Indoor navigation system using the Fetch robot*

Huishen Zhu[1], Brenton Leighton[2], Yongbo Chen[2], Xijun Ke[1], Songtao Liu[1], and Liang Zhao[2]

[1] Shanghai Institute of Micro Motor (The 21st Research Institute of China Electronics Technology Group Corporation), SIMM(CETC21),
[2] Centre for Autonomous Systems, University of Technology Sydney, Sydney NSW 2007, Australia
`Yongbo.Chen@student.uts.edu.au`

Abstract. In this paper, we present a navigation system, including off-line mapping and on-line localization, for the Fetch robot in an indoor environment using Cartographer. This framework aims to build a practical, robust, and accurate Robot Operating System (ROS) package for the Fetch robot. Firstly, using Cartographer and the fusion of data from a laser scan and RGB-D camera, a two-dimensional (2D) off-line map is built. Then, the Adaptive Monte Carlo Localization (AMCL) ROS package is used to perform on-line localization. We use a simulation to validate this method of mapping and localization, then demonstrate our method live on the Fetch robot. A video about the simulation and experiment is shown in https://youtu.be/oOvxTOowe34.

Keywords: data fusion, ROS framework, Fetch robot, mapping, localization

1 Introduction

The aim of indoor navigation is to provide a robot its location with respect to a map of an indoor environment. Simultaneous Localization and Mapping (SLAM) is commonly used to generate the map on an unknown environment [1]. Broadly speaking, the process of solving the SLAM problem can be divided into two parts; the front-end and the back-end. The front-end involves processing input sensor data, while the back-end is an optimization process to create a consistent map. The design of the front-end varies based on the sensor type, for example laser scan [2] or vision [3]. For laser scan based SLAM, the system may create an occupancy grid [4] or a point cloud using Iterative closest point (ICP) [5]. This paper aims to obtain the occupancy grid map.

This paper focuses on the navigation system of Fetch robot based on the fusion of the 2D laser sensor and the RGB-D camera sensor. This work provides mapping and localization for path planning and following, which is the basis of a larger application involving the Fetch robot. The basic algorithms of this

* Supported by Shanghai Institute of Micro Motor (SIMM) project: "Develop the SLAM Algorithm for Fetch Robot in the Indoor Office and Corridor Environment"

paper are the Cartographer [2] SLAM system and the Adaptive Monte Carlo Localization (AMCL) Robot Operating System (ROS) package. Cartographer is a system that provides real-time SLAM in 2D and 3D across multiple platforms and sensor configurations. It is the state of the art technology in the field of laser scan based SLAM, and performs well in some challenging environments. AMCL is a 2D probabilistic localization system for a mobile robot, which implements the KLD-sampling Monte Carlo localization approach. A fused laser scan, generated from laser scan and point cloud data, is provided to Cartographer to create the occupancy grid map, which AMCL uses for localization of the Fetch robot.

The main structure of this paper is shown as follows: In Section 3 we describe the problem and offer a practical solution framework. In Section 4 we describe the implementation, including data collection, data pre-processing, off-line mapping and on-line localization. In Section 5 we describe the simulations and experiments performed to verify the correctness and practicality of the proposed method. Finally, in Section 6, we make a brief conclusion of this paper.

2 Related work

Early SLAM techniques primarily focused on the filter-based method, including Extended Kalman Filter(EKF) [6], Particle filter(PF)[7], and so on. In [8], Christopher Weyers and Gilbert Peterson presents a SLAM solution combining stereo cameras, inertial measurements, and vehicle odometry into a Multiple Integrated Navigation Sensor (MINS) path based on the Kalman Filter(KF) and FastSLAM algorithm, which operates a Rao-Blackwellized Particle Filter (RBPF). Jinglin Zhang, Yongsheng Ou, et al. apply the RBPF method in the SLAM task of a restaurant service robot using a depth camera and a move base with odometry and gyro [9].

It is well-known that SLAM technology develops from the filter-based method to the optimization-based method. Because of the better consistency and efficiency, the optimization-based method, especially for the pose-graph SLAM framework [10], is widely used in the indoor navigation method of the robot system. Yu-Cheol Lee and Seung-Hwan Park present a mobile mapping system (MMS) whose front-end is the map-based scan matching and back-end is the graph-based optimization [11]. The pose-graph optimization method is a very general and efficient method. Cartographer is a popular system for 2D laser scan SLAM, and is also based on this technique.

