Research on simultaneous localization and mapping of indoor mobile robot

Xingdong Liu¹, Xiao Luo² and Xinliang Zhong¹,³

¹ School of Mechatronic Engineering, Beijing Institute of Technology, Beijing 100081, China
² School of Software, Beijing Institute of Technology, Beijing 100081, China
³ E-mail: xinliangzhong@foxmail.com

Abstract. Simultaneous localization and mapping (SLAM) is the key technology to fulfill mobile robot obstacle avoidance and autonomous navigation. As cameras such as binocular cameras or RGB-D cameras exist wide-angle limitations, the use of such sensors prone to map mismatches and localization errors. In order to solve the aforementioned problems, 2D laser radar and odometer are used as the core sensor to design an experimental system for indoor SLAM, which is configured on a mobile robot to examine the effect of indoor SLAM. Through the Rao-Blackwellized particle filter algorithm, the data of laser radar and odometer are fused with real-time processing, which realizes the construction of map as well as localization and navigation of the mobile robot in the unknown environment. Experiments on simulation and physical verification were carried out with open source robot operating system (ROS). The simulation and experimental results confirm the feasibility and practicability of the platform.

1. Introduction

The problem of simultaneous localization and mapping (SLAM) has been one of the hotspots in robotics and computer vision research communities over the past decade. Service-oriented indoor mobile robots have received extensive attention. However, there are still many key issues that need to be solved in the popularization and application of indoor robots. Real-time positioning and navigation is one of the core issues. In the research of this problem, three key points that need to be focused on: Firstly, the accurate localization of the robot in the unknown environment; Secondly, the map reconstruction of the unknown environment; The third is real-time path planning for Robot [2-4]. The accurate localization is the basis of these three key points in an unknown environment. Although in recent years, many effective solutions have emerged, such as Monte Carlo localization algorithms, simultaneous localization and mapping, A* and Dijkstra path planning algorithms [5]. However, these methods also have some specific problems, which hinder their widespread application.

Most of the robots being researched or in use employ a variety of processor architectures and sensor hardware modules. The performance of the hardware system has been greatly improved. However, as the kernel of the robot, the reusability and portability of the software system gradually declined, which is disadvantageous for the development of a general-purpose mobile platform. In response to this problem, Willow Garage released the open source robot operating system (ROS) in 2010, effectively solved software compatibility issues on different hardware, and provided the software platform foundation for the development of a robot universal platform [5-10].
The main contribution of this paper is to develop a mobile robot platform based on ROS, and specifically designed a set of high-performance indoor localization and navigation solutions for the platform. The simulation and physical verification experiments verify the feasibility and effectiveness of the scheme from the theoretical and technical aspects.

2. ROS system overview

The operating architecture of ROS is the packaging of the TCP/IP network protocol. It is essentially based on network protocol control. Therefore, it is extremely convenient to use ROS to realize communication through the network. The following three types of communication can be easily realized: (1) synchronous remote procedure call communication based on process services; (2) topic-based asynchronous data flow communication; (3) data storage communication on parameter servers [11]. But in essence, ROS is not a real-time operating system. The main features of ROS can be summarized as follows:

1) Point to point design

ROS is not a single-process system, but a multi-tasking, multi-process system. Assuming that these systems are all present on the hardware of a robot, the hardware requirements will become extremely high, and resources may also be wasted. While the point-to-point design can decentralize the task pressure, the task of decentralization is particularly prominent in multi-machine communication.

2) Multi-language support

ROS is an operating system that is neutral to the programming language. It can support C++, Python, LISP, and Octave, and other programming languages can also call ROS interfaces. ROS can do mixed programming, which essentially translates into a processing model for messages that enables multiple languages to play their strengths and complement each other.

3) Streamlined and integrated

Modular is an import characteristic of ROS. The code of each module can be compiled separately or combined. Compiling only requires writing corresponding CMake files.

4) Rich toolkit

In order to manage the complex ROS software framework, there are three-dimensional visualization debugging software similar to Rviz in ROS to facilitate the development of robots. In addition, the system node process and the process parameters can be directly observed in Rqt.

