IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 6, 2020, accepted January 16, 2020, date of publication January 21, 2020, date of current version January 31, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2968353

LCPF: A Particle Filter Lidar SLAM System With Loop Detection and Correction

FUYU NIE, WEIMIN ZHANG, ZHUO YAO, YONGLIANG SHI, FANGXING LI, (Member, IEEE), AND QIANG HUANG, (Fellow, IEEE)

School of Mechatronical Engineering, Beijing Institute of Technology, Beijing 100811, China
Key Laboratory of Biomimetic Robots and Systems, Ministry of Education, Beijing Institute of Technology, Beijing 100081, China
Beijing Advanced Innovation Center for Intelligent Robots and Systems, Beijing 100081, China

Corresponding author: Weimin Zhang (zhwm@bit.edu.cn)

This work was supported by the National Key Research and Development Program of China under Grant 2017YFB1302200.

ABSTRACT A globally consistent map is the basis of indoor robot localization and navigation. However, map built by Rao-Blackwellized Particle Filter (RBPF) doesn't have high global consistency which is not suitable for long-term application in large scene. To address the problem, we present an improved RBPF Lidar SLAM system with loop detection and correction named LCPF. The efficiency and accuracy of loop detection depend on the segmentation of submaps. Instead of dividing the submap at fixed number of laser scan like existing method, Dynamic Submap Segmentation is proposed in LCPF. This segmentation algorithm decreases the error inside the submap by splitting the submap where there is high scan match error and later rectifies the error by an improved pose graph optimization between submaps. In order to segment the submap at appropriate point, when to create a new submap is determined by both the accumulation of scan match error and the particle distribution. Furthermore, LCPF uses branch and bound algorithm as basic detector for loop detection and multiple criteria to judge the reliability of a loop. In the criteria, a novel parameter called usable ratio was proposed to measure the useful information that a laser scan containing. Finally, comparisons to existing 2D-Lidar mapping algorithm are performed with a series of open dataset simulations and real robot experiments to demonstrate the effectiveness of LCPF.

INDEX TERMS Simultaneous localization and mapping, mobile robots, indoor navigation, particle filter, loop detection, dynamic submap segementation.

I. INTRODUCTION

Indoor service robots localization and navigation have recently attracted much research interest. Most indoor localization is realized by Simultaneous Localization And Mapping (SLAM). In an unknown environment, the robot localize itself and build a map according to the information (2D or 3D lidar ranging readings [1] [2], infrared ranging readings [3], video data streams [4] [5], etc.) about the environment. An accurate map is the basis of localization and navigation. Generally speaking, mapping according to pose and observation is easy to realize. Therefore, researchers mainly focus on localization, videlicet, pose estimation.

The accumulative error of the pose obtained from the encoder odometer will increase in long-term running situation. Several methods were proposed to cope with this problem. Firstly, EKF-SLAM [6]–[8] represents the state uncertainty by an approximate mean and variance. The system noise is presumed Gaussian and the non-linear models are linearized by first-order Jacobian approximation. However, linear Gaussian system is a strong assumption in indoor robot localization environment. When the local linearity is not ideal, the irrational linearization will cause error in both motion model and observation model. Considering these states are estimated by linearization may not accurately match the real states, and the noise of real system mostly doesn't follow the Gaussian distribution strictly. Then, Rao-Blackwellizd particle filter are introduced in SLAM, such as [9], [9], [10]. In RBPF SLAM, each particle carries an individual map of the whole environment that has already been explored. Therefore, in RBPF, the previous built map carried by a particle cannot be changed in subsequent explorations. In this case, the accumulative error can't be rectified in RBPF. Cartographer in [11] introduced loop closure in 2D

The associate editor coordinating the review of this manuscript and approving it for publication was Heng Wang.

Lidar SLAM and used branch and bound algorithm to compute scan-to-map matches as constraints. This optimization based algorithm provides space for further improvement.

In this manuscript, we focus on fusing Loop Closure and RBPF SLAM to combine their advantages. To guarantee the efficiency of loop detection, we need to divide the map at appropriate breakpoint according to scan match error. Fortunately, particle filter can provide robust scan match result and the estimation error can be estimated according to the standard variance of pose distribution and scan match error. In this case, dynamic submap segmentation is proposed. This segmentation reduces the error inside submap and allocates the error between submaps, which minimizes the influence of ignoring the internal errors of submaps. Then, the residual error between submaps can be eliminated by global pose graph optimization. In order to get a better local constraint between adjacent submaps, we propose lag local constraint rectification that uses branch and bound algorithm to avoid falling into local maximum. To guarantee a robust loop detection, we propose a concept called *usable ratio* to measure the reliability of a laser scan for global loop detection. The usable ratio increases the reliability of loop detection and reduce the cost of loop detection by avoiding unreliable laser scan from joining global scan match. Our contributions can be summarized as follows:

1) Loop Closure and Particle Filter are fused in our algorithm, improving the global consistency of existing RBPF SLAM system.
2) This paper proposed an algorithm to divide the submap appropriately, decreasing the error inside the submaps.
3) Multiple criteria and branch and bound algorithm are applied to improve the reliability of loop detection.
4) Experiment are carried out to verify the performance of LCPF.

The paper is organized as follows. Section II describes related work about SLAM and loop closure. Section III presents the framework of LCPF, including RBPF lidar SLAM system with loop detection and correction. Section IV focuses on simulation, experiment and result analysis. Finally, section V is the conclusion and our future work.

II. RELATED WORK

Simultaneous locallization and mapping (SLAM) is a key issue in robotics [12], which has attracted great research interests in recent decades. The most classical algorithm is Extended Kalman Filter (EKF) based SLAM [6]–[8]. There are various of EKF SLAM that proposed for diffirent task and enviornment, such as [13]–[15]. EKF SLAM assume the sensor noise satisfies the Gaussian distribution and the nonlinearity of system is small locally [16]. Most EKF SLAM is a feature based system, and the map consists of features [17]. Therefore, as the landmark points continue to increase, the information matrix of the system increases in a quadratic way [18], taking up more and more computational source and storage space. In this way, EKF SLAM is not suitable for large scale environments as discussed in this paper.

To overcome the problem of nonlinearity and large scale, particle filter based SLAM are proposed, repsresented by RBPF SLAM [19]. Particle filter is a nonparametric estimation algorithm [9], [20], [21], which means it can be used to estimate any distribution. In these algorithm, each particle carries an individual map of the environment and an individual trajectory. The key idea is to estimate a posterior $p(x_{1:}|z_{1:t}, u_{0:t})$ about potential trajectroys $x_{1:t}$ of the robot given its observation $z_{1:t}$ and its odometry measurements $u_{0:t}$. This posterior trajectroy is used to compute a posterior maps. By introducing the improved proposal distribution and the adaptive resampling, [19] and [9] reduce the number of particles and resampling, and the resampling is performed only when needed. However, RBPF SLAM has no global information of particle. In long-term running, although it has good local continuity, the global consistency will decline over time and the map will be gradually misplaced.