It is easy to know that a single sensor can not offer the platform a accurate and robust solution. So the fusion between laser and camera is one of the popular solutions. Zerong Su, Xuefeng Zhou et al. propose a reliable global localization approach with the capability of addressing the kidnapped robot problem, where both laser and camera sensors are used based on the pose graph optimization [12]. Based on the ICP and pose-graph optimization, Adrian Ratter and Claude Sammut present a novel algorithm for fusing Laser Rangefinder and RGB-D data to accurately track the position of a robot and generate large 3D maps [13].

3 Problem description and solution Framework

Assume that the robot is performing some tasks in an unknown environment such as coverage, exploration or search. Our navigation framework is used to build a highly accurate and practical 2D map for the surrounding environment, for the purpose of localization. This navigation framework will offer the real-time robot poses for the Fetch robot [14] and help it to finish tasks such as path planning and control.

The solution framework can be divided into two main parts: the off-line part and the on-line part. In the on-line part, we capture data and fuse the laser scan with the point cloud from the RGB-D camera. In the off-line part we use the generated data to create a map of the environment. Once the map is generated we can then localize on-line. The basic solution framework is summarized and presented in Fig. 1. This framework is written as a ROS package [15] and applied in the real robot system.

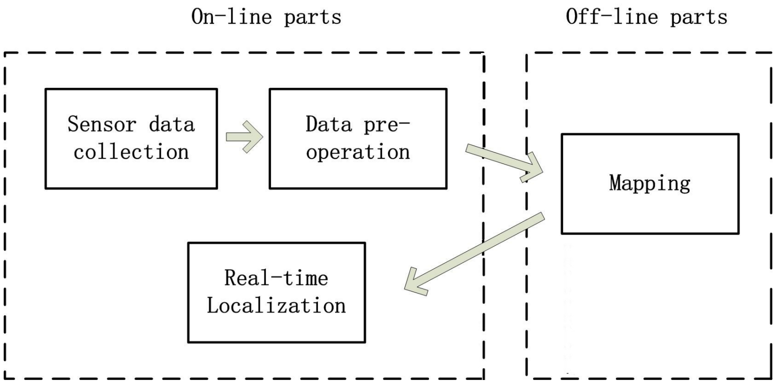

Fig. 1: Solution framework

4 Implementation

In this section we present the methodology and the implementation details of the system.

4.1 Sensors and data collection

Our ROS package is built for the Fetch robot. There are multiple kinds of sensors in Fetch robot, including a SICK TIM571 laser, a 6-axis inertial measurement unit (IMU), a Primesense Carmine 1.09 short-range RGBD sensor and gripper sensors (Fig. 2) [14]. For our application, the laser scanner and head camera are the most useful data sources on the Fetch robot. These sensors publish to the ROS topics ‘head_camera/depth_ downsampled/points’ and ‘base_scan’. The Fetch robot publishes many useful ROS topics and we can easily subscribe them.

(a) SICK TIM571 laser

(b) Primesense Carmine 1.09 short range RGB-D camera

Fig. 2: Important sensors on Fetch robot

4.2 Laser scan and RGB-D point cloud data fusion

Prior to the mapping process, it is necessary to fuse the laser scan data and RGB-D camera point cloud data to create the map. Our pre-processing involves transforming the point cloud from the RGB-D camera into the frame of the laser scanner, removing points that belong to the floor, converting the point cloud into a laser scan, then combing the generated laser scan with the actual laser scan.

The ROS topic ‘head_camera/depth_ downsampled/points’ is a 3D point cloud, and may include points belonging to the floor. In order to avoid the effect of these floor points we remove any points lower than the level of the laser scanner. This is achieved by transforming the points into the laser scanner frame using the ROS tf2 package, filtering any points with a Z axis values less than zero using Point Cloud Library (PCL), then transforming the point cloud back into the RGB-D camera frame for publishing.

With the floor removed from the point cloud, the data is flattened into a pseudo laser scan, so that is can be combined with the real laser scan. The pseudo laser scan is created with the same field of view and step angle as the real laser scan, and the two are combined by iterating through both and saving the shorter range value into the combined laser scan data. Fig. 3 and Fig. 4 show this process. The fused data is published as the ROS topic ‘base_scan_combined’, and is used by Cartographer for creating a map.