With ROS' distributed processing framework, this paper designs a high-performance mobile robot platform for indoor SLAM, and realizes autonomous navigation function on the platform.

3. Mobile robot platform for indoor SLAM

Localization is the process by which a robot determines its specific location in an unknown environment. Two conditions are needed if the robot wants to locate itself: First, the robot is equipped with various sensors that can help it to sense the unknown environment of the outside world; second, the robot must be able to get a map of the unknown environment that matches the sensor type through the sensor. Localization includes absolute and relative localization. In this paper, the relative localization method of odometry is combined with the absolute localization method of D-lidar, and the two-dimensional pose of the robot $(x, y, \theta)$ [12-13] is obtained by fusing relevant information.

3.1. Platform structure of mobile robot platform

The mobile robot platform designed in this paper consists of four parts: the perception layer, the decision control layer, the execution control layer, and the client layer [14]. As shown in figure 1, sensor data of the perception layer is transmitted to the decision control layer through the serial port. The layer builds maps based on this data and formulates navigation strategies. The most critical of these is the fusion of the data of the odometer and the 2D-lidar and gives the robot's final position in the constructed map, while real-time updating of the environmental map. Rao-Blackwellized particle filter is used to fuse the odometer data and 2D-lidar data [15]. The physical object of the mobile robot platform is shown in figure 2.
In this paper, Rao-Blackwellized particle filter is used to locate and map the mobile robot in 2D-SLAM, and the pose of the mobile robot is represented by different weighted particle sets. The pose estimation can be performed according to the robot's motion model [1], namely the positioning problem, and then complete the construction of the environment map on this basis. The algorithm flow of the mobile robot experimental platform is shown in Figure 3:

Figure 1. The hardware framework of the mobile robot.

Figure 2. Physical objects of the mobile robot.

Figure 3. The algorithm flow chart of the platform.
3.2. Data fusion based on particle filter

The core idea of particle filter is based on the Monte Carlo sampling method. The goal of Monte Carlo positioning is to use particle filter algorithm to calculate the current robot’s position. For two-dimensional space, the Cartesian position coordinates and the heading angle are expressed by \( (X, Y, \theta) \). Monte Carlo sampling is used to solve the integral problem of Bayesian posterior probability. To solve this problem, Monte Carlo sampling is used to replace the posterior probability \([16]\) calculations. The advantage is that it can calculate the nonlinear non-Gaussian state space problem of the space, its core idea is through random sampling in the probabilistic method, the random sample called "particles", particle filter using sample mean instead of integral operation makes on the state space of the value of the random sample approximate equal to the probability density function \([2]\).

The purpose of robot positioning is to obtain the current robot state \( x_k \). However, solving \( x_k \) requires recursive calculation \((3-1)\). For non-Gaussian, nonlinear systems, the analytical solution of this formula does not exist or is difficult to obtain.

\[
p(x_k|z_{1:k}) = p(z_k|x_k) p(x_k|z_{k-1})/p(z_k|z_{1:k-1}) \tag{3-1}
\]

Particle filter algorithm using discrete probability deduction method to solve equation \((3-1)\). Assume that a set of weighted random sampling \( \{x_k^i, \omega_k^i \}_{i=1}^{N_k} \) is used to represent the posterior probability density function \( \hat{p}(x_k|y_{1:k}) \). Where \( \{x_k^i, i = 0 \ldots N_k \} \) is a set of support points, and \( \{\omega_k^i, i = 0 \ldots N_k \} \) is its corresponding weight. Then we can get:

\[
p(x_k|y_{1:k}) \approx \sum_{i=1}^{N_k} \omega_k^i \delta(x_{0:k}^i - x_{0:k}^i) \tag{3-2}
\]