In this case, graph SLAM [22]–[24], represneted by Cartographer in [11] was proposed to solve the problem of real-time loop detection and loop closure. This algorithm inserts fixed number of laser scan (range finder reading) into a submap at the best estimated position, which assumed to be sufficiently accurate for short periods of time. When a submap is finished, it takes part in the global loop detection. The detection are implemented by a branch and bound scan match method. All finished submaps and laser scans are considered for loop detection and closure. If a loop is founded around the currently pose, it is added to the pose graph as an edge for optimization. Cartographer has been proved to be an effective method for loop detection and closure. However, to enhance the accuracy and efficiency of loop detection is still one of the research focuses.

III. PROPOSED METHOD

As is mentioned in Section I, LCPF is a particle filter lidar SLAM system with loop detection and correction. The method in this paper consists of front-end and back-end. The front-end of LCPF mainly focus on map-building and dividing the global map into submaps with our dynamic submap segmentation algorithm, which minimize the error in a submap and leave the error between submaps for later optimization. When building a map, the front-end also provide a constraint between adjacent submap. And the back-end consists of loop detection and correction thread and lag local constraint rectify thread. Lag local constraint rectify means using branch and bound scan matcher to rectify local constraint generated by local scan matcher. The framework of the RBPF lidar SLAM system with loop closure is shown in Fig. 1. The input of LCPF are 2D laser-scan and odometry information, and the output is the optimal global map consisting of submaps. The left area represents front-end, including RBPF and submap builder, which is realized by dynamic submap segmentation (DMS). The right area represents back-end

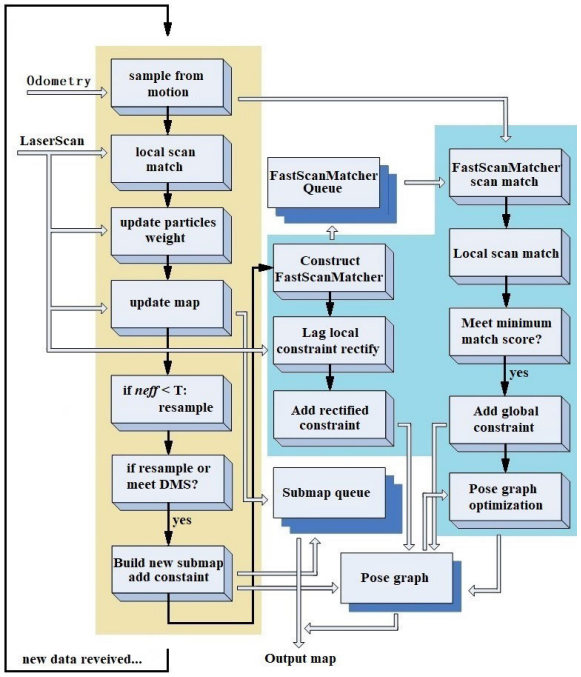


FIGURE 1. Framework of LCPF proposed in this manuscript. The left area represents front-end, and the right area represents back-end.

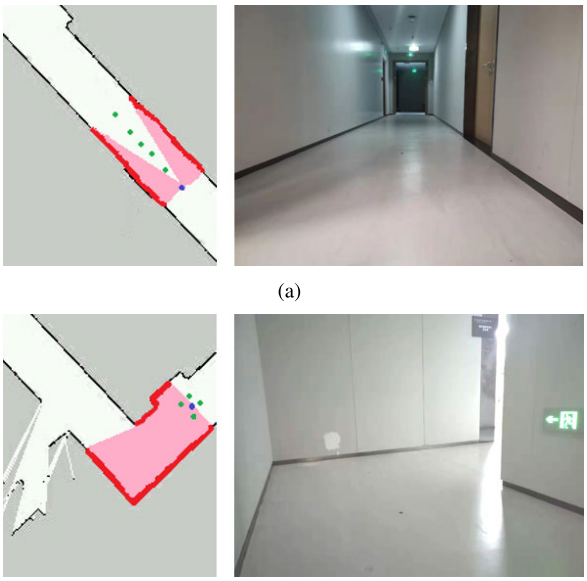


FIGURE 2. Illustration of the criterion for dynamic submap. (Occupancy grid maps and the corresponding actual scenes).

of LCPF, including banch and bound scan matcher and global pose graph optimizer.

A. DYNAMIC SUBMAP SEGMENTATION

Considering the computational cost and space occupancy of loop detection, it is unrealistic to search for possible loops on the full map. So loop detection needs to be performed on smaller units. And the whole map is not suitable for pose graph optimization. Therefore, it is necessary to divide the whole map into some smaller map, that is, submap, as the basic unit for loop detection and loop closure.

Cartographer in [11] assumes that the error within the submap is small enough to be ignored. Our method also uses the same assumption. However, this method introduce a new way of submaps segmentation while Cartographer insert fixed number of laser scan into submap. The new way take matching error estimation into consideration during the buiding of submap. Cause only error between submap will be minimized, it aims to decrease error inside submap and allocate more scan match error between submaps. However, the error of the local scan match is irregular and determined by local environment. If some laser scans inside submap are inserted at a wrong position, the loop closure cannot rectify these errors. Considering the computational source and the storage is limited, it is difficult to optimize all pose of laser scans that are inserted into the submap. However, in our algorithm, error inside the submap can be decreased by allocating more scan match error between submaps, which is the essence of dynamic submap segmentation. To achieve this, we use matching error $e(x, y, \theta)$ as the criterion for dynamic submap segmentation. And the matching error of laser scan is evaluated by the score of the best partilce and the distribution of all particle:

$$e(x, y, \theta) = s(x) * \xi_x + s(y) * \xi_y + s(\theta) * \xi_\theta \tag{1}$$

When inserting a laser scan, we uses (1) to calculate the estimation of matching error. ξ_x, ξ_y and ξ_θ represents the weight of translation and rotation error estimation, where x, y and θ represent the pose of robot. High weight means low tolerance in a variable, and this tolerance is determined by the score of the best particle. However, the score of the best particle can't always fully reflect the reliability of current pose. In Fig. 2, all the particles are with high score of matching, which means the laser scan match well with map at the pose of all particles. Points at the intersection of laser beams indicate the particles with highest score and the other points are suboptimal particles with scores close behind. In featureless environment like (a), particles are dispersed along the corridor, which can cause larger error than (b). As is shown in Fig. 2(a), when robot is in a long corridor like in (a), the pose of the best particle gets high matching score. But the map of corridor is long and straight, the likelihood laser scans of suboptimal points are not much different from the scan of best particle. Therefore, all the particles along the direction of corridor get high scores, and the pose uncertainty along the direction of corridor is high even though the best particle's score is high enough. So we need to take the distribution of particles into consideration to avoid the uncertainty under circumstance like this. And the matching error on each state variable depends on the dispersion of particles projected on this direction:

$$s(x) = \sqrt{[\sum_{i=1}^{n} (x_i - \bar{x})^2]/(n-1)}$$

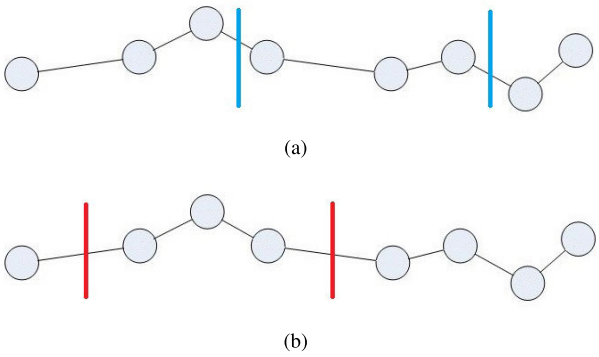


FIGURE 3. Example of fixed segmentation (a), and dynamic submap segmentation (b).