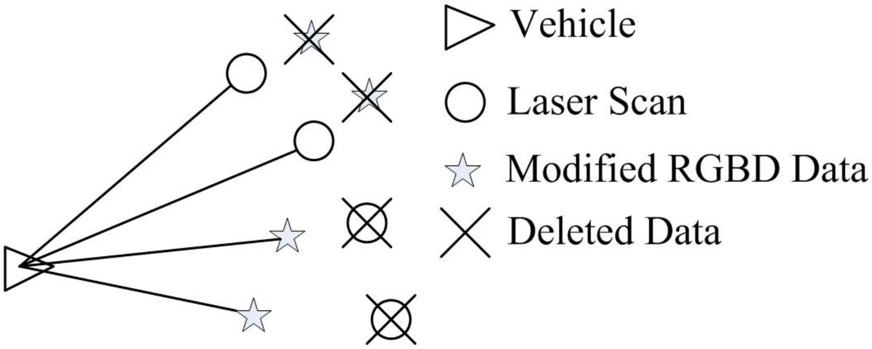

Fig. 3: Data fusion

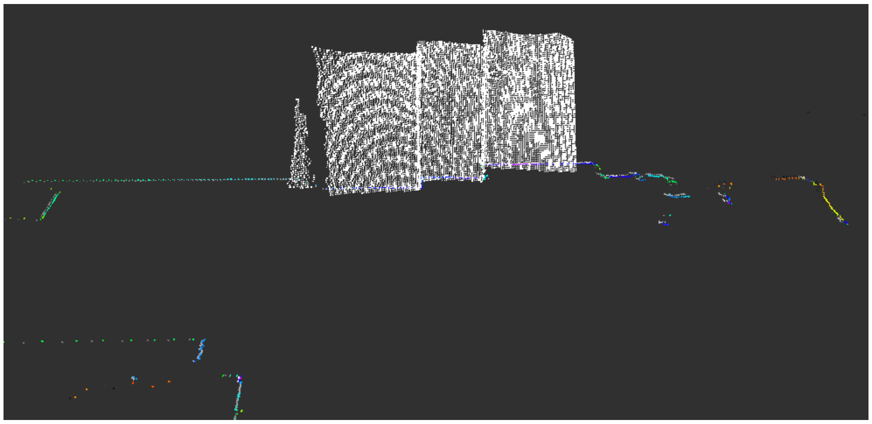

Fig. 4: Data fusion in real dataset (White wall points are the camera data, colourful points are laser data and the white horizon points are fused data)

4.3 Mapping using Cartographer

We save a ROS bag of the fused data and apply Cartographer off-line to create the occupancy grid map. Cartographer's mapping node outputs a protocol buffer stream file, which can be used in various ways.

Using the fused laser scan presents an issue for mapping, because the ranges and fields of view of the two sensors are significantly different The RGB-D camera has a maximum range of 3 m and horizontal field of view of 54 °, while the laser scanner has a range of up to 25 m and a field of view of 220 °. During the mapping process, obstacles that appear only in the point cloud are quickly erased once they are out of range of the RGB-D camera. To keep the point cloud obstacles we set the ”insert_free_space” to false when generating the obstacle grid from the protocol buffer stream file. This results in a map containing all obstacles but no unoccupied cells, which are needed for localization and path planning. To solve this problem we generate another map with ”insert_free_space” set to true, and combine the two maps with MATLAB, to produce a map that contains both free space and all obstacles.

4.4 On-line localization using AMCL

As the final step in our method, localization is performed on-line using the AMCL [16] package. It is noted that, because our map is generated from the fused data of the laser scanner and RGB-D camera, our AMCL node needs to subscribe the ‘base_scan_combined’ topic instead of the laser scan only topic.

5 Simulations and Experiments

In order to verify the correctness and practicality of our proposed method, we present some simulations and experiments.

5.1 Simulations

In this part, we build two simulations in the ROS Gazebo simulation environment and perform the mapping and localization process based on these two environments.

Simulation 1 This environment is an indoor office environment with some tables and chairs. The simulated Fetch moves around this office, collecting sufficient laser and point cloud data to build a complete map. The simulated Fetch robot and surrounding environment are shown in Fig. 5 and 6.

