doi:10.3772/j.issn.1006-6748.2021.03.001

An improved Gmapping algorithm based map construction method for indoor mobile robot①

Tao Yong (陶　永)②*,**, Jiang Shan*, Ren Fan*, Wang Tianmiao*, Gao He*
(*School of Mechanical Engineering and Automation, Beihang University, Beijing 100191, P. R. China)
(**Research Institute of Aero-Engine, Beihang University, Beijing 100191, P. R. China)

Abstract

With the rapid development in the service, medical, logistics and other industries, and the increasing demand for unmanned mobile devices, mobile robots with the ability of independent mapping, localization and navigation capabilities have become one of the research hotspots. An accurate map construction is a prerequisite for a mobile robot to achieve autonomous localization and navigation. However, the problems of blurring and missing the borders of obstacles and map boundaries are often faced in the Gmapping algorithm when constructing maps in complex indoor environments. In this pursuit, the present work proposes the development of an improved Gmapping algorithm based on the sparse pose adjustment (SPA) optimizations. The improved Gmapping algorithm is then applied to construct the map of a mobile robot based on single-line Lidar. Experiments show that the improved algorithm could build a more accurate and complete map, reduce the number of particles required for Gmapping, and lower the hardware requirements of the platform, thereby saving and minimizing the computing resources.

Key words: complex indoor environment, single-line Lidar, map construction, improved Gmapping algorithm, sparse pose adjustment (SPA) optimization

0 Introduction

In view of the implementation of the strategies of Industry 4.0, Intelligent Manufacturing, and Made in China 2025, a dramatic progress and prosperity has been achieved in the field of robotics. Indoor mobile robots have become a research hotspot in the field of service robots. At present, studies on indoor mobile robots are mainly centered on the map construction, localization, and navigation. Precise map construction and accuracy of the localization information in an unknown environment are mutually dependent on each other. Independent navigation and path planning of a mobile robot depend on the accurate environmental mapping and localization. Therefore, the map construction and localization are the basic and key technologies in mobile robots.

Simultaneous localization and mapping (SLAM) refers to the process where the mobile robot simultaneously achieves an independent localization and mapping in an unknown environment[1]. Studies on SLAM have become one of the important research aspects concerning the mobile robots[2-3]. Based on the type of sensors, SLAM is classified as laser SLAM and visual SLAM. Laser SLAM is further divided into laser 2D-SLAM and laser 3D-SLAM. Visual SLAM is further divided into monocular LSD-SLAM[4], binocular SLAM[5], RGB-D-SLAM-V2[6] and Kinect Fusion[7]. Both the SLAM approaches are based on the RGB-D camera.

The Lidar-based 2D SLAM algorithm relies on the probability model during map construction with state estimation for the mobile robot at its core. Extended Kalman filter (EKF) SLAM was invented to solve the dynamic nonlinear problems of the mobile robot model[8]. However, this method exhibited the defects of heavy calculation load, susceptibility to environmental noises, and accumulation of linearization error after long-term operation. Ref. [9] proposed the representative PF-based SLAMs that included the approaches FastSLAM 1.0 and FastSLAM 2.0.

Robot operation system (ROS), an open-source operating system, integrates the commonly used laser-

① Supported by the National Key Research and Development of China (No. 2019YFB1600700) and Sichuan Science and Technology Planning Project (No. 2021YFSY0003).
② To whom correspondence should be addressed. E-mail: taoy@buaa.edu.cn
Received on July 28, 2020

based 2D-SLAM[10]. The commonly used laser-based 2D-SLAM algorithms include Hector[11], Gmapping[12-13], and Cartographer[14]. The Hector SLAM algorithm places a high requirement on Lidar in terms of the turnover rate and measurement noise, thus it has certain limitations. The Cartographer algorithm needs to take up more computing resources. Hence, a higher requirement is placed on the hardware configuration of the mobile robots. Integration of the Gmapping algorithm into the ROS is a representative of Rao-Blackwellized Particle Filter-SLAM (RBPF-SLAM), which has been improved based on FastSLAM 2.0. Given the high speed and high accuracy, Gmapping has become one of the most extensively used algorithms in mobile robots. However, in a complex indoor environment, especially when the odometry is subject to noise interferences, the particle filters (PF) in Gmapping can encounter over-filtering problems. As a result, the obstacles and map boundaries are blurred. Scan match may fail due to poor observation or too small overlaps, leading to the missing boundaries of the obstacles and map. This further affects the quality of the constructed map.

A number of researchers have proposed improved methods for RBPF-SLAM to solve the above problems. Ref. [15] described an improved method of RBPF-SLAM, which reduced the algorithm complexity and enhanced the realtime of the system, but impaired its robustness. Ref. [16] addressed the problems of particle degeneracy and particle failure by developing an improved SLAM method. Although this method improved the accuracy and efficiency of map construction, the time complexity of the algorithm was increased. A large number of particles were still needed to achieve the error convergence. Ref. [17] proposed a scan match SLAM approach based on PF, which reduced the problems of memory consumption and closure of the cycle. However, the number of particles needed for the algorithm was directly proportional to the environmental scale, which meant a high consumption of the computing resources. Ref. [18] reported an improved RBPF-SLAM based on the geometric information, which enhanced the system robustness and consistency of environmental construction. However, 500 particles were needed for map construction using this algorithm, thereby placing a high demand on the hardware's computing capability. Ref. [19] improved the FastSLAM 2.0 algorithm based on FC&ASD-PSO. This method reduced the influence of the cumulative error and improved the accuracy of map construction, but increased the complexity of the algorithm.

All in all, although some improvements have been made in the RBPF-SLAM algorithm, the problems of high algorithm complexity and high consumption of computing resources still persist. Moreover, few efforts of improvements have been directed at Gmapping, a representative RBPF-SLAM method. In the present study, the Gmapping algorithm was optimized for its application in the map construction of a mobile robot in a complex indoor environment. The SPA pose-graph optimization was employed and the graph optimization thread was executed as an independent thread to lower the demand of the Gmapping algorithm for computing resources. By using the proposed method, a more accurate and integrated map could be constructed with fewer particles.

1 Construction of a mobile robot model

1.1 Transformation of coordinate system of the mobile robot

The coordinate system of a mobile robot based on Lidar consists of the world coordinate system, robot coordinate system, and Lidar coordinate system. The world coordinate system is an absolute coordinate system fixed in the environment and is represented by using rectangular coordinates. The robot coordinate system and Lidar coordinate system are relative coordinate systems that need to be converted to the world coordinate systems.

