Proceedings of the 38th Chinese Control Conference
July 27-30, 2019, Guangzhou, China

SLAM Algorithm Analysis of Mobile Robot Based on Lidar

Zhang Xuexi[1], Lu Guokun[1], Fu Genping[2], Xu Dongliang[1], Liang Shiliu[1],

1. College of Automation, Guangdong University of Technology, Guangzhou 511400, P. R. China
E-mail: zxxnet@gdut.edu.cn

2. College of Automation,Zhongkai University of Agriculture and Engineering,Guangzhou 510225, P. R. China
E-mail: pengchengniao@163.com

Abstract: In this work, we tested Simultaneous localization and mapping (SLAM) about mobile robots in indoor environment, where all experiments were conducted based on the Robot Operating System (ROS). The urban search and rescue(USAR) environment was build in the ROS simulation tool Gazebo, and our car was used to test hector SLAM in Gazebo. The rplidar A1 single-line lidar was used for 2D laser scan matching data acquisition in the practical experiments and the indoor map was built by using the open source algorithms gmapping, karto SLAM, hector SLAM software package for indoor SLAM, which can get the indoor grid maps in ROS graphical tool RVIZ. The experimental results of the three open source algorithms show that the mobile robot for simultaneous localization and mapping (SLAM) is feasible, and high-precision grid maps can be constructed.

Key Words: SLAM, lidar, hector SLAM, grid maps

1 Introduction

The intelligent mobile robot is a system that integrates communication technology, power technology, microelectronic technology and mechanical technology [1]. In recent years, the rapid development of mobile robots has attracted widespread attention. Moreover, mobile robots are also widely used in people's daily lives, such as house cleaning, intelligent express, medical services, catering services, military, intelligent transportation and entertainment. The premise that mobile robots work autonomously in these areas is that robots must be able to generate maps in the current environment. The SLAM that implements the robot must know the pose and location of the robot in the current environment. That is, the robot explores every corner of the environment and determines its position in the indoor environment and the orientation of the body according to the signs of the environment. Therefore, SLAM is a key issue in the field of mobile robots. The main content of this paper is to realize the algorithm of laser SLAM, that is, the mobile robot carries the lidar to scan the indoor environment and realize self-localization and construct indoor maps in an unknown environment. In order to measure the mapping error of each SLAM algorithm, Haiming Gao et al. [2] run gmapping, karto SLAM and graph-based SLAM based on UTM-30LX lidar to perform self-navigation in the office and corridor environment. Dong Shen et al. [3] used the Raspberry Pi as the main control to run the simple four-wheeled car for indoor SLAM, which get the result that the low-cost robot can also complete the indoor SLAM, and the effect of the map construction is not bad. Yi Cheng and Gong Ye Wang [4] ran gmapping and hector SLAM to build better grid maps in a small laboratory environment. Xieyuanli Chen et al. [5] combined the information obtained by monocular vision and single-line lidar to perform SLAM on USAR (urban search and rescue). The sensor system combined with these two sensors can not only be constructed into clear grip map, but also helps the operators control the rescue robot remotely. This can solve the problem of lidar tracking and positioning when the rescue robot climbs on the stairs and ramps. Branislav Holy [6] proposed a new scan matching algorithm that combines the registration of lines and the registration in polar coordinates. Its main idea is a new representation of the surrounding area, which makes the mentioned combination possible. The algorithm was tested by lidar on indoor venues, with real-time performance and good mapping results. Feng Zhang [7] et al. introduced probabilistic mobile robot SLAM algorithms including EKF-SLAM, UKF-SLAM, FastSLAM and UFastSLAM. Among them, EKF-SLAM and UKF-SLAM algorithms are based on Kalman filter, and FastSLAM, UFastSLAM are based on particle filter. According to the simulation experiment on Matlab, the UFastSLAM algorithm is superior to the other three methods in the estimation of the landmarks of the robot path and the nonlinear Gaussian distribution model. For nonlinear non-Gaussian motion models, the UFastSLAM algorithm is superior to FastSLAM.Although the FastSLAM algorithm can quickly and accurately perform SLAM, a grid-based FastSLAM [8] seems to be more promising. Lidar was used to detect the environment information, and during the sampling time of the lidar, the grid map was matched and updated with the new scan result, and the effect of obtained map is better than the only use of the FastSLAM algorithm. In this work [9], five commonly used laser 2D SLAM technologies are studied: hector SLAM, gmapping, karto SLAM, core SLAM and lago SLAM. Based on K-nearest neighbors, indoor mapping is compared to analyze he advantages and disadvantages of various algorithms. Finally, the CPU occupancy of various algorithms is analyzed. The 2D lidar rplidar A1 [10] is used as the main acquisition sensing in this paper, and three algorithm gmapping, hector SLAM, karto SLAM are tested in the experimental environment.