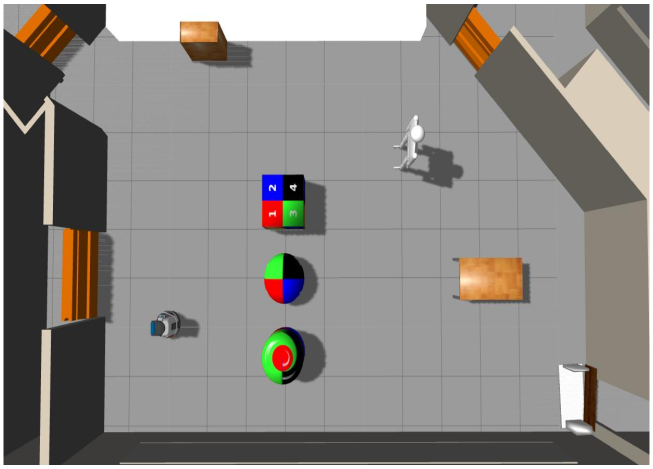

Fig. 5: Simulation environment 1

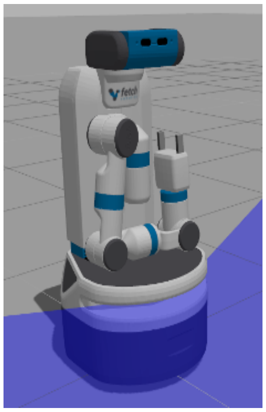

Fig. 6: Fetch robot simulator

The results including the built map and localization are presented in Fig.7. The relative coordinate errors are presented in Fig.8.

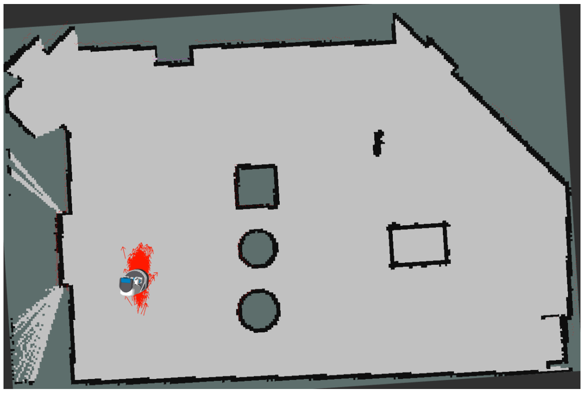

Fig. 7: Mapping and localization results of Simulation 1

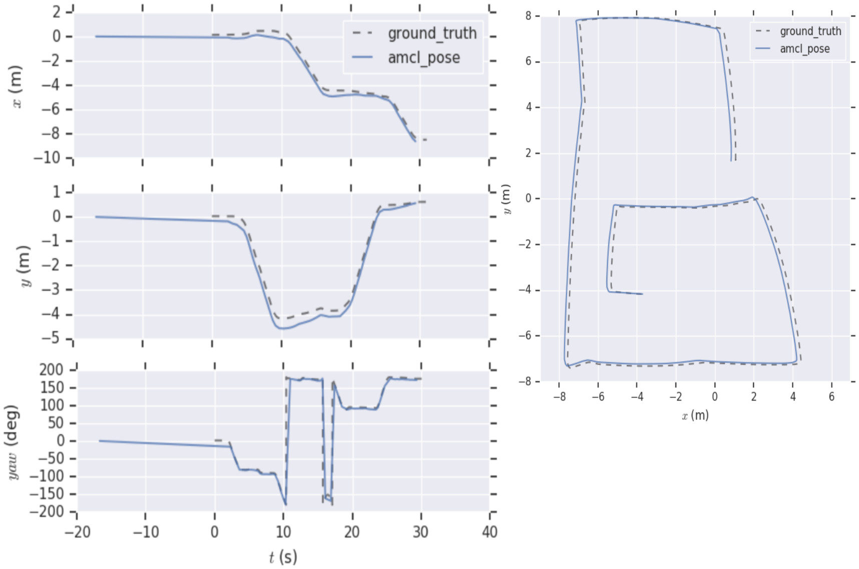

Fig. 8: Relative errors between the coordinates (Left) and trajectories (Right) of ground truth and AMCL result of Simulation 1

In Fig.8, it is easy to find that localization errors are small, which verifies the accuracy of our proposed method.