Let $\boldsymbol{X}_W\boldsymbol{O}_W\boldsymbol{Y}_W$ be the world coordinate system of a two-wheeled mobile robot and $\boldsymbol{X}_R\boldsymbol{O}_R\boldsymbol{Y}_R$ be the robot coordinate system. The representation is illustrated in Fig. 1. The origin of the robot coordinate system is located at the center of the robot. In the world coordinate system, the coordinates of the origin are $\boldsymbol{X}_W^k = (x_r, y_r, \theta_r)$ that represents the pose of the robot in the world coordinate system.

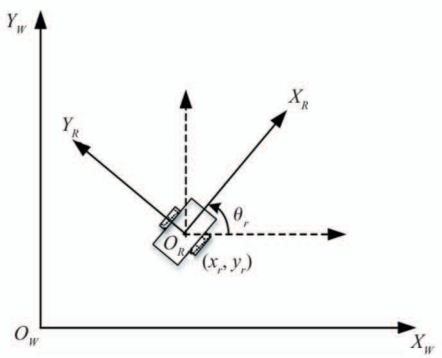


Fig. 1 Coordinate system of the mobile robot

Let the coordinates of a certain point in the robot coordinate system be $\boldsymbol{Q}_0 = [x_0, y_0]^{\mathrm{T}}$, and θ be the included angle between this point and $\boldsymbol{X}_W$. Then, in the

world coordinate system $\boldsymbol{X}_W\boldsymbol{O}_W\boldsymbol{Y}_W$, the coordinates $\boldsymbol{X}_W^0 = [x_W^0, y_W^0, \theta_W^0]$ can be calculated below, as shown in Eq. (1).

$$\begin{bmatrix} x_W^0 \\ y_W^0 \\ \theta_W^0 \end{bmatrix} = \begin{bmatrix} x_r \\ y_r \\ \theta_r \end{bmatrix} + \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ \theta \end{bmatrix} \tag{1}$$

1.2 Lidar model

Lidar observations are used for the map construction of an indoor two-wheeled robot based on single-line Lidar. Since the Lidar observations are represented in the form of polar coordinates, they are converted into the rectangular coordinates. The coordinates of the Lidar observation points are converted into the world coordinate system[2]. As shown in Fig. 2, the distance between the observation point and Lidar is Z_t^k. The included angle between the Lidar measurement point and the robot coordinate system is $\theta_{k,\,sens}$. k is the number of the Lidar measurement point. Let $\boldsymbol{X}_t = (x, y, \theta)$ be the pose of the robot at the moment t. $(x_{k,\,sens}, y_{k,\,sens})$ is the position of the Lidar in the robot coordinate system. Then, the coordinates $\boldsymbol{X}_z^k = (x_z^k, y_z^k, \theta_z^k)$ of the measurement point in $\boldsymbol{X}_W\boldsymbol{O}_W\boldsymbol{Y}_W$ can be calculated from Eq. (2).

$$\begin{bmatrix} x_z^k \\ y_z^k \\ \theta_z^k \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} x_{k,\,sens} \\ y_{k,\,sens} \\ \theta_{k,\,sens} \end{bmatrix} + z_t^k \begin{bmatrix} x_0 \\ y_0 \\ \theta \end{bmatrix} \begin{bmatrix} \cos(\theta + \theta_{k,\,sens}) \\ \sin(\theta + \theta_{k,\,sens}) \\ 0 \end{bmatrix} + \begin{bmatrix} \xi_x \\ \xi_y \\ \xi_\theta \end{bmatrix} \tag{2}$$

where, ξ_x, ξ_y and ξ_θ are the measurement noises, generally assumed to be zero-mean white Gaussian noises[2].

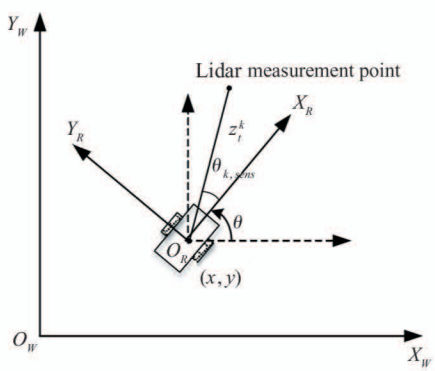


Fig. 2 Coordinate representation of Lidar

1.3 Kinematic model of the mobile robot

In odometry, a device is employed for pose estimation of a mobile robot that collects encoder data per unit time. The robot trajectory model can be obtained by calculating the wheel displacement of a robot per unit time. Then the pose of the robot at a given moment is estimated through integration, and the pose estimation of the robot between adjacent moments is conducted. Pose estimation of the robot between adjacent moments is illustrated in Fig. 3.

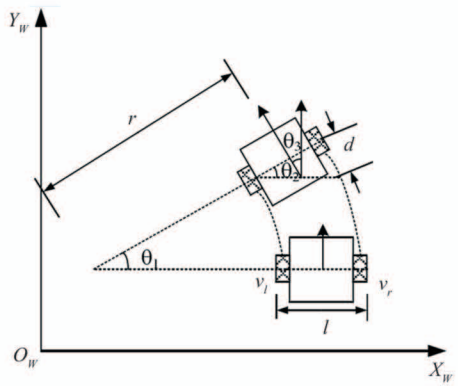


Fig. 3 Pose estimation of the mobile robot between adjacent moments

Let the pose of the robot at the moment $t-1$ be $\boldsymbol{X}_{t-1} = [x_{t-1}, y_{t-1}, \theta_{t-1}]^{\mathrm{T}}$, Δt is the sampling time, v and ω are the linear speed and angular speed of the robot. Then the pose of the robot at the moment t is $\boldsymbol{X}_t = [x_t, y_t, \theta_t]^{\mathrm{T}}$, as given by Eq. (3)[2].

$$\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix} + \begin{bmatrix} v\Delta t\cos(\theta_{t-1} + \omega\Delta t) \\ v\Delta t\sin(\theta_{t-1} + \omega\Delta t) \\ \omega\Delta t \end{bmatrix} \tag{3}$$

2 Map construction method for indoor mobile robot based on the improved Gmapping algorithm

2.1 Gmapping algorithm

The schematic representation of the SLAM process of the mobile robot is shown in Fig. 4.