The choice of weight \( \omega_k^i \) is based on the resampling principle. Drawing \( \omega_k^i \propto p(x_{i:k}^i|z_{1:k})/q(x_{0:k}^i|z_{1:k}) \). Where \( q(x_{0:k}^i|z_{1:k}) \) is the particle prediction distribution. With further derivation of equation \((3-1)\) we can obtain:

\[
\omega_k^i = \frac{p(z_k^i|x_k^i)p(x_k^i|x_{k-1}^i)p(x_{0:k-1}^i|z_{1:k-1})}{p(z_k^i|z_{1:k-1})} \times \frac{q(x_k^i|x_{0:k-1}^i,z_{1:k})}{q(x_k^i|x_{0:k-1}^i,z_{1:k})} \\propto \omega_k^{i-1} p(z_k^i|x_k^i)p(x_k^i|x_{k-1}^i) / q(x_k^i|x_{0:k-1}^i,z_{1:k}) \tag{3-3}
\]

If \( q(x_k^i|x_{k-1}^i,z_{1:k}) = q(x_k^i|x_{k-1}^i,z_{k}) \), the weights are only related to \( x_k \) and \( z_k \). The above equation can be simplified as:

\[
\omega_k^i \propto \omega_k^{i-1} \frac{p(z_k^i|x_k^i)p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{0:k-1}^i,z_{k})} \tag{3-4}
\]

The posterior probability density of the estimated state is equal to:

\[
p(x_k|z_{1:k}) \approx \sum_{i=1}^{N_k} \omega_k^i \delta(x_k - x_k^i) \tag{3-5}
\]

where \( z_{1:k} \) is the observed value, \( x_k \) is the current state, \( x_{1:k} \) represents all values from the initial time to \( k \), and \( \delta(x_k - x_k^i) \) is a Dirac function. Defining \( f(x) = \delta(x_k - x_k^i) \), then the posterior probability density has been discretized, then the posterior probability must be used for image tracking or filtering. Then there are:

\[
E[f(x_k)] \approx \int f(x_k) \hat{p}(x_k|y_{1:k}) dx_k = \sum_{i=1}^{N_k} \int f(x_k) \delta(x_k - x_k^i) dx_k \tag{3-6}
\]

If the total number of particles \( N_k \) approaches infinity. Then Equation \((3-5)\) will be approximately equal to the true posterior probability density of the state \( x_k \). Equation \((3-5)\) also forms the basis of the particle filter algorithm.

From \((3-6)\), we can see that the average value of the sampled particle state value can directly approximate the expected value of the current state \( k \). Of course, the prerequisite for this formula is that we must know the posterior probability, otherwise we cannot sample the posterior probability. In
this case, the introduction of the importance sampling method can solve the expectation of the current state $k$ when the posterior probability is not given.

When the number of samples is large enough, it can replace the population approximately, which is the particle filter approaching any form of probability density distribution. The Rao-Blackwellized Particle Filtering (RB-PF) method factorizes the joint posterior probability distribution into a robot path part and a map part, so that the estimation of the pose of the robot and the updating of the map are mutually complementary and alternate. The mathematical model of RB-PF is:

$$p(x_{1:k}, m|z_{1:k}, u_{0:k}) = p(x_{1:k}|z_{1:k}, u_{0:k})p(m|x_{1:k}, z_{1:k})$$

(3-7)

where $x_{1:k}$ is the pose of the robot at time $k$, its position in the 2D map is $(X, Y, \theta)$; $m$ represents the global map; $z_k$ and $u_k$ are the observations and the control variables; where subscripts "k" indicates the value of the variable at the current time; and subscript 1: $k$ indicates all values of the variable from the initial time to the current time.

3.3. Indoor positioning and navigation software framework

The software architecture of the mobile robot platform for indoor localization and navigation designed in this paper is shown in figure 4. The odometer data is obtained by merging the motor encoder and the attitude sensor MPU6050. The odometer data and 2D-lidar data are transmitted to the decision through the serial port. The control layer and decision control layer adopt Intel N29 IPC. The IPC is equipped with a Linux-based ROS system. The indoor real-time positioning and navigation functions can be implemented in ROS [17].