Simulation 2 Similar to the previous simulation, we perform our method in an indoor office environment with some tables and produce the map and localization result in Fig.9. We also output the relative error of the localization coordinate and trajectory, shown in Fig.9.

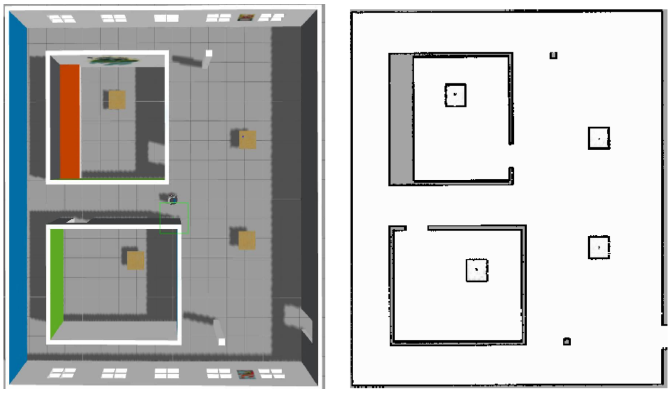

Fig. 9: Simulation environment 2 and built map

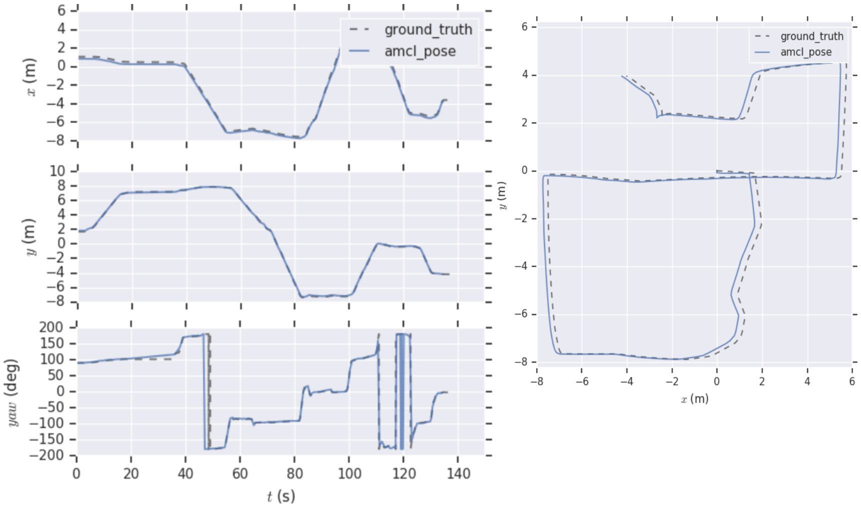

Fig. 10: Relative errors between the coordinates (Left) and trajectories (Right) of ground truth and AMCL result of Simulation 2

5.2 Experimental results in an indoor environment

For the on-line experiment, we use the real Fetch robot to perform the complete process.

Scenario 1 The first on-line scenario is a corridor environment of UTS Tech Lab. We combine two maps obtained by cartographer and get the final map with some obstacles, like chairs and tables, shown in Fig.11. The surrounding environment from on-board camera, built map and localization results are shown in Fig.12.