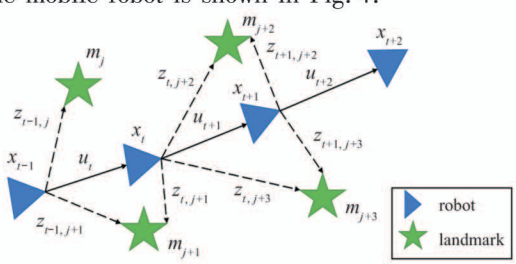


Fig. 4 Schematic representation of the SLAM process of the mobile robot

As shown in Fig. 4, from moment $t-1$ to $t+2$, the pose of the mobile robot is represented by $x_{t-1:t+2}$. The motion control quantity on the odometry is represented by $u_{t-1:t+2}$, the landmark $m_{j:j+3}$, and the scan-

ning observation $z_{t-1:t+1,\ j:j+3}$.

In the Gmapping algorithm, $p(x_{1:t},\ m \mid z_{1:t},\ u_{1:t-1})$ is the posterior probability estimate of the robot, where $x_{1:t}$ is the pose of the robot from the initial moment to moment t; $z_{1:t}$ is the Lidar observation from the initial moment to moment t; $u_{1:t-1}$ is the motion control quantity on the odometry. By introducing the observation into the proposal distribution of robot, the improved proposal distribution of the ith particle at the moment t is obtained, as given by Eq. (4). Whenever a new $(u_{t-1},\ z_t)$ appears, the proposal distribution is recalculated, and the particles are updated as well.

$$p(x_t \mid m_{t-1}^{(i)},\ x_{t-1}^{(i)},\ z_t,\ u_{t-1}) = \frac{p(z_t \mid m_{t-1}^{(i)},\ x_t) \cdot p(x_t \mid x_{t-1}^{(i)},\ u_{t-1})}{p(z_t \mid m_{t-1}^{(i)},\ x_{t-1}^{(i)},\ u_{t-1})} \quad (4)$$

The Gmapping algorithm is an improvement based on FastSLAM. However, the Gmapping algorithm often encounters the problem of over-filtering in a complex environment if the odometry is subject to interferences other than Gaussian noise. In such case, the problems of blurring, missing, and shift of the obstacle borders may occur on the constructed map, which impairs the accuracy and integrity of the map. Here, the Gmapping algorithm was optimized and improved in light of the above problems for map construction. Based on the graph optimization theory, the pose graph of the robot and the closed-loop constraint in the front-end are constructed through the graph optimization thread. The pose graph optimization method based on the sparse matrix was employed in the back-end for global nonlinear optimization during map construction. The robot's pose and map information carried by the particles were corrected. This method not only preserved the advantage of fast map construction in Gmapping, but also lowered the demand for computing resources in the robotic system. Furthermore, both the accuracy and integrity of the constructed map were improved.

2.2 Pose-graph optimization based on SPA

SPA is a nonlinear optimization method used to construct the sparse matrix for the online solving of linear systems[20]. Nonlinear optimization is a global optimization method. When compared with the filter-based algorithms such as EKF, nonlinear optimization effectively eliminates linear and cumulative errors of the system, thereby achieving good optimization effect and satisfactory map construction. Map construction is translated into the problem of finding the optimal nodes that satisfy all of the current constraints. The maps are represented by means of the graph. The objective is to jointly optimize the pose of nodes, thereby minimizing potential errors. The framework diagram of SLAM based on graph optimization is shown in Fig. 5.

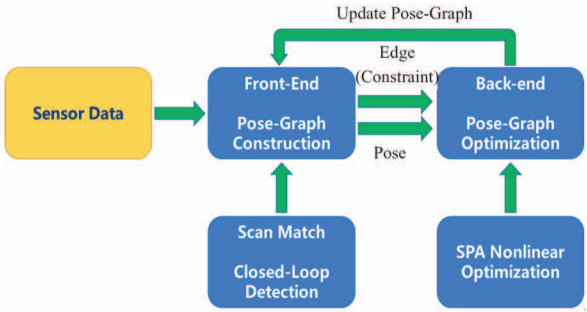


Fig. 5 Architecture of SLAM based on pose-graph optimization

Classical nonlinear optimization methods include the bundle adjustment (BA), Levenberg-Marquardt (LM) algorithm, and Graph-SLAM. However, as the environmental scale increases, the memory needed for the conventional nonlinear optimization increases rapidly. The computing load also increases dramatically, and the time complexity is aggravated. SPA is based on the Cholesky decomposition of a sparse matrix and performs the back-end graph optimization through the construction of a sparse matrix. SPA reduces the demand for memory and lowers the algorithm complexity, therefore it is suitable for map construction in a complex indoor environment.

Using LM as the architecture, the linear system model is constructed with SPA, as shown in Eq. (5).

$$(\boldsymbol{H} + \lambda diag\boldsymbol{H})\Delta x = \boldsymbol{J}^{\mathrm{T}}\boldsymbol{\Lambda} e \quad (5)$$

where, $\boldsymbol{H}$ is the direct sparse feature matrix; λ is the conversion factor from the gradient descent method to the Newton-Euler method; Δx is the correction quantity for the mobile robot's pose; $\boldsymbol{J}$ is the Jacobian matrix of error; $\boldsymbol{\Lambda}$ is the precision matrix, whose value is the reciprocal of covariance; and e is the error function. SPA solves the direct sparse matrix $\boldsymbol{H}$ by Cholesky decomposition, thereby solving the correction quantity for the mobile robot's pose Δx. Online pose estimation of the robot is carried out, and the map is updated incrementally, with the minimization of linearization error of the system. Thus, the map construction and map optimization of the mobile robot are finally realized.

The SPA algorithm is a full nonlinear optimization method, which achieves online continuous pose-graph optimization of the mobile robot by constructing a sparse feature matrix. This method is capable of optimal global estimation of all nodes without occupying too much computing resources. The SPA algorithm is featured by high processing efficiency, low failure rate, and fast convergence. However, as a back-end optimization algorithm, the SPA needs to be combined with the

front-end algorithm with scan match capability. Gmapping can estimate the robot's pose with high speed and accuracy during the front-end scan match, which makes it suitable to be combined with SPA optimization. By incorporating the advantages of the Gmapping algorithm in the scan match, an improved Gmapping algorithm based on SPA optimization is proposed. This method retains the speed of Gmapping in map construction and pose estimation of a mobile robot. The procedure also integrates SPA for optimization and correction of the constructed map and pose of a mobile robot. This further reduces the demand for computing resources and alleviates the problem of over-filtering with PF. The constructed map is more complete and exhibits clear and more accurate boundaries and obstacles.