2 2D Laser SLAM

2.1 Rao-Blackwellized Particle Filter(RBPF)

In 2000, Murphy [11] applied Rao-Blackwellized particle filter algorithm for the first time to solve the SLAM problem in indoor dynamic environment. Rao-Blackwellized particle filter generally requires a large number of particles to obtain a good mapping effect, but this will increase the computa-

This work is supported by Guangdong Natural Science Foundation (No. 2016A030310237) under Grant 100000

tional complexity and increase the cpu occupancy rate of the computer. Therefore, how to reduce the number of particles is the main difficulty for scholars to optimize the particle filter algorithm. The most common algorithm in particle filter is the SIR (Samping Importance Resampling) filter. The algorithm is completed in the following four steps:

(1) Prediction stage: At first, particle filter generates a large number of samples based on state transition function prediction. These samples are called particles, whose weighted sum is used to approximate the posterior probability density.

(2) Correction stage: As the observations arrive in turn, the corresponding importance weights are calculated for each particle. This weight represents the probability that the predicted pose will be observed when the first particle is taken. In this way, the more likely the particles are observed, the higher the weight obtained.

(3) Resampling stage: Redistribute the sampled particles according to the proportion of the weights. This step is very important because of the limited number of particles in continuous distribution. In the next round of filtering, the set of resampled particle is input into the state transition equation, and new predicted particles can be obtained.

(4) Map estimation: For each sampled particle, the corresponding map estimate is calculated from the trajectory and observation of the sampling. when new observations arrive, the SIR algorithm needs to evaluate the weight of the particles from the beginning. As the length of the trajectory increases over time, the computational complexity of this process will become higher and higher. Therefore, the importance probability density function is limited to obtain a recursive formula to calculate the importance weight.

2.2 Scan Matching

The scan matching algorithm [12] mainly derives from the Iterative Closest Point (ICP [17]) algorithm and its improved algorithm. Firstly, the algorithm iterates the initial pose condition given by the odometer, and then the scan information of the 2D lidar is matched to obtain accurate and less computationally map. However, the algorithm is implemented totally based on the small deviation between the initial pose of the robot and the true pose of the robot, so as to achieve a global optimal match. The three matching methods of scan matching are:

1) Feature to Feature matching: Reduces hundreds of range points to dozens of features in the shortest running time. The use of line segments and corner features can be done indoors. In addition, there are methods to use the Feature and Feature matching to complete the urban mapping. The reflection intensity of the lidar signal and the geometric primitives can also be used to determine the features.

2) Point to Feature matching: When some features in Feature to Feature matching are not robustly detected and contain some uncertainty, the overall performance of this method to reduce the amount of information is reduced. The Points and Features matching should be a balance between speed and accuracy. The Point to Feature matching method generally uses the scanned points to match the line features, or to match the line and abstract features. The feature is a Gaussian distribution with mean and variance, and the quadratic term is calculated by the scanning point falling into the grid unit.

3) Point to Point matching: The commonly used methods in Point to Point are as following, ICP iterative closest point, IMRP iterative matching range point, and IDC iterative dual correspondence.

2.3 Graph Optimization

Common graph-based optimization algorithms are karto SLAM and Google's open source algorithm Cartographer [20]. The so-called graph optimization is to express a conventional optimization problem in the form of a graph. In graph-based SLAM, the pose of a robot is a node or a vertex, and the relationship between poses constitutes an edge. Specifically, for example, an odometry relationship between time t+1 and time t constitutes an edge, or a pose conversion matrix calculated by vision may also constitute an edge. Once the diagram is built,the pose of the robot is adjust to satisfy the constraints imposed by these edges. So the graph optimization SLAM problem can be broken down into two tasks:

1. Graph constrution. The robot pose is taken as the apex, and the relationship between the poses are taken as the edge, this step is often referred to as the front-end, which is often the accumulation of sensor information.

2. Graph optimization. The robot pose vertices are adjusted to satisfy the edge constraints as much as possible. This step is called back-end. The graph optimization process is shown in Fig 1.

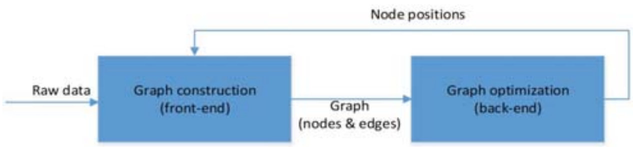


Fig. 1: The process of graph optimization

2.4 Data Association

Data association, also known as Correspondence Problem, is one of the challenges faced by SLAM. Data association refers to the process of establishing correspondence between various sensor measurements obtained at different times and different locations between sensor measurements and map features, or between map features to determine whether they originate from the same physical entity in the environment. The correctness of the data association is critical to the state estimation in the SLAM problem.

