultural University, Guangzhou, China. He is also a Visiting Fellow of the Medical and Robotic Surgery Group (NEARLab) in Politecnico di Milano, Milano, Italy. His research interests mainly include mobile robotics, motion control, robot vision, and image processing. Prof. Li is the Academic Committee Member of Youth Working Committee of CAAI and CICC. He has been awarded the Best Conference Paper Finalist of IEEE ICARM2020, the Outstanding Reviewer of CAC2021, and the Outstanding Session Chair of WRC SARA2022. He is the Conference Session Chair of IEEE ICUS2022, ICIRA2022 and YAC2022. He has served as the Guest Editor of IET Control Theory & Applications, Frontiers in Neurorobotics, and the Associate Editor of Journal of Control Science and Engineering.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.