2.3 Improved Gmapping method based on SPA optimization

Based on SPA, the pose-graph optimization process is executed independently as a single thread to optimize map construction. An improved Gmapping algorithm is proposed.

The improved algorithm consists of the following steps:

Step 1 The pose estimation and particles information is initialized. The pose $x_t^{'(i)}$ of the ith particle estimated from the pose $x_{t-1}^{(i)}$ of the ith particle at the previous moment and the odometry information u_{t-1} are obtained. The calculation is shown in Eq. (6). The proposal distribution p is calculated. The initial particle set $S_t = \{\}$ is defined. The information of all particles $s_{t-1}^{(i)}$ is stored at the previous moment into the particle set S_{t-1} for the previous moment. The particle information $s_{t-1}^{(i)}$ of the previous moment is given by Eq. (7).

$$x^{'(i)} = x_{t-1}^{(i)} \oplus u_{t-1} \tag{6}$$

$$s_{t-1}^{(i)} = \langle x_{t-1}^{(i)}, m_{t-1}^{(i)}, \omega_{t-1}^{(i)} \rangle \tag{7}$$

Step 2 A scan match is performed. Based on map information $m_{t-1}^{(i)}$, estimated pose $x_t^{'(i)}$, and observation z_t, a scan match is performed on the surrounding finite region for the pose $x_t^{'(i)}$ estimated for the ith particle.

If the scan match is successful, the maximum likelihood estimate of the robot's pose $\hat{x}_t^{(i)}$ is solved. The calculation formula is shown in Eq. (8). Decision of the pose-graph optimization thread is done. If the number of robot's pose coordinates reaches 20 in Δt time, start the pose graph optimization thread, then Step 3 and Step 3′ is executed simultaneously; otherwise only Step 3 is executed.

$$\hat{x}_t^{(i)} = \arg\max_x p(x \mid m_{t-1}^{(i)}, z_t, x_t^{'(i)}) \tag{8}$$

If the scan match fails, then Step 3 and Step 4 are skipped. The pose $x_t^{(i)}$ and weight $w_t^{(i)}$ of the ith particle are recalculated, using Eqs(9) and (10), respectively.

$$x_t^{(i)} \sim p(x \mid x_{t-1}^{(i)}, u_{t-1}) \tag{9}$$

$$\omega_t^{(i)} = \omega_{t-1}^{(i)} \cdot p(z_t \mid m_{t-1}^{(i)}, x_t^{(i)}) \tag{10}$$

Step 3 The scan match region $\hat{x}_t^{(i)}$ is sampled. The means and covariance matrices of each point are calculated. The target distribution $p(z_t \mid m_{t-1}^{(i)}, x_j) \cdot p(x_j \mid x_{t-1}^{(i)}, u_{t-1})$ is evaluated at the sampling position x_j. The normalization factor $\eta^{(i)}$ is calculated.

Step 3′ The graph optimization thread is executed. The pose-graph and closed-loop constraint is constructed.

Step 4 The Gaussian approximation of proposal distribution $N(\mu_t^{(i)}, \sum t^{(i)})$ is calculated, using Eq. (11) and Eq. (12). The pose $x_t^{(i)}$ of the ith particle is sampled using Eq. (13).

$$\mu_t^{(i)} = \frac{1}{\eta^{(i)}} \cdot \sum_{j=1}^{K} x_j \cdot p(z_t \mid m_{t-1}^{(i)}, x_j) \cdot p(x_j \mid x_{t-1}^{(i)}, u_{t-1}) \tag{11}$$

$$\sum t^{(i)} = \frac{1}{\eta^{(i)}} \cdot \sum_{j=1}^{K} p(z_t \mid m_{t-1}^{(i)}, x_j) \cdot p(x_j \mid x_{t-1}^{(i)}, u_{t-1}) \cdot (x_j - \mu_t^{(i)})(x_j - \mu_t^{(i)})^{\mathrm{T}} \tag{12}$$

$$x_t^{(i)} \sim N(\mu_t^{(i)}, \sum t^{(i)}) \tag{13}$$

Step 4′ The SPA optimization is executed. The pose correction quantity Δx is calculated by using the equation of the linear system thus constructed. The calculation result is introduced into Step 6 to correct the pose $x_t^{(i)}$ of the ith particle.

Step 5 The weight of the ith particle is updated.

Step 6 The pose and map information $m_t^{(i)}$ of the ith particle is updated. The pose correction quantity Δx calculated from Step 4′ is employed to correct the pose $x_t^{(i)}$ of the ith particle, by using Eq. (14). The map information $m_{t-1}^{(i)}$ of the ith particle is updated through the pose $x_t^{(i)}$ and observation z_t of the ith particle and the map information $m_{t-1}^{(i)}$ of the ith particle at the previous moment, by using Eq. (15).

$$x_t^{(i)} = x_t^{(i)} + \Delta x \tag{14}$$

$$m_t^{(i)} = integrateScan(x_t^i, z_t, m_{t-1}^{(i)}) \tag{15}$$

Step 7 The information of the particle set S_t is updated by using Eq. (16).

$$S_t = S_t \cup \{\langle x_t^{(i)}, m_t^{(i)}, \omega_t^{(i)} \rangle\} \tag{16}$$

Step 8 The completion of the map construction is determined. If yes, then the algorithm workflow is terminated, and Step10 is executed. If the map construction did not finish, then Step 9 is executed.

Step 9 The particles are re-sampled. The effective sample size N_{eff} is calculated and checked whether

it is smaller than the threshold T. Then the need for re-sampling is evaluated. If the effective sample size is smaller than the threshold, then re-sampling is performed. If no, there is no need for re-sampling. The particle re-sampling is done using Eq. (17). If the resampling is successful, go to Step 5; if there is no need for re-sampling, the algorithm returns to Step 1.

$$S_t = resample(S_t) \tag{17}$$

Step 10 The algorithm workflow is terminated.

The workflow of the improved algorithm is presented in Fig. 6.