3 Analysis of Laser SLAM Algorithms

3.1 Gmapping

The open source gmapping algorithm is based entirely on particle filter algorithms, and gmapping is based on the papers of Giorgio Grisetti, Cyrill Stachniss and Wolfram Burgard [12, 13]. Positioning requires mapping, and mapping needs to be positioned first, which makes the SLAM problem difficult. Therefore, RBPF is introduced to solve the SLAM problem, that is, to position first and rec onstruct the map after. Gmapping is the open source SLAM software package in ROS in 2007. It is the most widely used software package for 2D lidar. Doucet A et al. proposed an improved importance probability density function based on particle filter algorithm and added adaptive resampling technique. To reduce the number of resampling steps, Doucet A proposed a theoretical decision method to determine whether resampling is required or not. Resampling is performed only when the number of particles drops to a given threshold, which reduces the number of samples and slows down particle degradation. Since resampling is only performed when needed, the number of resamplings in the entire mapping algorithm will be greatly reduced. Multiple indoor SLAM experiments have demonstrated that this approach greatly reduces the risk of filtering good particles. The gmapping algorithm can be used for indoor and outdoor positioning and mapping, and the improved adaptive RBPF algorithm is used for positioning and mapping. The gmapping algorithm not only requires 2D lidar information, but also relies heavily on the odometer information to reduce the number of particles. However, for some uneven ground conditions and some slopes, the mapping effect shows that the gmapping algorithm is not satisfactory. As the number of particles required for an increase in the scene increases, because each particle carries a map, the amount of memory and computation required to build a large map increases. Therefore, it is not suitable for building a large scene map. And there is no loop closure detection, so the map may be misaligned when the loop is closed, although increasing the number of particles can make the map closed but at the expense of increased computation and memory.

3.2 Hector SLAM

Hector SLAM is one of many scan matching algorithms. Hector SLAM [15] was proposed by Kohlbrecher, von Stryk, Meyer and Klingauf at the Technical University of Darmstadt (Kohlbrecher et al., 2013). The Hector SLAM algorithm utilizes the high sampling frequency of modern two-dimensional lidar (eg, the Hokuyo UTM-30LX has a scan frequency of 40hz) and provides a 2D pose estimate at the sensor's scan rate. Although the system does not provide explicit loop closure detection, it is accurate enough for many real-world scenarios. Scan matching SLAM is an algorithm that estimates the translation and rotation of the robot between two scans based on nearest neighbor scan matching. The final map generated by hector SLAM is also the same as the gmapping. The laser beam lattice is optimized by using the obtained map to represent the laser point on the map, and the probability of occupying grid are estimated. The Gaussian-Newton method is used to solve the scan matching problem, and the set of laser point is mapped to the rigid body transformation(x, y, theta) of the existing map. In order to avoid local minimum rather than global optimality by using multi-resolution maps. State estimation in navigation is added to the inertial measurement system(IMU) by using EKF filtering. If a 2D lidar with a high sampling frequency is used, the Hector SLAM has a significant mapping effect in a small indoor environment, and does not need to stay in some complicated geographical environment. Because the algorithm does not need to collect odometer data, it is very suitable for some robots with limited computing.

3.3 Karto SLAM

Karto SLAM [16] is a graph optimization method that uses a highly optimized and non-iterative cholesky matrix for sparse system decoupling as a solution. The mean of the graph is used to represent the map, and each node represents a location point of the robot trajectory and the data set of sensor measurement. The arrow pointing to the connection represents the movement of the continuous robot position point. When new node joins, the map will be calculated and updated according to the constraint of the node arrow in the space. The ROS version of karto SLAM, which uses the Spare Pose Adjustment (SPA), is related to scan matching and loop closure detection. The more landmarks, the greater the memory requirements. However, the graph optimization method has a greater advantage than other methods in the large environment. In some cases, karto SLAM is more effective because it only contains a point graph (robot pose). The framework of karto SLAM algorithm is shown in Fig 2.

