Research on SLAM based on RBPF algorithm in indoor environment

Qiang Li1,a, Jia Kang2* and Xiaofang Cao
1,2*Lanzhou Jiaotong University School of Electronics and Information Engineering, Lanzhou, 730070, China
*Corresponding author’s e-mail:690516002@qq.com
dshen@mail.lzjtu.cn

Abstract. In order to realize the simultaneous localization and mapping (SLAM) of robots in indoor environments, a SLAM method for four-wheel mobile robots based on RBPF algorithm and lidar is proposed. The mobile robot realizes its own positioning during the movement. The lidar scans the location of indoor obstacles, updates the map in real time, and gradually realizes the construction of a local map to a global map through data association. Aiming at the particle barrenness that may occur when the RBPF algorithm realizes SLAM during resampling, an adaptive resampling method is adopted to ensure that there are enough particles to realize SLAM every time. The experimental results show that when the linear velocity and angular velocity of the four-wheel mobile robot are small, indoor SLAM can be better realized.

1. Introduction

SLAM for mobile robots has become a research hotspot in the robotics field in recent years [1-2], and is regarded as one of the important technologies for robot autonomy. The research on SLAM issues was mentioned in a paper on SLAM issues published by Smith et al. of Stanford University in the 1980s [3]. In 1991, the concept of SLAM was formally proposed by Leonard et al. [4]. The algorithms used for SLAM mainly include the Kalman filter method and the particle filter method. These two methods are mainly based on the recursive Bayesian state estimation theory to estimate the posterior probability of the system state [5]. It can be realized only when the system's starting time and the observation information and control information at the present time are known. Among the methods based on KF, EKF (Extended Kalman Filter) realizes the SLAM method in the calculation speed, but the accuracy is low, and the data association problem is not easy to solve [6-7]. UKF (Unscented Kalman Filter) implements the SLAM method with a small amount of calculation, but the filter parameters and actual gain cannot be adjusted online in practical applications [8-9]. Among the PF-based SLAM methods, the Rao-Blackwellized Particle Filter (RBPF) method proposed by Doucet et al. is the most representative algorithm [10]. The algorithm assumes that the map feature points are independent of each other, and only need to control the robot. The trajectory is correlated, and the SLAM problem is decomposed into the product form of the posterior probability of the robot path estimation and the posterior probability of the map part. In the research, the calculation and fusion are calculated separately, which makes the calculation efficiency of the PF-based SLAM method greatly improved[11-12].

At present, there are two kinds of SLAM methods that can be used indoors: camera-based SLAM and laser-based SLAM [13-14]. Camera-based SLAM can obtain three-dimensional information, but when the speed is too fast, the image will be blurred. In the face of weak texture or repeated texture, it
will cause the feature to be difficult to extract or mismatch. Laser-based SLAM can realize the construction of indoor two-dimensional maps [15], can display the location of obstacles on the map, and can realize path planning and autonomous navigation on this basis [16].

Therefore, in order to better realize the SLAM of the robot in the indoor environment, this paper proposes to combine the RBPF algorithm to realize the SLAM on the four-wheeled mobile robot equipped with lidar to realize the positioning and map construction in the indoor environment. Aiming at the problem of particle barrenness in resampling when the RBPF algorithm implements SLAM, an adaptive resampling method is adopted to ensure that there are enough particles to achieve SLAM. Through experiments, the linear velocity and angular velocity of the four-wheel mobile robot are tested with different settings, and it is concluded that when the linear velocity and angular velocity of the mobile robot are small, SLAM can be better realized.

2. SLAM based on RBPF algorithm

In the SLAM problem [17], the RBPF algorithm formula is decomposed into:

\[
p(x_t|m,z_{1:t},u_{1:t}) = p(m|x_t,z_{1:t})p(x_t|z_{1:t},u_{1:t})
\]

(1)

Where: \(x\) represents the trajectory of the robot, \(m\) represents the map, \(z\) represents the observation information, and \(u\) represents the odometer information. Use particle filters to estimate the posterior probability of the robot pose, where each particle represents a possible path; The Kalman filter algorithm is used to update the map information. The implementation of SLAM with RBPF algorithm includes the steps of particle initialization, sampling, and particle importance calculation. Figure 1 shows the flow chart of RBPF algorithm to achieve SLAM.