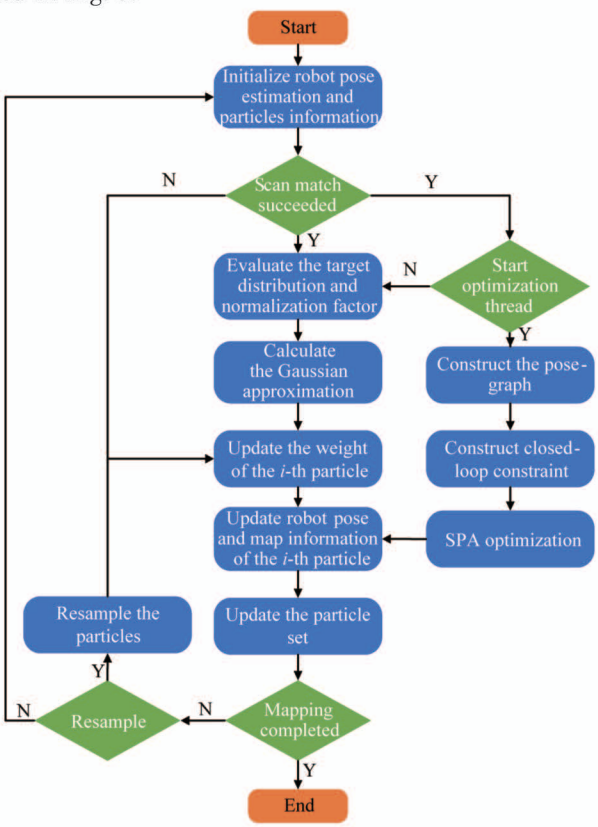


Fig. 6 Workflow of the improved Gmapping algorithm based on SPA optimization

3 Experiment and analysis

3.1 Construction of the experimental robot platform

In this section, the hardware platform for mobile robot, robot communication framework based on ROS, and Stage simulation platform are designed and constructed. The hardware platform mainly consists of LSLIDAR-N301, Intel NUC7i5BNH, STM32 control board, mobile robot chassis, display, and 24 V battery power supply. The structural chart of the system platform is shown in Fig. 7.

LSLIDAR-N301 is applied to acquire real-time measurement data, which is applied to laser SLAM of the mobile robot. The LSLIDAR-N301 is shown in Fig. 8. Intel NUC7i5BNH Mini PC is employed for the following functions, including real-time data processing, map construction, localization, and motion planning, as shown in Fig. 9.

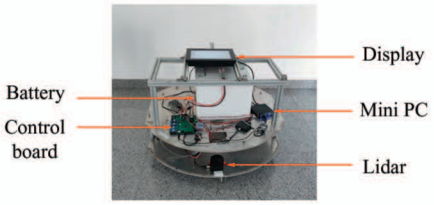


Fig. 7 Structural chart of the system platform

Fig. 8 LSLIDAR N301 model

Fig. 9 Dimensions of the mobile robot on the mini PC

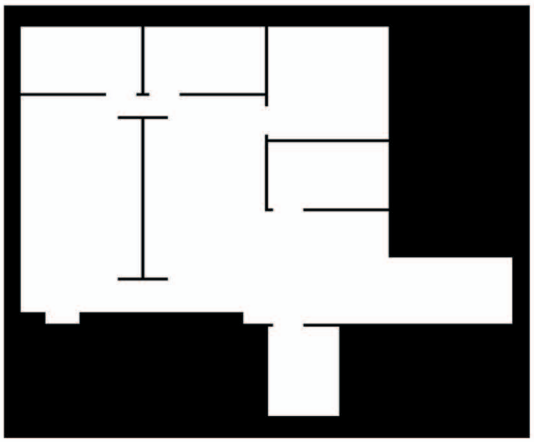
The mobile robot chassis is composed of two driving wheels and two universal driven wheels. The two driving wheels are arranged in a horizontal symmetry across the chassis. They are directly driven by the DC motor control system. The two driven wheels are arranged in a longitudinal symmetry across the chassis to assist the steering. In the experiments, Dell laptop with Ubuntu 16.04, Intel CoreTM i7-7700HQ, 16 GB memory, Stage and ROS Kinetic are used for the robot simulation. The hardware and software configurations fulfill the requirements for simulation experiment.

3.2 Experiment and result analysis

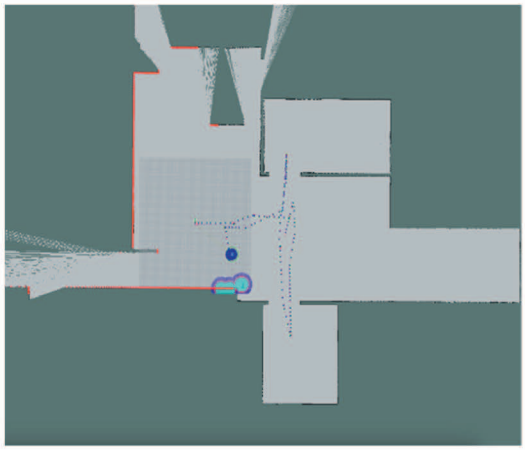
Map construction is performed by controlling the robot with the keyboard. The simulation process is displayed in real-time in the Rviz window and Stage simulator. The binary environmental map used for the experiment is shown in Fig. 10(a), and the map size is 47 m × 33 m. The map construction process shown in the Rviz window is presented in Fig. 10(b). The map resolution is 0.025 m.

Two groups of experiments are conducted. One is to compare the Gmapping algorithm and the improved Gmapping algorithm based on SPA in the performance of map construction under the same particle number

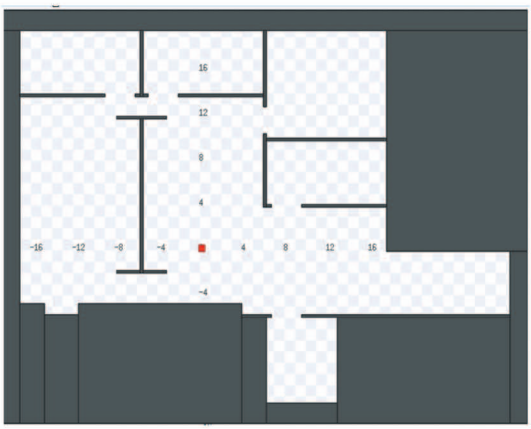
without obstacles. The other is to compare the two algorithms under the same particle number with obstacles. The display interfaces of the experiments on the Stage simulator are shown in Fig. 10(c) and (d). The details of the parameters in the simulation process are shown in Table 1.