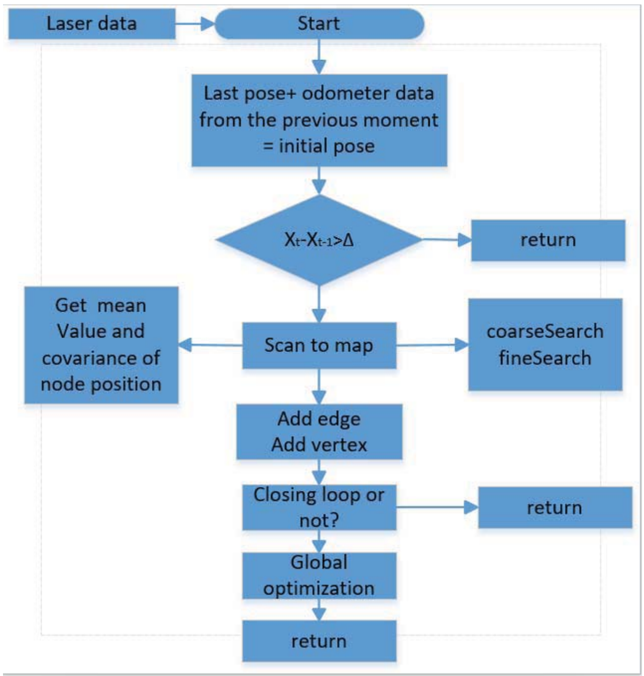


Fig. 2: Framework of karto SLAM algorithm

Fig. 3: Robot platform

4 Introduction of Robot Platform and Laser Slam System

4.1 Robot Platform

Our mobile robot is a crawler vehicle based on two-wheel differential drive motion model. Although the crawler struc-