$$s(y) = \sqrt{[\sum_{i=1}^{n}(y_i - \bar{y})^2]/(n-1)}$$
$$s(\theta) = \sqrt{[\sum_{i=1}^{n}(\theta_i - \bar{\theta})^2]/(n-1)} \quad (2)$$

In situation like Fig. 2(a), the dispersion of the particles is high which can lead to large error inside the submap, so the matching error ξ_x, ξ_y increase when particles are more dispersed. In situation like Fig. 2(b), the robot meets a corner showing distinct feature and the uncertainty of pose is low, the front-end continues to build the submap. Besides, when the accumulation of scan match error is slow and the sum of error exceeds a threshold, we also add a segmentation point there to avoid the influence of slowly growing error. As in Fig. 3, points represents laser scan and the pose that insert them, and their connection represents the scan match error between poses (longer connection means larger error estimation). The vertical lines show where to split submaps. Fig. 3(a) shows submap containing fixed number of laser scans, and Fig. 3(b) shows dynamic submap segmentation. When the matching error at certain pose exceeds the dynamic segmentation threshold, a new submap are created and the error was left between submaps rather than inside them. In Fig. 3(a), the fixed submap segmentation divides the map into submaps with fixed number of laser scans, regardless of whether the breakpoint need optimization or not. The dynamic submap segmentation in Fig. 3(b) divides the map at breakpoints where there is high matching error between submaps to be generated. Generally speaking, DMS allocates the optimization computing resources to nodes with high uncertainty and assemble the area with low uncertainty into a submap without further optimization. In this way, DMS decrease the error in mapping with higher computing efficiency.

B. FRONT-END OF LCPF

The front-end is responsible for providing the map carried by the best particle, as well as the initial value of the pose

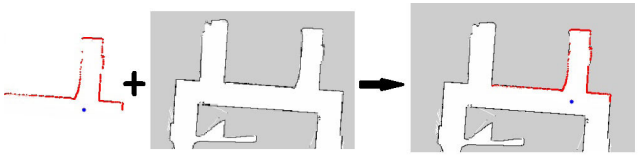


FIGURE 4. Process of scan match. Red points represents laser scan and the blue point represents the location of lidar.

graph optimization, and the relative pose between adjacent submaps will be added as constraints to the pose graph. The pose constraint between adjacent submaps is also called local constraint, and the constraint between noncontiguous submaps is also called global constraint. Global constraints are added after loop detection, and pose graph are optimized after adding global constraints. When create new submap in front-end, our method choose the pose of current best particle as the origin. And the submap will follow the particle till the submap is finished. The best particle can change before current submap finished, but in order to reduce internal error, the particle that the submap followed will be unchanged. And if the RBPF particle filter resampling, current submap will be finished immediately and new submap will be created. The concrete details of front-end model is as follows.

1) UPDATING CRITERIA

The front-end is based on an improved RBPF SLAM. It maintains a particle swarm, in which each particle represents a possible trajectory of the robot and the corresponding map of the trajectory. The frequency of laser scan is 15Hz, which contains redundant information. After receiving one frame of laser scan and its corresponding odometry data, to expurgate the information that have high comparability, we control the update frequency of submap by

$$\sqrt{\Delta x^2 + \Delta y^2} > T_{\text{tran}} \quad (3)$$
$$\Delta\theta > T_{\text{rot}} \quad (4)$$

If the translation Δx, Δy or the rotation $\Delta\theta$ exceeding the threshold T_{tran} and T_{rot}, the distant is considered to be sufficiently large. This frame of laser scan will be added to the submap of each particle, and their trajectory will be updated too.

2) POSE UPDATE

When the motion of robot meets the updating condition, we use particle filter to propagate the pose and update it according to observation. We denote the robot pose at different times as:

$$x_{0:t}^{[i]} = x_0^{[i]}, x_1^{[i]}, \ldots, x_t^{[i]} \quad (5)$$

In (5), $x_{0:t}^{[m]}$ is the pose of m-th particle from time 0 to t. Sequence like this makes up a trajectory, and a map is obtained by superimposing lidar observation on this trajectory. To estimate the trajectory, we need a particle swarm that containing many possible trajectories:

$$X_{0:t} = x_{0:t}^{[1]}, x_{0:t}^{[2]}, \ldots, x_{0:t}^{[n]} \quad (6)$$

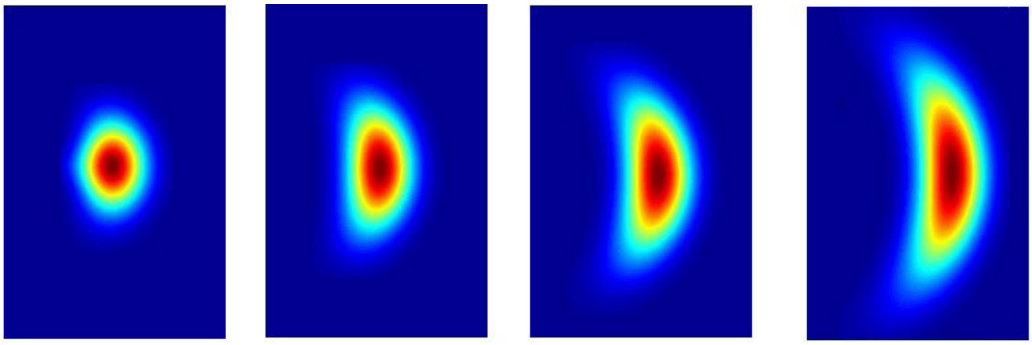


FIGURE 5. Uncertainty distribution of two-dimensional lidar observation in beam projection model.

$X_{0:t}$ in (6) is a particle swarm containing n particles. Each particle maintains a trajectory and a map. To estimate the trajectory and build an accurate map with odometry information $u_{1:t}$ and lidar observation $z_{1:t}$, we need to compute the posterior distribution of the trajectory:

$$\begin{aligned} & p\left(x_{0:t}|z_{1:t}, u_{1:t}\right) \\ & = \eta p\left(z_t|x_{0:t}, z_{1:t-1}, u_{1:t}\right) p\left(x_{0:t}|z_{1:t-1}, u_{1:t}\right) \\ & = \eta p\left(z_t|x_t\right) p\left(x_{0:t}|z_{1:t-1}, u_{1:t}\right) \\ & = \eta p\left(z_t|x_t\right) p\left(x_t|x_{0:t-1}, z_{1:t-1}, u_{1:t}\right) p\left(x_{0:t-1}|z_{1:t-1}, u_{1:t}\right) \\ & = \eta p\left(z_t|x_t\right) p\left(x_t|x_{t-1}, u_t\right) p\left(x_{0:t-1}|z_{1:t-1}, u_{1:t-1}\right) \end{aligned} \tag{7}$$

The result in (7) shows the the relationship between posterior and likelihood. The particles is update according to odometry information. However, the difference between proposal distribution and target distribution will increase after several propagations. We need to adjust the distribution by resampling according to the importance weight:

$$w^{[m]} = \frac{f\left(x^{[m]}\right)}{g\left(x^{[m]}\right)} \tag{8}$$

$f(x^{[m]})$ is the target distribution and $g(x^{[m]})$ is the proposal distribution. Just as RBPF in [9], our algorithm also uses improved proposal distribution and adaptive resampling, which ensures good local continuity and computational efficiency.