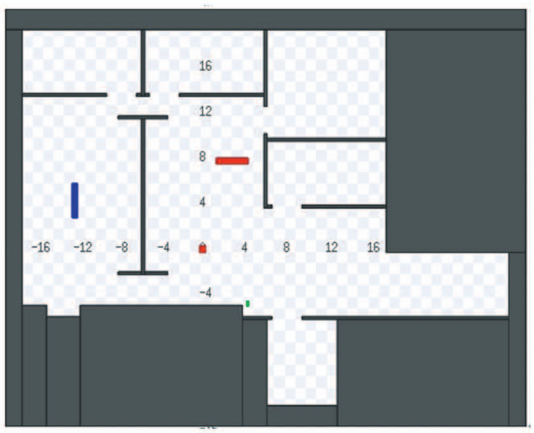
(a) Environmental map for binary simulation



(b) Rviz window displaying the map construction process



(c) Snapshot of Stage simulator without obstacles



(d) Snapshot of Stage simulator with obstacles

Fig. 10 Diagram of the system simulation experiment

Table 1 Table of experimental parameters of the system

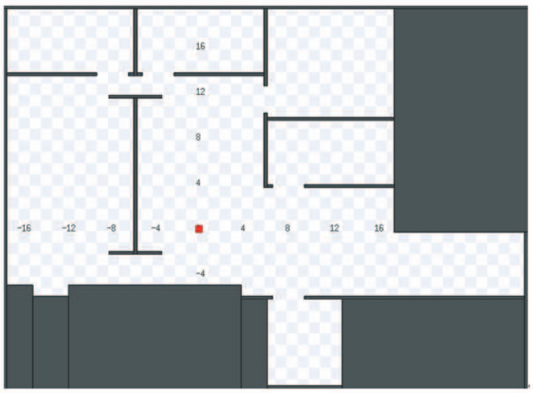
Parameter name	Parameter value
Radius of the mobile robot	0.5 m
Linear speed of the mobile robot	1.5 – 3 m/s
Angular speed of the mobile robot	0.6 – 1 rad/s
Map resolution	0.025 m
Map size	4733 m

3.2.1 Experiment 1: map construction experiment without obstacles

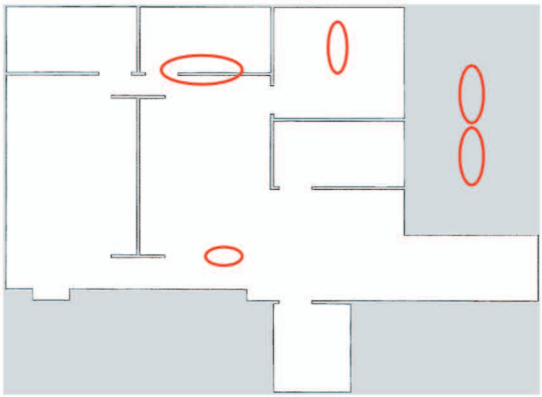
Initially, the Gmapping algorithm and the improved Gmapping algorithm are compared on the basis of SPA for the performance of map construction without obstacles by using the control variable technique. The main parameter influencing map construction using the Gmapping algorithm is the particle number. For a general complex indoor environment, the optimal particle number needed for map construction is 30. Thus, the experiment is conducted by setting up three different particle numbers.

As shown in Fig. 11, (a) is the display of the binary map on the Stage simulator; (b) and (c) are the comparison of the two algorithms under the particle number of 30; (d) and (e) are the comparison of the two algorithms under the particle number of 15; (f) and (g) are the comparison of the two algorithms under the particle number of 5.

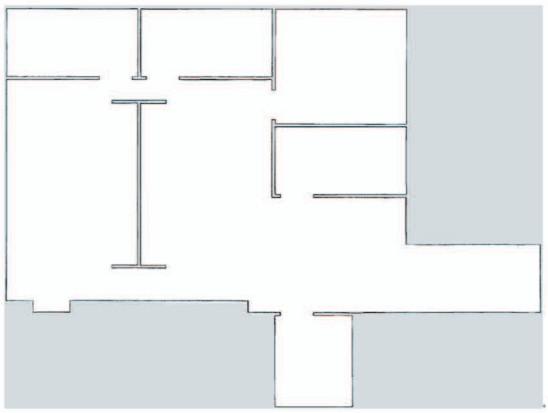
In Fig. 11(b) and Fig. 11(c), the map constructed by Gmapping with the particle number of 30 is fairly accurate. However, in some areas marked in ellipse color in Fig. 11(b), the boundaries are blurred, leading to an incomplete display of the obstacles and map and impairment of map accuracy. The improved Gmapping algorithm based on SPA optimization is executed for real-time detection and optimization of the map features. Highly accurate and clearer map boundaries and obstacle contours are delineated in the above areas. Therefore, the map constructed by improved Gmapping



(a) Map of the experimental environment

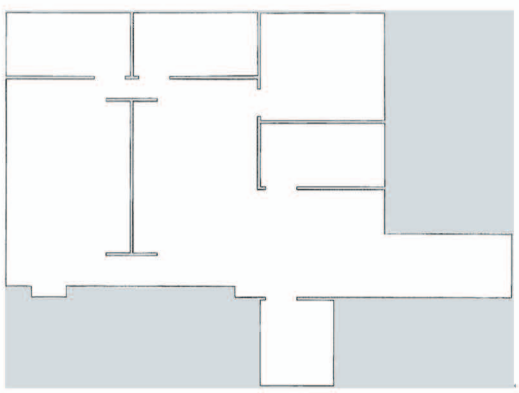


(b) Map construction using the Gmapping algorithm with 30 particles

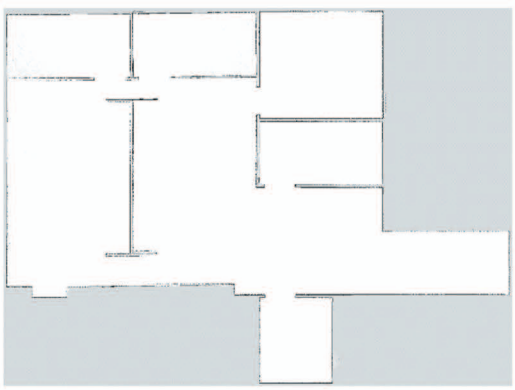


(c) Map construction using the proposed algorithm with 30 particles

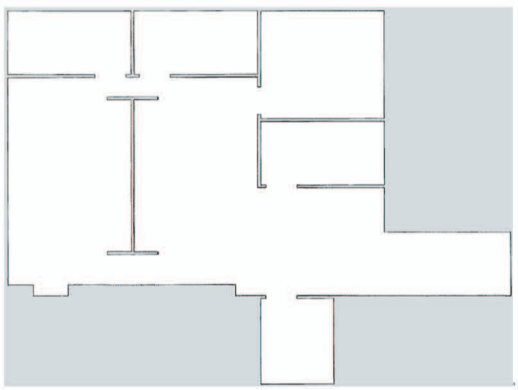
(d) Map construction using the Gmapping algorithm with 15 particles