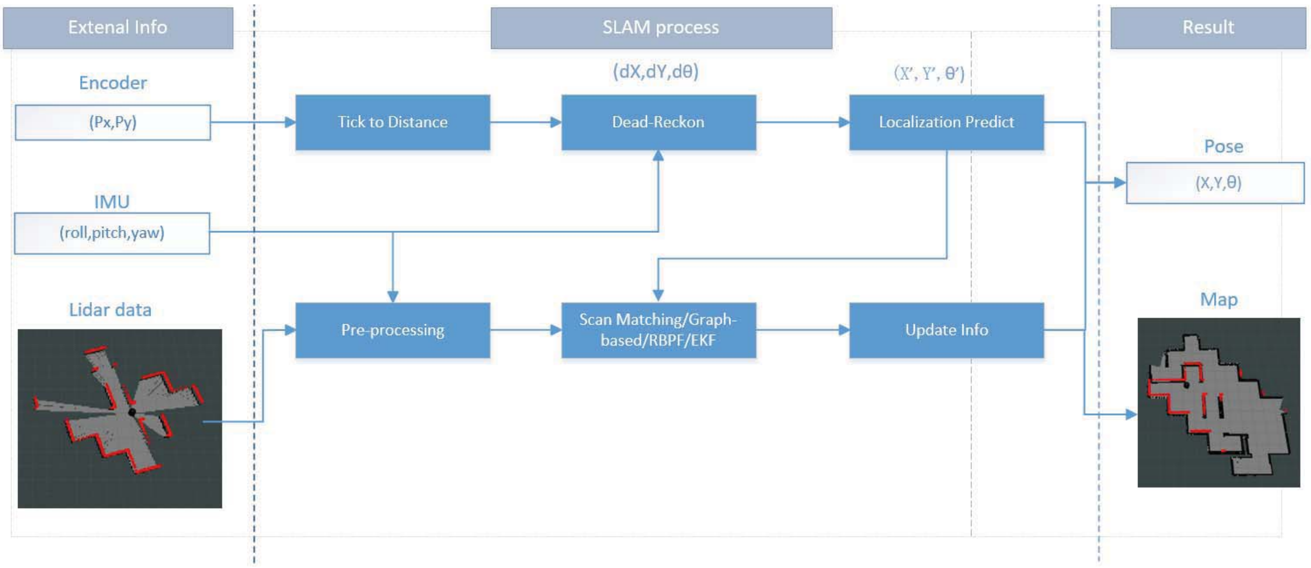


Fig. 4: Framework of laser SLAM system

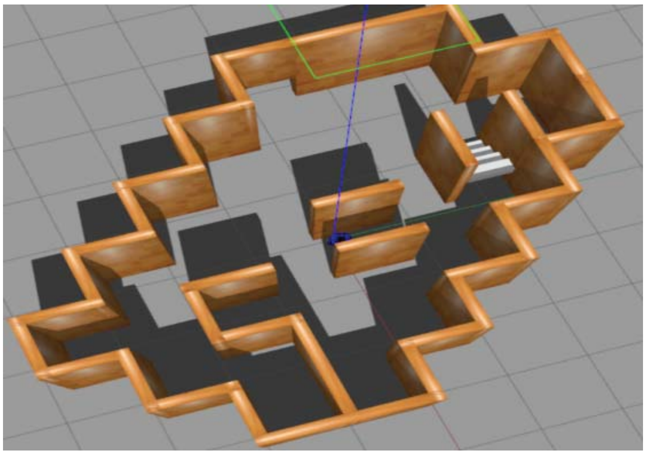


Fig. 5: USAR environment

ture is slower than the wheel structure, the crawler car can rotate in place and perform better on uneven roads with a strong stability. The robot platform is equipped with S-LAMTEC's 2D lidar rplidar A1. The fastest sampling frequency of the this lidar is 10hz and the scanning radius is 6m. All our algorithm experiments are based on Robot Operating System (ROS) platform. Because ROS system can be applied to different robots and sensors, it is widely used in the field of robots. The ROS Kinetic of Ubuntu 16.04 is used on the microcomputers. Robot Platform is shown in the Fig 3.

4.2 Introduction of Laser SLAM System Framework

The typical hardware components of the laser SLAM system are mainly lidar, odometry, motion control unit, IMU and so on. The simplest mobile robot at least contains one ranging unit. The process includes preprocessing, track estimation, self-positioning, SLAM update, etc., as shown in Fig 4. There are multiple implementations for each of these sections. The object information collected by the lidar presents a series of scattered points with accurate angle and distance information, which is called a point cloud. Generally, the

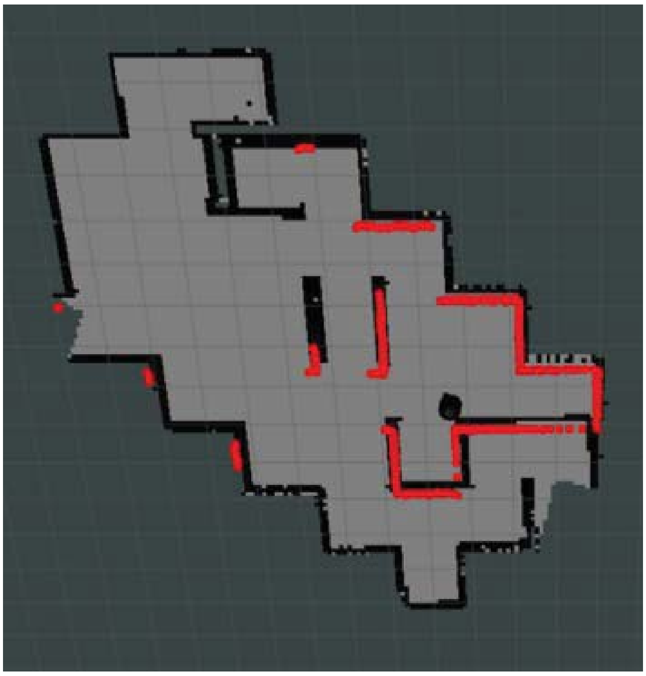


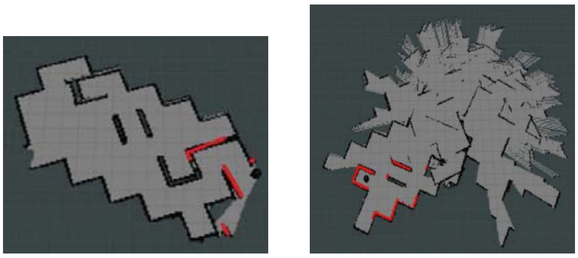
Fig. 6: Gmapping gazebo simulation results

laser SLAM system uses different algorithms to calculate the distance and attitude of the relative motion of the lidar by matching and comparing two point clouds at different times, thus completing the self-positioning.

5 Results and Discussions

5.1 Simulation Experiment

We use ROS simulation tool Gazebo to draw the competition environment of the USAR, which is a maze of 10m*6m, as shown in Fig 5. We first establish the URDF model of our car, using the Rviz tool and running the gmapping to simulate the results in the match environment as shown in Fig 6. We also used the hector SLAM algorithm to further simulate the SLAM. When we set the speed of the car to 0.2 m/s and the angular velocity to 1 rad/s, the experimental results are shown in Fig. 7(a). However, once we set the speed of the car to more than 0.2 m/s and the angular velocity to more than 1 rad/s, and the experimental results are shown in Fig. 7(b).



(a) lower speed (b) higher speed

Fig. 7: Simulation results of the hector SLAM algorithm at different speeds

It can be seen from Fig. 7(b) that when the speed of the car is greater than 0.2m/s, the mapping effect of hector SLAM is not satisfactory, because hector SLAM does not use the data of the odometer in the robot. So when the the car spins too fast, it does not know it is rotating. At this time, the laser data collected is not matched with other parts in the map, resulting in 'ghosting'. Hector SLAM only uses 2D lidar data, and there is no obvious choice for loop closure detection. This is why the hector SLAM results are less accurate.