3) BEAM PROJECTION MODEL

Generally, the range reading data consists of the angle and the corresponding distance. So, the error of range reading data consists of angle error (tangential error) and distant error (radial error). We assume both of them follows Gaussian distribution and are independent of another.

$$\rho \sim N\left(\mu_\rho, \sigma_\rho\right) \tag{9}$$

$$\theta \sim N\left(\mu_\theta, \sigma_\theta\right) \tag{10}$$

θ means the real angle, ρ means the real distance, μ_ρ means the distant reading and μ_θ means the angle. And the variance σ_ρ and σ_{theta} are determind by the lidar. The range readings are represent in polar coordinates. The coordinates in the polar coordinate system is convert to the Cartesian coordinate system by (11-14), where x and y are the coordinate of point in the field, f_ρ and f_θ are the probability distribution function corresponding to the Gaussian distribution in (9-10).

$$x = \rho * \sin(\theta) \tag{11}$$

$$\mathrm{y} = \rho * \cos(\theta) \tag{12}$$

$$p(x, y) = f_\rho(\rho) * f_\theta(\theta) \tag{13}$$

$$\iint p(x, y)dxdy = 1 \tag{14}$$

In general, the variance of the two Gaussian distribution is constant when the lidar work. So as the distance to obstacle increase, the projection of the beam in the tangential direction will become more scattered, as shown in Fig. 5. So the probability field model is not suite for short range indoor lidar, but for long range lidar. The model project the probability field into the probability grid map, rather than just project a point into map.

4) SCAN MATCHER ERROR ESTIMATE

Many incremental map building methods use local scan match to get better pose estimation. When given an inital pose, a laser scan and local map, the local scan matcher outputs a new pose maximizing the probabilities that observe the laser scan from map, as shown in Fig. 4. This is a nonlinear least squares problem, as shown in (15).

$$\zeta^* = \arg\min_{\zeta} \sum_{i=1}^{n} \left[1 - M\left(S_i(\zeta)\right)\right]^2 \tag{15}$$

$$S_i(\zeta) = \begin{pmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{pmatrix} \begin{pmatrix} s_{i,x} \\ s_{i,y} \end{pmatrix} + \begin{pmatrix} x \\ y \end{pmatrix} \tag{16}$$

In (15) and (16), $\zeta = (x, y, \psi)$ indicates the pose of the lidar in the world coordinate system; n indicates the number of scan points; and $S_i(\zeta)$ is a function of ζ (Equation (16)), which converts a laser scan point $s_i = (s_{i,x}, s_{i,y})$ from lidar coordinate system to the world coordinate system. $M(S_i(\zeta))$ will return the distance to nearst occupied grid as the coordinates given by $S_i(\zeta)$.

So considering the local environment is unknown, we cannot propose a distribution to estimate scan match error. So we

FIGURE 6. Example of usable ratio. The green area is unusable and the red is usable area.

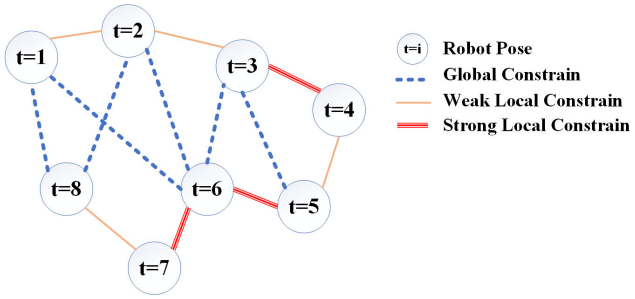


FIGURE 7. Example of global pose graph optimization with different constraints.

construct a particle swarm around the initial value by Gaussian distribution, perform scan match for each particle, and calculate the standard variance of pose deviation of each particle. The the standard variance reveals the degree of particle dispersion after scan match. The more local maximum, the bigger the standard variance. In our implementation, the Gaussian distribution is realized by sample from motion model.

C. BACK-END OF LCPF

The back-end of LCPF consists of loop detection and correction, lag local constraint rectification. Lag local constraint rectification means use branch and bound scan match to rectify local constraint generated by local scan matcher. The constraint between two submaps means the relative pose between them. Considering it costs much time to initialize the branch and bound scan matcher, the rectification is always later than local scan matcher. We use multi-criterion to determine whether a loop is true or false.

1) MULTI-CRITERION LOOP DETECTION

In order to increase the reliability of loop detection and reduce the cost of loop detection, we need to avoid unreliable laser scan join the global scan match. Therefore, we propose a parameter called usable ratio to measure the reliability of a laser scan for global loop detection:

$$\mathrm{R}\left(l_{\mathrm{i}}\right)=\frac{range_{\mathrm{valid}}}{range_{\mathrm{max}}-range_{\mathrm{min}}} \tag{17}$$

The usable ratio $\mathrm{R}\left(l_{\mathrm{i}}\right)$ is the proportion of valid range reading of lidar. Generally, the higher the ratio, the more information the laser scan contains. As shown in Fig. 6, the figure shows the observation of the lidar in the different areas of the map with the same parameters (max range: 5 m, angle range: 180 degrees), the red area indicates the lidar data in the effective range, and the green area indicates the lidar data outside the effective range. The observation in Fig. 6(a) contain more information about local environment and has higher usable ratio. The usable ratio of laser scan that joins loop detection must be bigger than the threshold τ_R.

$$\mathrm{score}\left(l_i, m, x_i\right) * \mathrm{R}\left(l_i\right) \geq \tau_{\mathrm{score}} \tag{18}$$

$$\mathrm{R}\left(l_i\right) \geq \tau_R \tag{19}$$

The smaller the threshold, the more chance global loop detection finds a false loop. So the score is weighted by usable ratio and compared to the score threshold τ_{score}. Only when the score and usable ratio satisfy (18) and (19), will the loop closure be added as a global constrain. In the case of indoor localization and mapping, the further the maximum distance a lidar has, the higher the usable ratio.

2) LAG LOCAL CONSTRAINT RECTIFICATION

We uses local scan matcher to calculate the relative pose between two adjacent submaps. As mentioned above, local scan matcher cannot always find the right answer. And the branch and bound method can search the entire space of plausible solutions, so it is not affected by local maximum. If we apply branch and bound method to calculate relative pose between two adjacent submaps, we can get more reliable result. However, it always costs lots of time to finish the precomputation of branch and bound scan matcher. So we first use local scan matcher to calculate local constraint, then use branch and bound scan matcher to rectify it when the precomputation finished. When creating a new submap, we uses the first laser scan that inserted to new submap to match (local scan match) the previous submap to get relative pose between two submaps. After several seconds, we uses the same laser scan to match (branch and bound scan match) the same submap.