(e) Map construction using the proposed algorithm with 15 particles



(f) Map construction using the Gmapping algorithm with 5 particles



(g) Map construction using the proposed algorithm with 5 particles

Fig. 11 Map construction experiment without obstacles

algorithm is more accurate.

From Fig. 11(d) and (e), it is clear that the accuracy of map construction using Gmapping under the particle number of 15 decreases to a certain extent. In the areas marked with ellipse in Fig. 11(d), boundaries are missing to a certain extent, thereby affecting the accuracy of map construction. The accuracy of the improved Gmapping algorithm based on SPA optimization slightly decreases for the particle number of 15 than for the particle number of 30. However, a little influence is produced on the accuracy of map construction. As compared with Fig. 11(d), more accurate map boundaries and features are generated in the corresponding areas using the improved algorithm in Fig. 11(e). The map constructed is more complete.

In Fig. 11(f) and (g), when five particles are used to construct the maps, the use of the Gmapping algorithm leads to severe blurring of map boundaries and obstacles. In contrast, the improved algorithm achieves better accuracy for map construction; the obstacle features are clearer, and the map boundaries are more regular.

All in all, in a structured indoor environment, the improved Gmapping algorithm based on SPA optimization could construct highly accurate maps than the conventional Gmapping algorithm, under the same particle number. The advantages of this improved algorithm are even more salient under a small particle number. The improved algorithm exhibites an accuracy of map construction similar to that under a smaller particle number, and with high stability and adaptability.

3.2.2 Experiment 2: map construction experiment with obstacles

In experiment 2, three obstacles are added to make the structured indoor environment more complex. These three obstacles are the longitudinal obstacle of 0.5 m × 3 m × 1.5 m on the left, wide obstacle of 3 m × 0.5 m × 1.5 m in the middle, and a pedestrian of 0.2 m × 0.4 m × 1.5 m in the corner on the right, respectively (Fig. 12). Two control groups are set up in the experiment, and a comparison between the two algorithms in map construction is conducted under the particle number of 30 and 5, respectively.

The map construction experiment with obstacles is shown in Fig. 12. As shown in Fig. 12(b) and Fig. 12(c), in a complex structured indoor environment, the problems of boundary blurring and distortion occured in areas marked with ellipse using Gmapping under the particle number of 30. In contrast, the improved algorithm has a better adaptability, and the map constructed is more accurate. In Fig. 12(d) and Fig. 12(e), the boundary shift and blurring occurred with the Gmapping algorithm, when the particle number was 5. On the contrary, the improved algorithm exhibits a higher stability, and the map accuracy is higher than that with the conventional Gmapping algorithm.

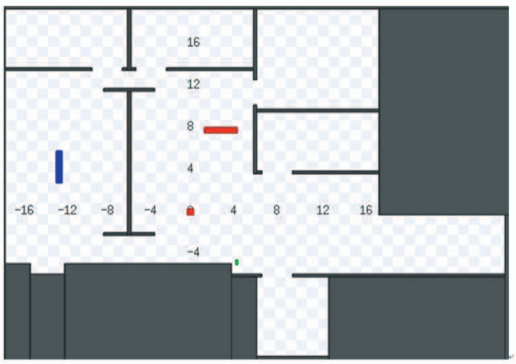
All in all, under the same resolution and minimum match score, the accuracy of map construction using the conventional Gmapping algorithm is considerably influenced by particle number. In contrast, the improved Gmapping algorithm based on SPA optimization is less affected by the particle number. In other words, the improved algorithm does not require a large particle number and too much computing resources for accurate map construction. It shows a better adaptability and places a lower demand on the hardware of the mobile robot. As the particle number decreases, the accuracy of map construction using the Gmapping algorithm decreases as well. Therefore, the algorithm could hardly meet the requirements on map construction in a complex indoor environment under a small particle number. The proposed algorithm is superior to the conventional Gmapping algorithm in 2D map construction for robots under a small particle number, both in accuracy and in integrity.

Under the same particle number, the improved Gmapping algorithm based on SPA optimization exhibits a higher accuracy than the conventional Gmapping algorithm. As shown by the above experiments, the problems of local boundary blurring and boundary distortion occur during map construction with Gmapping. Boundary shift also occurs under a small particle number. The Gmapping algorithm based on SPA optimization offers a higher accuracy than the conventional Gmapping algorithm, and the obstacles and map boundaries constructed are clearer and more complete.

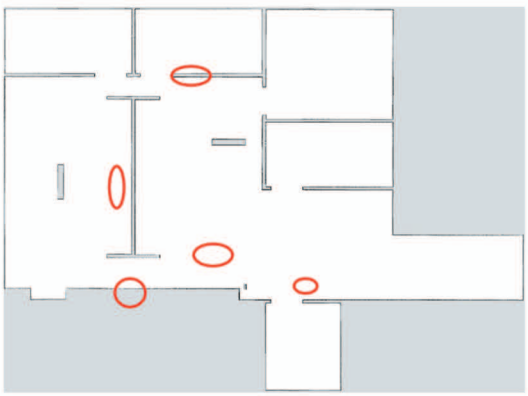
4 Conclusions

Boundary blurring and distortion are the common problems encountered during the 2D Lidar map construction by a mobile robot in a complex indoor environment using the conventional Gmapping algorithm. To address this problem, an improved Gmapping algorithm based on the SPA optimization is proposed, which is subsequently applied in the map construction for the indoor mobile robot. The robot simulation platform based on ROS and Stage is built, and experiments are conducted for comparison between the two algorithms. When compared with the conventional Gmapping algorithm, the improved algorithm needs a smaller particle number for map construction, saves computing resources, and lowers the requirements for hardware. The simulation experiments verify the accuracy, stabili-