5.2 Real World Tests

The experiments in our paper were carried out in the same laboratory environment. The environment has a variety of different structures, including slopes, stools, tables, walls, columns, etc. Due to the different placement of each stool, the complexity of the environment increases. The experimental environment is a closed environment of 16m*10m, and all experiments of the SLAM algorithm are performed in the same environment, and the laboratory environment is as shown in Fig 8. We implemented three different SLAM algorithms by using the robot shown in Figure 3. In order to better implement the algorithm, our robot is equipped with rplidar A1, imu sensor and odometry to provide more information feedback. The actual implementation of the gmapping in the experimental environment is shown in Fig 9. Fig 10 shows the map taken from the hector SLAM under the experimental environment. In the Experimental environment, the map constructed by using the karto SLAM algorithm is shown in Fig 11.

Fig. 8: Experimental environment

Since we are using the rplidar A1, the sampling frequency is much lower than other lidar on the market. All physical tests in the experimental environment require the robot to stay in some complicated environments for a while waiting for the lidar to collect more infomation, so that the collected data can be better looped. It can be seen from the figure that if the robot is equipped with the imu and the odometer, the accuracy of the maps constructed by the three algorithms is

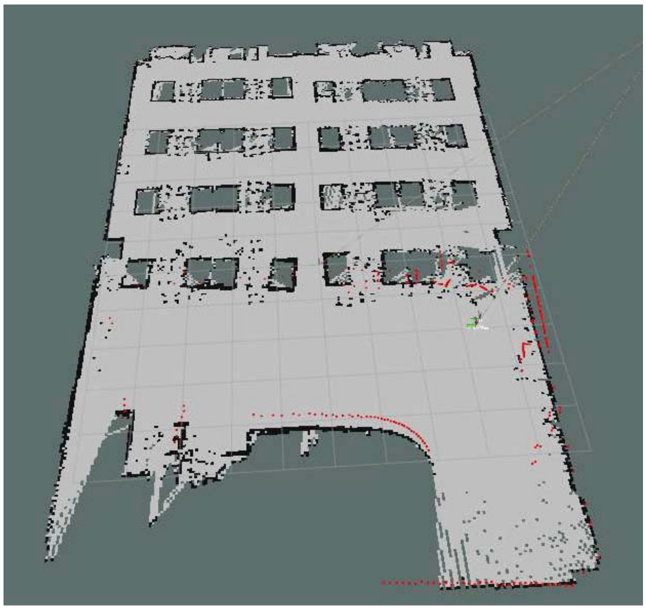


Fig. 9: Mapping result of gmapping

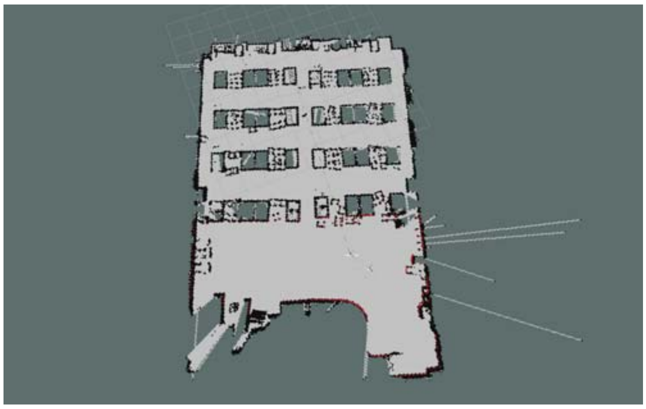


Fig. 10: Mapping result of hector SLAM

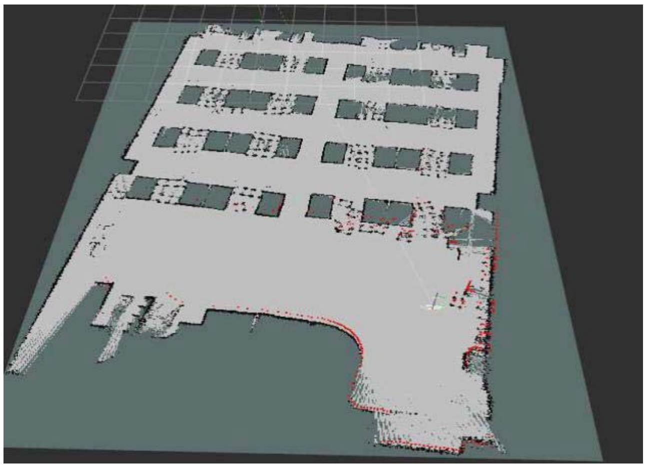


Fig. 11: Mapping result of karto SLAM

relatively high, and the position of each stool can be accurately marked.