3) GLOBAL POSE GRAPH OPTIMIZATION

A typical pose graph are shown in Fig. 7. This figure shows a typical pose graph, where the circle represents the node, each node corresponds to the origin of a submap in the global coordinate system, and the number in the circle represents the order the submap was created, '1' means the first submap. The lines represent the constraints between the submaps. The blue dotted lines represent the global constraints, the thin orange lines represent local constraints with low weight, and red thick lines represent the local constraints with high weight. There are two kinds local constraint in the pose graph, the one with high weight means it gets high score after local scan matching while another doesn't. The larger the weight of constraint, the more complete the constraint is retained after loop closure. Generally, loop closure is a over-constraint problem. Loop closure is implemented by

pose graph optimization, which is defined as adjusting the pose of submaps and laser scans to minimize the residuals established by both local and global constraints. So it is also formulated as a nonlinear least squares problem in (20).

$$\arg\min_{\Xi^m,\Xi^s}\frac{1}{2}\sum_{ij}\rho\left(E^2\left(\xi_i^m,\xi_j^s;\Sigma_{ij},\xi_{ij}\right)\right) \tag{20}$$

The submap poses $\Xi^m=\left\{\xi_i^m\right\}_{i=1,\ldots,m}$ and the laser scan poses $\Xi^s=\left\{\xi_j^s\right\}_{j=1,\ldots,n}$ under the global coordinate are optimized given some constraints. Given a pair of submap i and laser scan (j), the pose ξ_{ij} is pose where the laser scan matched under the submap coordinate. The residual E for such a constraint is computed by (21) and (22). The covariance Σ_{ij} defines the proportion of translation error and rotation error in E^2, can be evaluated by [25].

$$E^2\left(\xi_i^m,\zeta_j^s;\Sigma_{ij},\xi_{ij}\right)=e\left(\xi_i^m,\xi_j^s;\xi_{ij}\right)^T\Sigma_{ij}^{-1}e\left(\xi_i^m,\xi_j^s;\xi_{ij}\right) \tag{21}$$

$$e\left(\xi_i^m,\xi_j^s;\xi_{ij}\right)=\xi_{ij}-\begin{pmatrix}R_{\xi_i^m}^{-1}\left(t_{\xi_i^m}-t_{\zeta_j^s}\right)\\ \zeta_{i;\theta}^m-\zeta_{j;\theta}^s\end{pmatrix} \tag{22}$$

For pose graph optimization, it is vital to add correct loop closure constraints to the pose graph. So we introduce multi-criterion in loop detection to ensure only true loops will be added. And lag local constraint rectification is applied to avoid local scan match from fall into local maximum.

4) ADAPTABILITY TO SCALE

The submap segmentation parameters of this algorithm do not add the constraints corresponding to the size of the environment. A large environment mostly consists of several smaller environments. If the same submap segmentation threshold is used in both large and small environment, it will cause that a parameter suitable for a small environment generates a large number of extra submaps in a large environment, which will certainly lead to the low efficiency of loop correction and global pose optimization. But in fact, every algorithm is faced with the tradeoff between absolute error and calculation efficiency. Besides, in LCPF, if we set the segmentation threshold of submap as an adaptive parameter for environment scale instead of a run-time fixed but off-line adjustable parameter, this is only to fix the relative error of the algorithm. In large environment, the number of segmentations will be reduced if the adaptive threshold is applied. In this way, the absolute error will still increase with the scale expansion of environment. In practical application, in order to complete a specific task, we need to know the absolute error instead of the relative error of the algorithm under the limited computing resources to control the performance. Because we need to know the absolute distance that the robot may deviate from the true value. In addition, the experiment in this paper is carried out in two scenes with large scale difference, where our algorithm shows good performance in overall map error and loop detection.

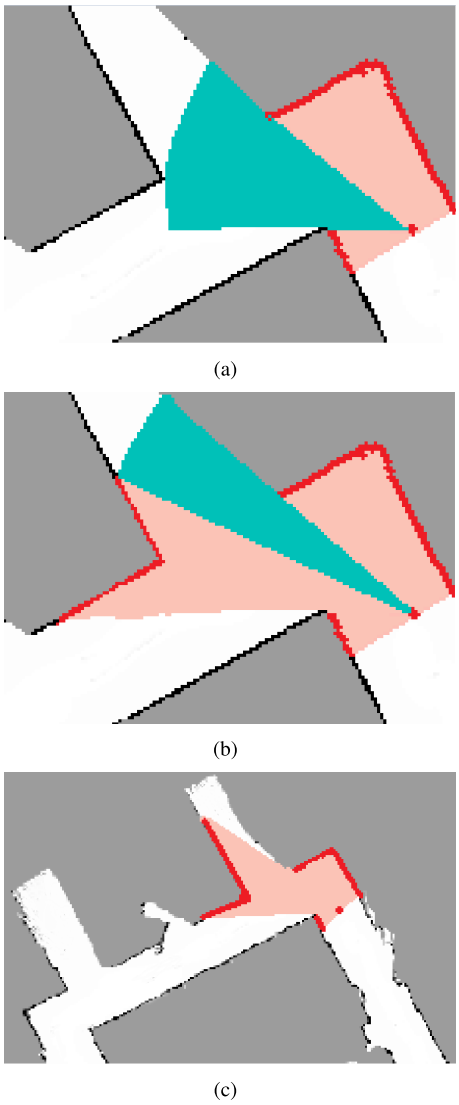


FIGURE 8. The influence of environment scale and lidar maximum range on usable ratio. (a) and (b) use different lidars in the same large environment. (a) and (c) use the same lidar in environments with different scale.

Fig. 8(a)(c) are the data acquired by the same lidar in environments of different scale. During loop detection, with lidar range fixed, when the environment is large, the usable ratio is often very low (shown in Fig. 8(c)), and the absolute pose uncertainty is very high. On the contrary, when the environment is small, the usable ratio will be closer to 1 (shown in Fig. 8(c)), and the uncertainty of absolute pose is low. Therefore, the verification of loop detection with usable ratio has already considered the uncertainty brought by large scale of environment. Then we discuss the case that the lidar's maximum range changes with environment size (shown in Fig. 8(a)(b)). In other word, lidar with short range is used in small scenes (Fig. 8(a)) and lidar with large range is used in large scenes (Fig. 8(b)). In large scenes, the larger distance measurement will lead to higher uncertainty according to the beam projection model. In this way, the projection