The following are the general steps to implement RBPF-SLAM:

1. Initialization of particles: In the initial unknown environment, select \(N\) particles according to the prior probability \(p(x_0)\) of the machine motion model, denoted as \(x_0^{(i)}(k = 1, 2, 3, ..., N)\), and the initial weight of each particle is \(1/N\), so that \(p(x_0)\) is obtained from the prior probability of the target state;

2. Sampling: According to the proposed distribution \(\pi\), the next generation particle set \(\{x_t^{(i)}\}(k = 1, 2, 3, ..., N)\) is generated from the particle set \(\{x_{t-1}^{(i)}\}(k = 1, 2, 3, ..., N)\), and the odometer motion model \(p(x_t|x_{t-1}, u_{t-1})\) is usually used as the proposed distribution \(\pi\);

3. According to the importance sampling principle, the importance weight of the \(k\)-th particle is
Then the important weights of normalization are:

\[
\tilde{w}_i = \frac{w_i}{\sum_{i=1}^{N} w_i}
\]

(3)

(4) Adaptive resampling: resampling is the process of gradually replacing particles with low weights by sampled particles with high weights, making the particles in the particle collection more concentrated. However, this process will reduce the number of particles and cause the particles to become barren, so that the collection of particles cannot better describe the probability distribution of the state. In order to avoid this problem, the adaptive resampling method is directly adopted. This method introduces the effective particle number \(m_{\text{Neff}}\) to estimate the approximate degree of the current particle to the objective function. The effective particle number \(m_{\text{Neff}}\) reflects the size of the particle weight variance, which can be expressed as

\[
m_{\text{Neff}} = \left( \frac{1}{\sum_{i=1}^{N} \tilde{w}_i} \right)
\]

(4)

When it is less than the set threshold, re-sampling is implemented;

(5) Update the map. Update map \(p(n|x_o,z_o)\) according to the current pose of the particles and historical observation information.

3. Speed correction

3.1. Software platform

The programming operating platform selects an Acer laptop equipped with Intel dual-core, CPU 1.8GHz, and 4GB of memory, and installs a virtual machine-VMware. The virtual machine runs on the host, which has strong independence and the operation of the virtual machine will not affect the host. Install the Ubuntu mate 16.04 system and the robot operating system ROS kinetic on the virtual machine. The ROS system framework contains many nodes, mainly including SLAM, lidar, mileage, basic controller, and Arduino. All nodes are managed in the node manager. Set the mobile robot and the ROS system in the same WIFI environment, and the ROS system will issue control commands to the mobile robot to realize the coordinated work between the nodes, and then obtain the posture, position, and speed of the mobile robot in the three-dimensional space, and then Implement SLAM.

3.2. Speed correction of mobile robot

In order to accurately know the distance and position of the mobile robot in motion, the angular velocity and linear velocity of the robot are corrected to make the movement of the mobile robot itself more accurate. First, set the angular velocity and linear velocity of the robot on the robot operating platform, set the rotation angle and the moving distance of the mobile robot, enter the command to control the movement of the robot, record each measurement data, and then calculate the error and correction. Path test content: The robot rotates in situ at an angular velocity of 3rad/s. The data of each test is recorded in Table 1. From Table 1, it can be seen that the larger the rotation angle, the greater the error. The error is basically within the range of 2°. The laws of turning and reversal are basically the same. During linear velocity correction, the robot is tested for movement at a linear velocity of 0.60m/s. The moving distances are 1m, 2m, and 4m respectively. The linear velocity and the feedback velocity value of the robot are measured with a special tachometer. The measurement data of the linear velocity is shown in Table 2. It can be seen from the table that the linear velocity error is within 4mm/s, and the actual measured velocity is slightly larger than the feedback velocity value. After program modification and improvement, the mobile robot platform can use the keyboard and mouse to control the robot's precise
movement, the robot moves smoothly, and can feedback the status of the mobile robot in real time, ensuring the stability and reliability of the system. In the experiment, the lidar data and the corrected speed are fused to ensure that each data correction has a good improvement in the accuracy of the actual positioning and map construction.