6 Conclusion

In this paper, we first analyze the principle of the commonly used algorithms gmapping, hector SLAM, karto SLAM, and then we use a crawler-based robot to realize these three in the same experimental environment. In the future, the robot system will be able to explore and build a complete geotiff map in USAR. Moreover, we will build a 6-axis robot arm and a camera platform on the robot to complete other projects. In future, we will investigate the routing methods in the transportation filed, such as route optimization for a single independent vehicle [22–26], and multiple cooperative vehicles [27–30], which can be used to help solving the SLAM problem. Moreover, we will also consider the criterion of 'faster path' [31] for solving the SLAM problem.

References

[1] Wei Chen, Tingbo Liao, Zhihang Li, Haozhi Lin, Hong Xue, Le Zhang, Jing Guo, and Zhiguang Cao, Using FTOC to Track Shuttlecock for the Badminton Robot, *Neurocomputing*, 2019, 334: 182-196.

[2] Haiming Gao, Xuebo Zhang, Jian Wen, Jing Yuan, Yongchun Fang. Autonomous Indoor Exploration Via Polygon Map Construction and Graph-Based SLAM Using Directional Endpoint Features[J]. IEEE Transactions on Automation Science and Engineering (Early Access), pp.1-12, December 2018.

[3] Dong Shen, Yakun Huang, Yangxi Wang, Chaoyang Zhao. Research and Implementation of SLAM Based on LIDAR for Four-Wheeled Mobile Robot[J]. 2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE),August 2018.

[4] Yi Cheng, Gong Ye Wang. Mobile Robot Navigation Based on Lidar[J]. 2018 Chinese Control And Decision Conference (CCDC), July 2018.

[5] Xieyuanli Chen, Hui Zhang, Huimin Lu, Junhao Xiao, Qihang Qiu, Yi Li. Robust SLAM system based on monocular vision and LiDAR for robotic urban search and rescue[J]. 2017 IEEE International Symposium on Safety,Security and Rescue Robotics (SSRR),October 2017.

[6] Branislav Holy. Registration of lines in 2D LIDAR scans via functions of angles, Engineering Applications of Artificial Intelligence[J]. Volume 67, Pages 436-442,January 2018.

[7] Feng Zhang, Siqi Li, Shuai Yuan, Enze Sun, Languang Zhao. Algorithms analysis of mobile robot SLAM based on Kalman and particle filter[J]. 2017 9th International Conference on Modelling,Identification and Control (ICMIC),March 2018.

[8] Fumitaka Hashikawa, Kazuyuki Morioka. Mobile robot navigation based on interactive SLAM with an intelligent space[J]. 2011 8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI),February 2012

[9] J. M. Santos, D. Portugal, and R. P. Rocha. An evaluation of 2d slam techniques available in robot operating system[J]. in Safety,Security,and Rescue Robotics (SSRR),2013 IEEE International Symposium on, pp. 1–6, IEEE, 2013.

[10] Xiaobin Xu, Minzhou Luo, Zhiying Tan, Min Zhang, Hao Yang. Plane segmentation and fitting method of point clouds based on improved density clustering algorithm for laser radar,Infrared Physics and Technology[J]. Volume 96,Pages 133-140, January 2019.

[11] Murphy K, Russell S. Rao-Blackwellized particle filtering for dynamic Bayesian networks[M]. New York: Springer-Verlag, 2001: 587-633.

[12] Giorgio Grisetti, Cyrill Stachniss, Wolfram Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters[J]. IEEE Transactions on Robotics (ITRO), vol. 23, pp. 34-46, 2007.

[13] Giorgio Grisetti, Cyrill Stachniss, Wolfram Burgard. Improving grid-based SLAM with Rao-Blackwellized Particle Filters by adaptive proposals and selective resampling[J]. IEEE International Conference on Robotics and Automation(ICRA 2005),2005.

[14] Doucet A.Godsill S,Andrieu c. On sequential Monte Carlosampling methods for Bayesian filtering[J]. Statistics and Com-puting,2000(10)1197–208.

[15] Stefan Kohlbrecher, Oskar von Stryk, Johannes Meyer, Uwe Klingauf. A flexible and scalable SLAM system with full 3D motion estimation[J]. 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics,December 2011.