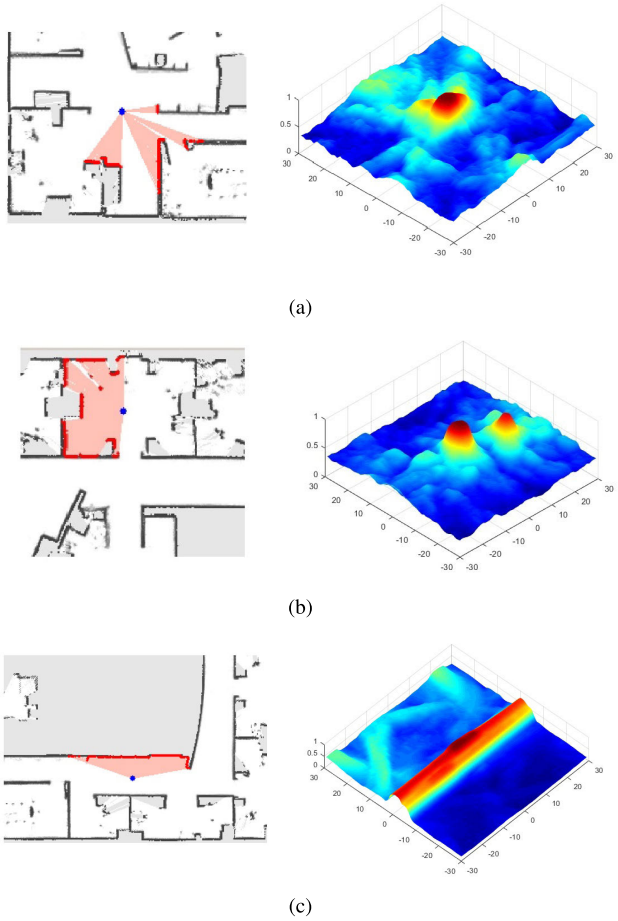


FIGURE 9. Score distribution in different situation. The left represents real maps and laser scan information, the right shows scores on different pixels.

model has also considered the adaptability to the scale of scenes.

IV. EXPERIMENT

A. LOCAL OPTIMAL EXPERIMENT

Existing method such as [9] uses hill-climbing to search for the best match. So the local maximum may cause the result far from global maximum. Serveral possible situations are shown in Fig. 9. Diffirent local environment leads to diffirent local maximum situation. As is shown in Fig. 9 (c), the lidar faces the wall and the point cloud shapes like a straight line, so the point cloud has a higher matching score for the same wall in the direction along the wall. In the image on the right, it is embodied as a long ridged area. In Fig. 9 (b), the area has obvious symmetry, so there are two peaks. One of them is lower than the other, because the left and right environments are not totally the same, and the higher peak on the left corresponds to the real lidar position. In Fig. 9 (a), the area illuminated by the laser radar is significantly different from the surrounding environment, so the matching score shows a single peak, and the peak corresponding to the actual position of the lidar, which is an ideal situation.

TABLE 1. Dynamic submap segmentation experiment result.

Result	Experiment in (a)	Experiment in (b)
Threshold	0.03	0.2
Fixed-seg Error (%)	0.0495	0.1636
Dynamic-seg Error (%)	0.0437	0.1418
Error Decline (%)	0.0058	0.0218
Decline ratio (%)	0.1172	0.1333

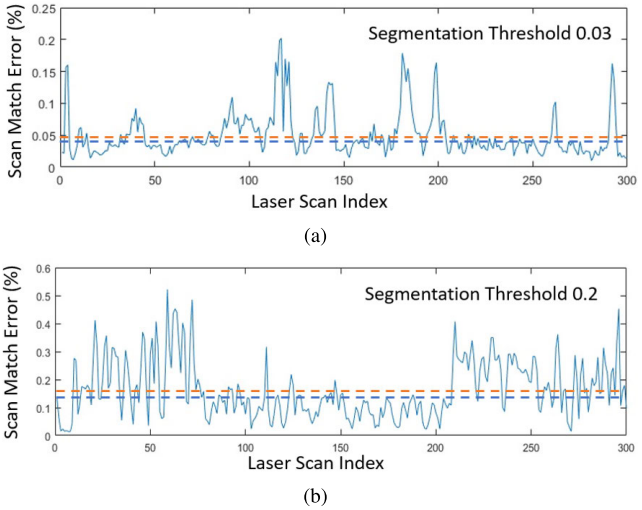


FIGURE 10. Experiment on scan match error relative to the segmentation threshold. In the figure, the horizontal axis is the index of the laser scan, which increases with time, and the vertical axis is the scan matching error estimate corresponding to the laser scan.

In Fig. 9, all the blue points in the left picture are the real position of the lidar. The red point is the laser scan provided by the lidar, the white area represents the passable area (the laser beam can pass), the black indicates the obstacle (the laser beam is blocked) and the gray indicates the unknown area (obstructing the laser beam passing). All the right pictures represent the values of matching scores (0∼1.0) in the neighborhood of 60*60 pixels near the real position corresponding to the left image. When the prior pose predicted by odometry is not accurate, the optimization result can converge to a wrong local optimal. When the scan match error is large and the particles are difficult to converge after many iterations, our algorithm can divide the map at this pose and leave the error for later optimization. Thus better global consistency can be obtained.

B. DYNAMIC SUBMAP SEGMENTATION EXPERIMENT

In order to verify that the dynamic submap segmentation can reduce the scan matching error inside the submaps, dynamic submap segmentation and fixed submap segmentation (submap consists of a fixed number of laser scan) are used under the same two dataset. The scan matching error and the error inside the submap are recorded, as shown in Fig. 10. For fixed submap segmentation, each submap contains 10 laser scans. For dynamic segmentation, the segmentation threshold corresponding to Fig. 10(a) is 0.03, Fig. 10(b) is 0.2.

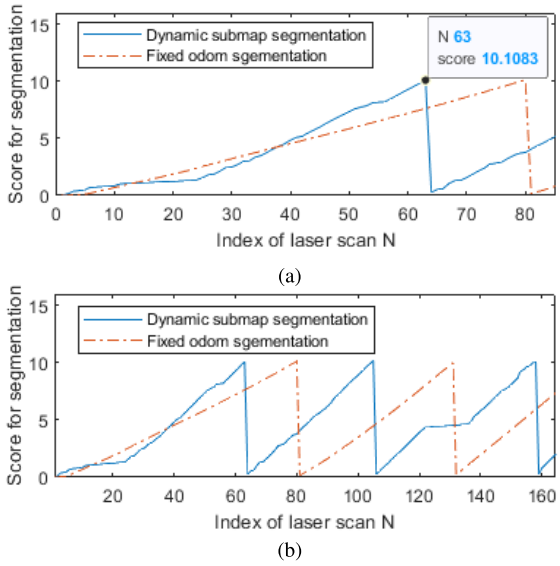


FIGURE 11. Scan match error gradually accumulates, and the score for segmentation increases rapidly when the error of front-end is large. When the score exceeds the threshold, segmentation is applied, and the cumulative score is reset.

In Fig. 10(a), the average scan match error inside submap for fixed segmentation is 0.0495 (orange dotted line), while the dynamic segmentation is 0.0437 (blue dotted line). Our algorithm decrease the scan match error to 80% of fixed segmentation.

In Fig. 10(b), the average scan match error for fixed segmentation is 0.1636, while the dynamic segmentation is 0.1418 (86.67% of fixed segmentation). Therefore, the dynamic submap segmentation leads to less scan match error inside submap than fixed submap segmentation. Comparing Fig. 10(a) and Fig. 10(b), a high threshold leads to high scan match error. For a map with detailed geometric features, a low threshold is suitable for higher precision division, but causes more computational cost. High threshold can guarantee a quick global pose graph optimization, but the small number of submaps can lose the global consistency. In Fig. 11, when the accumulation score (different from the scan match error threshold in Fig. 10) exceed a predetermined threshold, a breakpoint is added to the map. Dynamic submap segmentation has better division flexibility and helps to add the breakpoint to where there really needs one instead of using a fixed frame segmentation.