Table 1. Test data of in-situ rotation angle of mobile robot

| Number of measurements | Rotation angle/° | Measured value/° | Feedback value/° | error/° |
|------------------------|------------------|------------------|------------------|---------|
| 1                      | 90               | 90.3             | 90.0             | 0.3     |
| 2                      | 180              | 180.7            | 180.1            | 0.6     |
| 3                      | 270              | 271.2            | 270.1            | 1.1     |
| 4                      | 360              | 361.8            | 360.2            | 1.6     |
| 5                      | -90              | -90.5            | -90.0            | 0.5     |
| 6                      | -180             | -180.9           | -180.1           | 0.8     |
| 7                      | -270             | -271.2           | -270.1           | 1.1     |
| 8                      | -360             | -361.7           | -360.1           | 1.6     |

Table 2. Robot 0.60m/s linear velocity test data

| Number of measurements | Distance/cm | Measured value/m/s | Feedback value/m/s | Error/m/s |
|------------------------|-------------|--------------------|--------------------|-----------|
| 1                      | 100         | 0.603              | 0.601              | 0.002     |
| 2                      | 100         | 0.600              | 0.600              | 0         |
| 3                      | 100         | 0.602              | 0.600              | 0.002     |
| 4                      | 200         | 0.604              | 0.600              | 0.004     |
| 5                      | 200         | 0.604              | 0.601              | 0.003     |
| 6                      | 200         | 0.603              | 0.601              | 0.002     |
| 7                      | 400         | 0.607              | 0.604              | 0.003     |
| 8                      | 400         | 0.605              | 0.601              | 0.004     |
| 9                      | 400         | 0.605              | 0.602              | 0.003     |

4. Experiment and result analysis

4.1. experiment procedure
In this experiment, the linear velocity and angular velocity of the robot were set differently, and SLAM was realized in the same environment. The experiment environment was the reference room of the School of Electronics and Information Engineering, Lanzhou Jiaotong University. When the angular velocity of the robot is 1.5rad/s, the linear velocity is set to 0.15m/s, 0.6m/s, 1.0m/s, respectively, to realize the positioning of the robot and map construction; when the online speed is 0.15m/s, the robot The angular velocity is set to 1.5rad/s, 3 rad/s, and 6 rad/s to realize the positioning and map construction of the robot. Before each test, the linear velocity and angular velocity of the trolley are corrected for error. The experimental steps are as follows:

1. Set the network adapter of the virtual machine to bridge mode, so that the upper computer of the robot can establish communication with the ROS operating system, and realize the control of the robot through remote login on the personal computer;
2. Enter control commands on the robot operating system ROS, so that the virtual machine can control the robot's movement and establish communication with the radar at the same time;
3. The map is updated. View the constructed map and the location of the mobile robot on rviz (3D visualization tool). According to the current position of the mobile robot and the area scanned by the lidar, the robot is controlled to continue to move and build the map.
4.2. Experimental results and analysis

As shown in Figure 2 and Figure 3, they are the 2D grid maps constructed by the four-wheel mobile robot at different linear and angular velocities. In Figure 2, the grid map constructed by the four-wheeled mobile robot when the angular velocity is 1.5 rad/s and the linear velocity is 0.15 m/s, 0.6 m/s, and 1.0 m/s, respectively. Compare Figure 2(a), (b), (c), it can be seen that as the linear velocity of the robot increases, the grid error increases, the quality of the map decreases, and the uncertainty of the map edge increases. When moving at 1.0 m/s, the RBPF algorithm has a large error, and the map has obvious uncertainty. Figure 3 is a grid map constructed when the online velocity is 0.15 m/s and the angular velocity is 1.5 rad/s, 3 rad/s, and 6 rad/s respectively. Comparing Figures 3(a), (b), and (c), we can see that With the increase of angular velocity, the robot's scanning and construction of obstacles at the turn will be incomplete, and the uncertainty of the map will continue to increase. In Figures 2 and 3, we can clearly see the robot's pose in the two-dimensional grid map, and the overall positioning is accurate. In some places, the robot's position and obstacles overlap.

Comparing Figure 2 and Figure 3, it is easy to see that there are common errors and shortcomings in these figures. Where there is glass in the cabinet, there is a slight error between the obstacles in the constructed map and the real environment. The laser points scanned by the lidar are in some places. The place does not overlap the map. In the figure, the positioning of the robot has an error that deviates from the actual channel. The reasons for the error are:

1. The penetration of the laser to the glass will cause the position of the obstacle to deviate from the actual position.
2. As the number of times the robot moves in an unknown environment increases, the cumulative error of the robot will gradually increase. The red dot in Figure 3 represents the boundary point of the obstacle scanned by the lidar. When the control robot passes through the place where it originally walked, it can be seen that there is a slight deviation from the boundary in the map.
3. The noise generated by the four-wheeled robot in motion will also affect the construction of the map.
4. The deviation of the positioning position of the robot is caused by the movement of the robot through the remote operation in an unknown environment. When the robot contacts an obstacle, the operation is improper. The accumulation of the odometer information by the robot leads to the deviation of the positioning.