![Figure 4. The Software architecture diagram.](image)

The positioning and navigation algorithms of mobile robots are extremely complex and require a lot of computer resources and space. Therefore, the entire ROS navigation framework can be implemented on IPCs. The overall framework for positioning and navigation of indoor mobile robots is shown in figure 5:

![Figure 5. Schematic diagram of the overall framework of positioning and navigation of mobile robots.](image)
As can be seen from figure 5, the ROS positioning and navigation framework uses distributed nodes, including robot coordinate transformation, odometry information and laser radar information as input, based on this information fusion robot pose and weight. In this map, local path planning and absolute path planning are performed on the path to achieve the purpose of positioning and navigation.

4. Indoor positioning and navigation simulation and experiment

The main task of the mobile robot designed in this paper is to complete the indoor SLAM, and the indoor SLAM requires the mobile robot to start moving from an unknown location, and can estimate the location and build the map during the movement. The simulation and physical verification experiments were carried out on the mobile robot designed in this paper.

4.1. Simulation

Building an indoor experiment environment and send motion control information through the PC keyboard to control the mobile robot to move in the indoor environment. When the robot receives the motion information and takes corresponding actions, the data of the current time odometer and the 2D-Lidar scan are acquired in real time while the motion is being performed, and the odometer node and the 2D-Lidar node are published to the ROS. SLAM calculation is performed through the move_base node to construct and update the map of the unknown environment and visualize the map. Figure 6 shows the process of building map for mobile robot. The red line shows the walking path of the mobile robot.

![Figure 6. The process of building map for mobile robot.](image)

Figure 7 shows comparison of the simulation indoor environment and map reconstructed by the robot. The red dashed box part is basically consistent with the map drawn by the right mobile robot.

![Figure 7. Comparison of the simulation indoor environment and map reconstructed by the robot.](image)
Figure 8. Comparison of the theoretical and estimated trajectory.

The estimated trajectory compared with the theoretical value is demonstrated in figure 8. The final absolute trajectory error approximately 70mm, which means the proposed method has higher accuracy on localization.

After the completion of the construction of the map, in order to verify the feasibility and effectiveness of robot navigation, specify the target position in the ROS visualization tool (Rviz) (the position shown by the small arrow on the left side of the figure), the navigation node in the ROS will pass the current The position and target position are combined with the map to calculate a reasonable global path, and according to the current motion speed and direction during the movement of the robot, the optimal speed of each motion cycle is controlled, that is, the local path plan. As shown in figure 9, the robot finally reaches the target position.

Figure 9. Global and local path planning.

4.2. Physical verification experiment
After verifying the feasibility of the algorithm through simulation, a physical verification experiment of the mobile robot is carried out. First of all, a 3m x 3m experimental environment was set up. The mobile robot was controlled by Bluetooth to continuously move in the environment. The map was constructed and updated in real time during the robot movement. As shown in figure 10, it can be seen that the map constructed by the mobile robot is basically consistent with the built environment. The black line on the right side of the figure is the obstacle boundary. The white area is the feasible area of the robot. The green path are the paths the robot has traveled. During the experiment, the robot's target position is specified, and the global_planner node draws the route to the target position according to the current position of the robot. The local_planer node makes a plan for the local path of the robot.
The robot does not collide with the obstacle during the movement and can be based on the mileage. The data of the gauge and 2D-lidar corrected the global path and the local path, and finally reached the target position smoothly.

![Figure 10. Mobile robot physical verification experiment scene.](image)

5. Conclusions
The accuracy and real-time performance of mobile robots for indoor operations, as well as the rapidity and effectiveness of construction, determine the robot's performance and application prospects. In this paper, a mobile robot platform for real-time indoor positioning and mapping is designed. Based on this platform, relevant simulations and experiments are carried out. The platform which equipped 2D laser radar and odometer as the core sensors, uses the distributed framework of open source robot operating system (ROS) to handle odometer data and lidar data, and integrates data through Rao-Blackwellized particle filter algorithm. Both simulation and experimental results verify the feasibility and practicality of the platform. Compared with most mobile robots currently in use, the mobile robots designed in this paper have autonomous positioning functions, and can simultaneously construct real-time maps in an unknown environment. It can provide technical support for robot autonomous navigation, and its technology is mature. After the functions are expanded, it can be applied to the positioning and navigation system of indoor service robots, making the indoor robot more intelligent, autonomous, and practical.