[16] Kurt Konoli, Giorgio Grisetti, Rainer Kummerle, Benson Limketkai, Regis Vincent. Efficient Sparse Pose Adjustment for 2D mapping[J]. 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, December 2010.

[17] F.A. Donoso, K.J. Austin, P.R. McAree. Three new Iterative Closest Point variant-methods that improve scan matching for surface mining terrain[J]. Robotics and Autonomous Systems, Volume 95, Pages 117-128, September 2017.

[18] Pavlo Denysyuk, Vasyl Teslyuk, Iryna Chorna. Development of mobile robot using LIDAR technology based on Arduino controller[J]. 2018 XIV-th International Conference on Perspective Technologies and Methods in MEMS Design (MEMSTECH), May 2018.

[19] Wolfgang Hess, Damon Kohler, Holger Rapp, Daniel Andor. Real-time loop closure in 2D LIDAR SLAM[J]. 2016 IEEE International Conference on Robotics and Automation (ICRA),June 2016.

[20] Brian Hampton, Akram Al-Hourani, Branko Ristic, Bill Moran. RFS-SLAM robot: An experimental platform for RFS based occupancy-grid SLAM[J]. 2017 20th International Conference on Information Fusion (Fusion), August, 2017.

[21] Rauf Yagfarov, Mikhail Ivanou, Ilya Afanasyev. Map Comparison of Lidar-based 2D SLAM Algorithms Using Precise Ground Truth[J]. 2018 15th International Conference on Control, Automation,Robotics and Vision (ICARCV), December, 2018.

[22] Zhiguang Cao, Hongliang Guo, Jie Zhang, Frans Oliehoek, Ulrich Fastenrath. Maximizing the Probability of Arriving on Time: A Practical Q-Learning Method [C]. 31th AAAI Conference on Artificial Intelligence, 2017: 4481-4487.

[23] Zhiguang Cao, Hongliang Guo, Jie Zhang, Dusit Niyato, Ulrich Fastenrath. A Data-Driven Method for Stochastic Shortest Path Problem [C]. 17th IEEE International Conference on Intelligent Transportation Systems (ITSc), 2014: 1045-1052.

[24] Zhiguang Cao, Hongliang Guo, Jie Zhang, Dusit Niyato, Ulrich Fastenrath. Improving the Efficiency of Stochastic Vehicle Routing: A Partial Lagrange Multiplier Method [J]. IEEE Transactions on Vehicular Technology, 2016, 65(6): 3993-4005.

[25] Zhiguang Cao, Hongliang Guo, Jie Zhang, Oliehoek Frans, Fastenrath Ulrich. Finding the Shortest Path in Stochastic Vehicle Routing: A Cardinality Minimization Approach [J]. IEEE Transactions on Intelligent Transportation Systems, 2016, 17(6):1688-1702.

[26] Zhiguang Cao, Yaoxin Wu, Akshay Rao, Felix Klanner, Stefan Erschen, Wei Chen, Le Zhang, Hongliang Guo. An Accurate Solution To The Cardinality-based Punctuality Problem [J]. IEEE Intelligent Transportation Systems Magazine, 2018.

[27] Zhiguang Cao, Siwei Jiang, Jie Zhang, Hongliang Guo. A Unified Framework for Vehicle Rerouting and Traffic Light Con-

trol to Reduce Traffic Congestion [J]. IEEE Transactions on Intelligent Transportation Systems, 2017, 18(7):1958-1973.

[28] Zhiguang Cao, Hongliang Guo, Jie Zhang. A Multiagent-Based Approach for Vehicle Routing by Considering Both Arriving on Time and Total Travel Time [J]. ACM Transactions on Intelligent Systems and Technology, 2018, 18(3): Article No. 25.

[29] Hongliang Guo, Zhiguang Cao, Madhavan Seshadri, Jie Zhang, Dusit Niyato, Ulrich Fastenrath. Routing Multiple Vehicles Cooperatively: Minimizing Road Network Breakdown Probability [J]. IEEE Transactions on Emerging Topics in Computational Intelligence, 2017, 1(2):112-124.

[30] Zhiguang Cao, Hongliang Guo, Jie Zhang, Ulrich Fastenrath. Multiagent-based Route Guidance for Increasing the Chance of Arrival on Time [C]. 30th AAAI Conference on Artificial Intelligence, 2016: 3814-3820.

[31] Jing Guo, Yaoxin Wu, Xuexi Zhang, Le Zhang, Wei Chen, Zhiguang Cao, Lu Zhang, Hongliang Guo. Finding the 'Faster' Path in Vehicle Routing [J]. IET Intelligent Transport Systems, 2017, 11(10): 685-694.