C. MAP BUILDING EXPERIMENT ON DATASETS

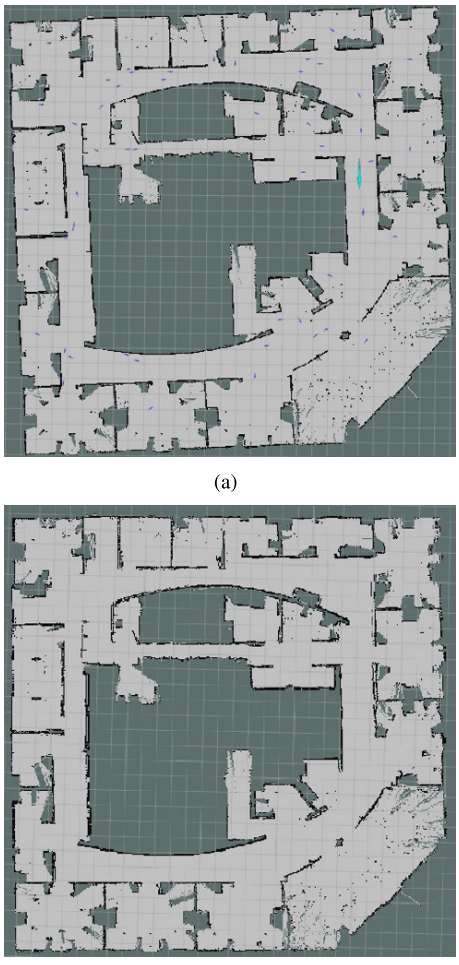
In this experiment, we compare our algorithm with RBPF on Intel Research Lab raw dataset in order to reflect the effect of dynamic submap segmentation and loop closure. As is shown in Fig. 12, (a) is built by RBPF, and (b) is built by LCPF. The dataset record laser scan and odometry information about 40 minutes. After long-term running with odometry noise, accumulative error makes the latest map overlap the old ones without correction. The dislocation of map makes the map unusable for localization or later navigation. Our algorithm performs better after global pose graph optimization.

FIGURE 12. When the error of front-end is large, map built by RBPF (a) gradually rotated a certain angle for lack of global consistency.

By adding loop detection and loop correction, the LCPF can correct the accumulative error and build a map with better global consistency. Maps merge in places that trigger loop correction, making the map available for localization and navigation.

Generally speaking, map built by RBPF has better local continuity, which means the map already built by a particle will not change in the future. This leads to a smoother map but the accumulative error can result in global shifting. After adding loop detection and correction, the local continuity can decline, which is characterized by local ghosting and dislocation. However, the global consistency is guaranteed after global pose graph optimization. As in Fig. 12(a), the map rotates after the robot's long-term running. In Fig. 12(b), we rectify the rotation by loop correction.

In Fig. 13, we test our algorithm and RBPF on the same Intel Reserach Lab dataset when the front-end error is low. Both algorithms naturally perform better than in Fig. 12. However, the rotation of the map in Fig. 13(a) still reflect the accumulative error in RBPF, which is effectively solved in LCPF by loop correction.



(a)

(b)

FIGURE 13. When the front-end is accurate enough, LCPF (b) still has better global consistency than RBPF (a).

D. REAL ROBOT MAPPING EXPERIMENT

In order to prove the practicability of our algorithm, we test LCPF on a real robot. Equipped with coded disk and two-dimensional lidar, our robot can build map with LCPF in an indoor environment. The robot is shown in Fig. 14. The CPU used for experiment is an Intel Core i7-6700HQ 2.60GHz. The system is built on Robot Operating System (ROS). In order to reflect the effect of loop closure, we choose a large closed-loop scene with a long corridor as the experimental scene. The robot moves at walking speed about 0.5m/s in the corridor and builds the map in real time using the collected odometry and laser scan data. In the experiment, as is shown in Fig. 15(a), the accumulative error occurs after long translation along the corridor, and dislocation happens at the upper-right corner of the map. As expected, when the robot arrives at the corner again, loop detection of LCPF finds the loop at the corner. The accumulative error is rectified by later loop correction and global pose graph optimization.

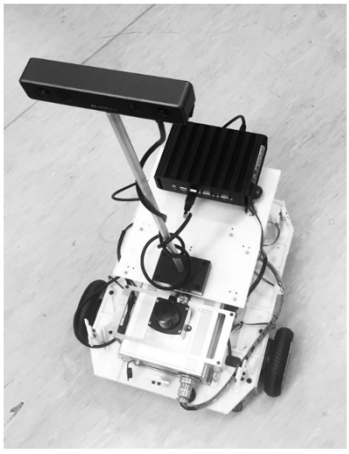
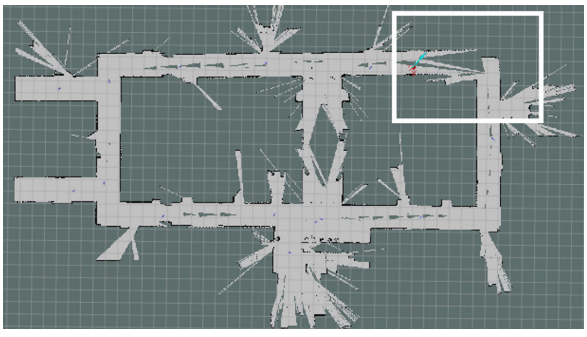
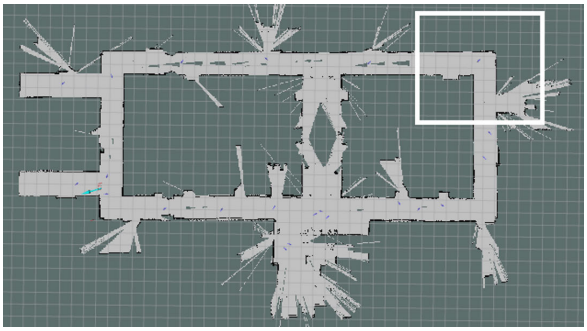


FIGURE 14. Our robot platform carries a two-dimensional lidar on the front and encoder on wheels.



(a)



(b)

FIGURE 15. Experiment on loop closure. A loop is detected in (a) and corrected in (b).

V. CONCLUSION

We proposed a particle filter lidar SLAM system with loop detection and correction. The proposed method has improved the performance of loop detection and the global consistency of map by adding loop closure to traditional RBPF. On the one hand, comparing with the existing particle filter SLAM, LCPF can eliminate the accumulated error through loop detection and correction. On the other hand, LCPF makes the error inside the submap smaller by dynamic submap segmentation compared with the traditional fixed submap segmentation. To testify the validity of our algorithm, we analysis

the scan match error of fixed segmentation and dynamic segmentation, and find the error inside the submap decreases by about 10%. Map building experiments on open datasets and our real robot prove that our algorithm have better global consistency and can rectify the accumulative error and build a better map.