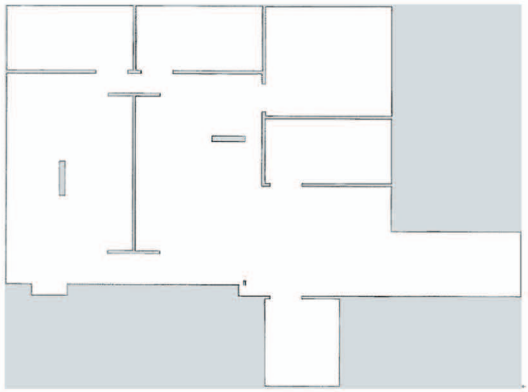
ty, and adaptability of the proposed algorithm. Future study will be centered on the deepened and extended application of the improved Gmapping algorithm based on the SPA optimization in the multi-sensor fusion field.



(a) Map of the experimental environment

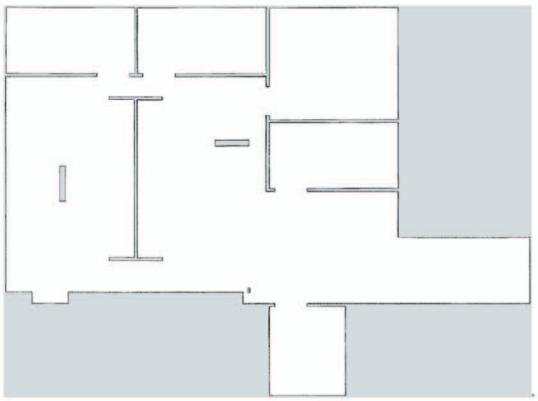


(b) Map construction using the Gmapping algorithm with 30 particles



(c) Map construction using the proposed algorithm with 30 particles

(d) Map construction using the Gmapping algorithm with 5 particles



(e) Map construction using the proposed algorithm with 5 particles

Fig. 12 Map construction experiment with obstacles

References

[1] Yagfarov R, Ivanou M, Afanasyev I. Map comparison of lidar-based 2D SLAM algorithms using precise ground truth[C]// Proceedings of IEEE 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), Singapore, 2018: 1979-1983

[2] Zhao X Y. Research on simultaneous localization and indoor map construction algorithm based on lidar[D]. Harbin: School of Astronautics, Harbin Institute of Technology, 2017: 7-12

[3] Liang M J, Min H Q, Luo R H. Overview of simultaneous localization and map construction based on graph optimization[J]. *Robotics*, 2013, 35(4): 500-512

[4] Engel J, Schöps T, Cremers D. LSD-SLAM: large-scale direct monocular SLAM[C]// Proceedings of Computer Vision-ECCV 2014, Zurich, Switzerland, 2014: 834-849

[5] Wang X H, Li P F. Improved data association method in binocular vision-SLAM[C]// Proceedings of 2010 International Conference on Intelligent Computation Technology and Automation, Changsha, China, 2010: 502-505

[6] Felix E, Jürgen H, Jürgen S, et al. 3-D mapping with an RGB-D camera [J]. *IEEE Transactions on Robotics*, 2014, 30(1): 177-187

[7] Izadi S, Kim D, Hilliges O, et al. KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera[C]//Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, Santa Barbara, USA, 2011: 559-568

[8] Huang S D, Gamini D. Convergence and consistency analysis for extended Kalman filter based SLAM [J]. *IEEE Transactions on Robotics*, 2007, 23(5): 1036-1049

[9] Montemarlo M, Thrun S, Koller D, et al. FastSLAM: a factored solution to the simultaneous localization and mapping problem [C]// Proceedings of the AAAI National Conference on Artificial Intelligence, Edmonton, Canada, 2002: 593-598

[10] Mishra R, Javed A. ROS based service robot platform [C]//Proceedings of the 4th International Conference on Control, Automation and Robotics, Auckland, New Zealand, 2018: 55-59

[11] Kohlbrecher S, Stryk O V, Meyer J, et al. A flexible and scalable SLAM system with full 3D motion estimation[C]//Proceedings of the 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics, Kyoto, Japan, 2011: 155-160

[12] Grisettiyz G, Stachniss C, Burgard W. Improving grid-based SLAM with Rao-Blackwellized particle filters by adaptive proposals and selective resampling [C]// Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, 2005: 2432-2437

[13] Grisettiyz G, Stachniss C, Burgard W. Improved techniques for grid mapping with Rao-Blackwellized particle filters [J]. *IEEE Transactions on Robotics*, 2007, 23(1): 34-46

[14] Hess W, Kohler D, Rapp H, et al. Real-time loop closure in 2D Lidar SLAM [C]// Proceedings of the 2016 IEEE International Conference on Robotics and Automation, Stockholm, Sweden, 2016: 1271-1278

[15] Luo Y, Yu J H, Wang L F, et al. Synchronous localization and map construction of mobile robot with improved RBPF [J], *CAAI Transactions on Intelligent Systems*, 2015, 10(3): 460-464

[16] Wang T C, Cai Y F, Tang Z M. SLAM method based on regional particle swarm optimization and partial Gaussian resampling[J], *Computer Engineering*, 2017(11): 316-322

[17] Xiong H, Chen Y P, Li X P, et al. A scan matching simultaneous localization and mapping algorithm based on particle filter[J], *Industrial Robot*, 2016, 43(6): 607-616

[18] Liu F C, Chen Y F, Li Y Z. Research on indoor robot SLAM of RBPF improved with geometrical characteristic localization[C]//Proceedings of the 29th Chinese Control and Decision Conference, Chongqing, China, 2017: 3325-3330

[19] Lv T Z, Feng M Y. An improved FastSLAM 2.0 algorithm based on FC&ASD-PSO [J], *Robotica*, 2017, 35(9): 1795-1815

[20] Konolige K, Grisetti G, Kümmerle R, et al. Efficient sparse pose adjustment for 2D mapping[C]//Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taibei, China, 2010: 22-29

Tao yong, born in 1979. He received his Ph. D degree in School of Mechanical Engineering and Automation, Beihang University in 2009. He also received his B. S. and M. S. degrees from Chongqing University in 2002 and 2005 respectively. His research interests include intelligent robot integration application, embedded mechatronics and aircraft flexible assembly technology.