Acknowledgments
This work was supported by Research on Multi-model Similar Image Retrieval Method Based on Improved Visual Attention Model, China (No. 6150134)

References
[1] Grisetti G, Stachniss C & Burgard W 2007 Improved techniques for grid mapping with rao-blackwellized particle filters IEEE transactions on Robotics 23(1) 34-46
[2] Hu C, Hu C, He D & Gu Q 2015 A new ROS-based hybrid architecture for heterogeneous multi-robot systems Control and Decision Conference (pp.4721-4726) IEEE
[3] Ren W, He D X & Zhao J 2012 A Component Based Hybrid Embedded Framework for Mobile Robot International Conference on Parallel and Distributed Computing, Applications and Technologies Vol.23, pp.16-19 IEEE Computer Society
[4] Lawonn K, Mönch T & Preim B 2013 Streamlines for Illustrative Real - Time Rendering. In Computer Graphics Forum Vol. 32 No. 3pt3 pp. 321-330 Blackwell Publishing Ltd
[5] Schwarz J, Mankoff J & Hudson S 2011 Monte carlo methods for managing interactive state, action and feedback under uncertainty In Proceedings of the 24th annual ACM symposium on User interface software and technology pp. 235-244 ACM
[6] Liu H, Liu G & Shen Y 2006 A Novel Harmonics Detection Method Based on Wavelet Algorithm for Active Power Filter Intelligent Control and Automation 2006. WCICA 2006. The Sixth World Congress on Vol.2 pp.7617-7621 IEEE
[7] Cornejo A, Lynch A J, Fudge E, Bilstein S, Khabbazian M & McLurkin J 2013 Scale-free coordinates for multi-robot systems with bearing-only sensors The International Journal of Robotics Research 32(12) 1459-1474
[8] Demski P, Mikulski M & Koteras R 2013 Characterization of Hokuyo UTM-30LX Laser Range Finder for an Autonomous Mobile Robot Advanced Technologies for Intelligent Systems of National Border Security. Springer Berlin Heidelberg
[9] Li T, Sattar T P & Sun S 2012 Deterministic resampling: Unbiased sampling to avoid sample impoverishment in particle filters Elsevier North-Holland, Inc
[10] Tsao C L, Sanadhya S & Sivakumar R 2015 A super-aggregation strategy for multi-homed mobile hosts with heterogeneous wireless interfaces Wireless Networks 21(2) 639-658
[11] Koren Y & Borenstein J 1991 Potential field methods and their inherent limitations for mobile robot navigation. In Robotics and Automation, 1991. Proceedings. 1991 IEEE International Conference on pp. 1398-1404 IEEE
[12] Davison A J 2007 Real-time single camera slam IEEE Transactions on Pattern Analysis & Machine Intelligence 29
[13] Hao C K & Mayer N M 2013 Real-time SLAM using an RGB-D camera for mobile robots Automatic Control Conference pp.356-361 IEEE
[14] Araújo A, Portugal D, Couceiro M S & Rocha R P 2015 Integrating Arduino-based educational mobile robots in ROS Journal of Intelligent & Robotic Systems 77(2) 281-298
[15] Fioraio N & Stefano L D 2014 Slamdunk: affordable real-time rgb-d slam 8925 401-414
[16] Singhal A & Seborg D E 2010 Dynamic data rectification using the expectation maximization algorithm Aiche Journal 46(8) 1556-1565
[17] Alexander Kleiner & Christian Dornhege 2011 Mapping for the support of first responders in critical domains Journal of Intelligent & Robotic Systems 64(1) 7-31