![Figure 2. Comparison of composition at different linear speeds](image1)

![Figure 3. Comparison of composition of different angular velocities](image2)
5. Conclusion
This paper proposes a four-wheel mobile robot based on RBPF algorithm and lidar to realize SLAM. In the process of RBPF algorithm to realize SLAM, adaptive resampling method is used to overcome particle barrenness. Through the comparison of multiple sets of experiments, it is concluded that when the RBPF algorithm and lidar are combined to achieve SLAM, when the linear velocity and angular velocity of the mobile robot are small, the effect of the map constructed is better.

References
[1] Yang Xuemeng, Yao Minru, Cao Kai. Overview of key issues and solutions for mobile robot SLAM[J]. Computer System Application, 2018, 27(07):1-10.
[2] Wang Zhongli, Zhao Jie, Cai Hegao. Graph construction method based on graph optimization SLAM in large-scale environment[J]. Journal of Harbin Institute of Technology, 2015, 47(01):75-85.
[3] SMITH R, SELF M, CHEESEMAN P. Estimating uncertain spatial relationships in robotics [C]/Proceedings of the Second Annual Conference on Uncertainty in Artificial Intelligence. philadelphia: Elsevier, 1986:435-461.
[4] HOLMES S A, KLEIN G, MURRAY D W. An O (N2) square root unscented Kalman filter for visual simultaneous localization and mapping [J].IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009, 31(7):1251-1263.
[5] Chu Mingshuai. Research on simultaneous localization and map construction of mobile robots based on incremental [D]. Beijing: Beijing Jiaotong University, 2016.
[6] SMITH R, CHEESEMAN P. On the representation and estimation of spatial uncertainty [ J]. The International Journal of Robotics Research, 1986, 5(4): 56-68.
[7] Ji Xiaoling, He Qing, Chi Zongtao. Application of SLAM algorithm based on EKF in robot positioning[J]. Science & Technology Economic Tribune, 2016(13):17-19.
[8] Zhang Wenling, Zhu Mingqing, Chen Zonghai. Adaptive SLAM algorithm based on strong tracking UKF [J]. Robot, 2010, 32(2): 190-195.
[9] Wang X, Zhang H. A UPF-UKF Framework for SLAM[C]// Proceedings 2007 IEEE International Conference on Robotics and Automation, Roma, 2007, pp. 1664-1669.
[10] Wen Guocheng, Zeng Bi, Chen Yunhua. A closed-loop detection method suitable for laser SLAM large-scale maps[J]. Application Research of Computers, 2018, 35(06):1724-1727+1732.
[11] Li Yang, Yu Xiantao, He Yong, Gong Pan, Lei Jintao. Implementation of Lidar SLAM based on Cortex-A53 platform[J]. Electronic Technology Application, 2018, 44(02): 105-108.
[12] Gao Wenyan, Ping Xueliang, Bei Xuying, Chen Wei. Analysis of the optimal parameters of two SLAM algorithms based on lidar[OL]. Sensors and Microsystems, 2018(04):28-30+33[2018-09-12].https://doi.org/10.13873/J.1000-9787 (2018)04-0028-03.
[13] Sun Manhui, Yang Shaowu, Yi Xiaodong, Liu Hengzhu. Autonomous navigation of robots in a large-scale environment based on GIS and SLAM[J]. Chinese Journal of Scientific Instrument, 2017,38(03):586-592.
[14] Quan Meixiang, Park Songhao, Li Guo. Overview of visual SLAM [J]. Journal of Intelligent Systems, 2016, 11(06):768-776.
[15] Gong Wanting, Sun Zuolei, Pan Pan. Robot indoor composition based on two-dimensional laser[J]. Microcomputer and Application, 2017,36(18):55-58+62.
[16] Guo Wenxian, Gao Chenxi, Zhang Zhi, Zhang Lei. Research on indoor robot SLAM based on feature sparse strategy[J]. Computer Engineering and Applications, 2017, 53(16):110-115.