The proposed method can be improved by parallel computing technique and more efficient retrieval algorithm in front end such as DBoW or structural unit encoding. Higher computing efficiency can ensure the real-time performance and accuracy of loop detection, which is our future work.

REFERENCES

[1] M. Li, H. Zhu, S. You, L. Wang, and C. Tang, "Efficient laser-based 3D SLAM for coal mine rescue robots," *IEEE Access*, vol. 7, pp. 14124–14138, 2019.

[2] M. Aldibaja, N. Suganuma, and K. Yoneda, "Robust intensity-based localization method for autonomous driving on snow–wet road surface," *IEEE Trans. Ind. Informat.*, vol. 13, no. 5, pp. 2369–2378, Oct. 2017.

[3] L. Chen, L. Sun, T. Yang, L. Fan, K. Huang, and Z. Xuanyuan, "RGB-T SLAM: A flexible SLAM framework by combining appearance and thermal information," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 5682–5687.

[4] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[5] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[6] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot, "Consistency of the EKF-SLAM algorithm," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2006, pp. 3562–3568.

[7] S. Huang and G. Dissanayake, "Convergence and consistency analysis for extended Kalman filter based SLAM," *IEEE Trans. Robot.*, vol. 23, no. 5, pp. 1036–1049, Oct. 2007.

[8] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "Analysis and improvement of the consistency of extended Kalman filter based SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008, pp. 473–479.

[9] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Trans. Robot.*, vol. 23, no. 1, pp. 34–46, Feb. 2007.

[10] L. Carlone, M. K. Ng, J. Du, B. Bona, and M. Indri, "Rao-blackwellized particle filters multi robot SLAM with unknown initial correspondences and limited communication," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 243–249.

[11] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LIDAR SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 1271–1278.

[12] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.

[13] M. Montemerlo and S. Thrun, "Simultaneous localization and mapping with unknown data association using FastSLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 2, Mar. 2004, pp. 1985–1991.

[14] J. Weingarten and R. Siegwart, "EKF-based 3D SLAM for structured environment reconstruction," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Aug. 2005, pp. 3834–3839.

[15] O. Stasse, A. J. Davison, R. Sellaouti, and K. Yokoi, "Real-time 3D SLAM for humanoid robot considering pattern generator information," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2006, pp. 348–355.

[16] L. M. Paz, J. D. Tardós, and J. Neira, "Divide and conquer: EKF SLAM in $o(n)$," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1107–1120, Oct. 2008.

[17] L. Teslić, I. Škrjanc, and G. Klančar, "EKF-based localization of a wheeled mobile robot in structured environments," *J. Intell. Robot. Syst.*, vol. 62, no. 2, pp. 187–203, May 2011.

[18] J. Wen, C. Qian, J. Tang, H. Liu, W. Ye, and X. Fan, "2D LiDAR SLAM back-end optimization with control network constraint for mobile mapping," *Sensors*, vol. 18, no. 11, p. 3668, Oct. 2018, doi: 10.3390/s18113668.

[19] G. Grisettiyz, C. Stachniss, and W. Burgard, "Improving grid-based SLAM with rao-blackwellized particle filters by adaptive proposals and selective resampling," in *Proc. IEEE Int. Conf. Robot. Autom.*, Jan. 2006, pp. 2432–2437.

[20] R. Sim, P. Elinas, and J. J. Little, "A study of the rao-blackwellised particle filter for efficient and accurate vision-based SLAM," *Int. J. Comput. Vis.*, vol. 74, no. 3, pp. 303–318, Jul. 2007.

[21] P. Elinas, R. Sim, and J. Little, "σSLAM: Stereo vision SLAM using the Rao-Blackwellised particle filter and a novel mixture proposal distribution," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Jul. 2006, pp. 1–7.

[22] A. Walcott-Bryant, M. Kaess, H. Johannsson, and J. J. Leonard, "Dynamic pose graph SLAM: Long-term mapping in low dynamic environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 1871–1878.

[23] Y. Latif, C. Cadena, and J. Neira, "Robust loop closing over time for pose graph SLAM," *Int. J. Robot. Res.*, vol. 32, no. 14, pp. 1611–1626, Dec. 2013.

[24] P. De La Puente and D. Rodriguez-Losada, "Feature based graph-SLAM in structured environments," *Auto. Robots*, vol. 37, no. 3, pp. 243–260, Oct. 2014.

[25] E. Olson, "Real-time correlative scan matching," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 4387–4393.

FUYU NIE received the Mechatronical Engineering degree from the Beijing Institute of Technology, in 2019, where he is currently pursuing the M.S. degree with the Bionic Robot Technology, Navigation and Visual SLAM Group. His research interests include localization and semantic mapping.

WEIMIN ZHANG received the B.S., M.S., and Ph.D. degrees in mechatronics engineering from the Beijing Institute of Technology (BIT), Beijing, China, in 1999, 2002, and 2005, respectively. He was a Visiting Scholar with the Department of Modern Mechanical Engineering, Waseda University, Japan, in 2008. He is currently an Associate Professor with the School of Mechatronical Engineering, BIT. His research interests include mechanical design and bionic vision of humanoid robot.

ZHUO YAO was born in China, in 1995. He received the Mechatronic Engineering degree from the Beijing Institute of Technology, in 2017, where he is currently pursuing the M.S. degree with the Robot Navigation and visual SLAM Group, Beijing Advanced Innovation Center for Intelligent Robots and Systems (BAICIRS). His research interests include robot navigation and SLAM.

YONGLIANG SHI received the B.S. and M.S. degrees from the North University of China, in 2013. He is currently pursuing the Ph.D. degree in mechanics engineering with the Beijing Institute of Technology. His research interests include SLAM and semantic localization for robotics.

FANGXING LI (Member, IEEE) received the B.S. degree from the Chiba Institute of Technology, in 2001, the M.S. degree from Chuo University, in 2004, and the Ph.D. degree from the Beijing Institute of Technology. He is currently a Lecturer with the School of Mechatronical Engineering, BIT. His research interests include mechanical design and bionic vision of robot navigation.

QIANG HUANG (Fellow, IEEE) received the B.S. and M.S. degrees in electrical engineering from the Harbin Institute of Technology, China, in 1986 and 1989, respectively, and the Ph.D. degree in mechanical engineering from Waseda University, Japan, in 1996. He was a Research Fellow with the National Institute of Advanced Industrial Science and Technology, Japan, from 1996 to 1999. He was a Research Fellow from The University of Tokyo, Japan, from 1999 to 2000. He is currently a Professor with BIT. He is also the Director of the Intelligent Robotics Institute, and the Director of the Key Laboratory of Biomimetic Robots and Systems, Ministry of Education of China. His research interests include biped locomotion and bio-robotic systems. Prof. Huang received a First Class Prize of Ministry of Education Award for Technology Invention. He serves as chairs in many IEEE conferences, such as the organizing committee chair of the 2006 IEEE/RSJ IROS, a General Chair of the 2017 IEEE ROBIO and the 2018 IEEE-RAS ICHR.