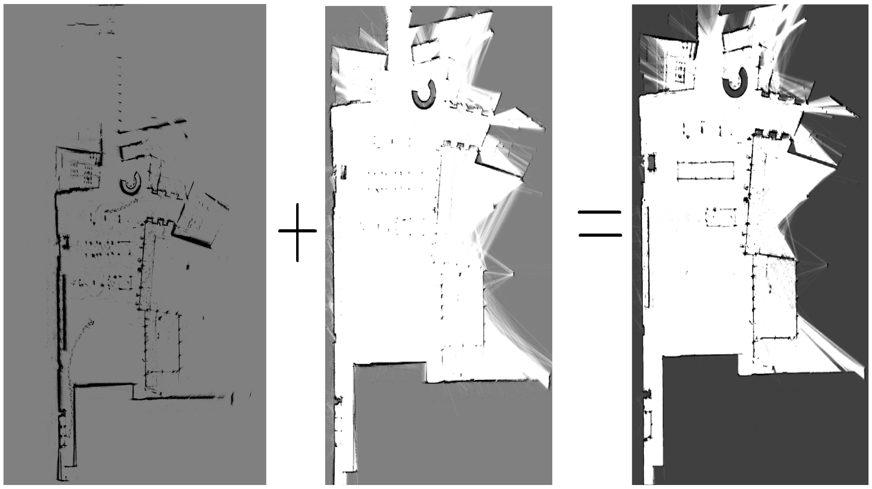

Fig. 11: Built maps

Scenario 2 The second scenario is in an experimental room of UTS Tech Lab. Using a similar technique, we can get the following map and localization result and the corresponding camera image from Fetch robot, shown in Fig. 13.

Based on these results, we can find that our method can be successfully used in the practical environment.

6 Conclusion and Future work

This paper presents a practical ROS system for mapping and localization on the Fetch robot. We first collect the laser scan and point cloud data, and fuse them using them into a combined laser scan. Based on the fused data and Cartographer, two maps are created that are combined and used for on-line localization with AMCL. The majority of this process is written by C++ and tested both in

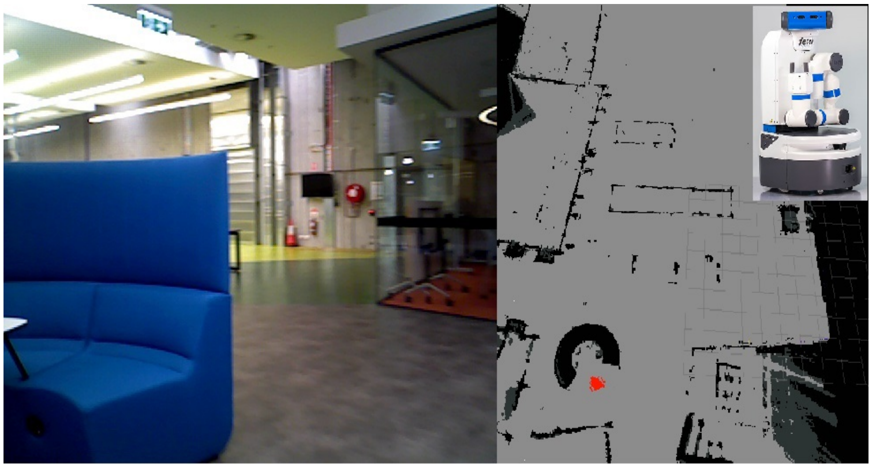

Fig. 12: Built map, localization results and corresponding camera image of the on-line experiment 1

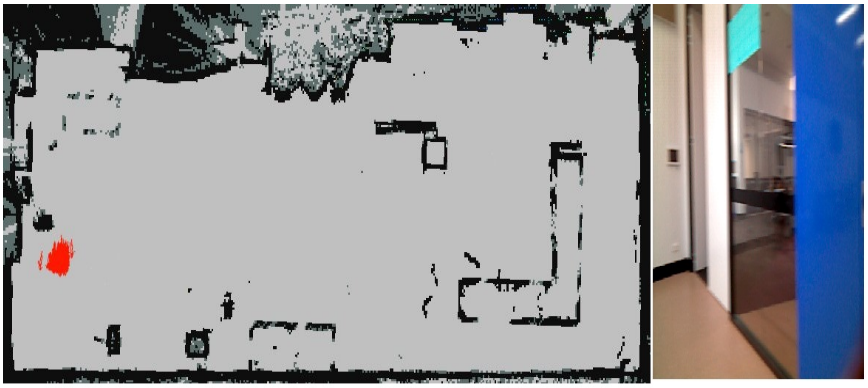

Fig. 13: Built map, localization results and corresponding camera image of the on-line experiment 2

the Gazebo simulation and on the real Fetch robot. The simulation and experiment results prove that our solution framework is efficient and practical.

This is just the first step of our research on the navigation system of the Fetch robot. In the future work we will introduce the machine learning based computer vision technology and the sub-map localization method into this navigation system.

References

1. C. Cadena and L. Carlone and H. Carrillo and Y. Latif and D. Scaramuzza and J. Neira and I. Reid and J.J. Leonard, Past, Present, and Future of Simultaneous Localization And Mapping: Towards the Robust-Perception Age, in IEEE Transactions on Robotics, 32(6), pp. 1309-1332, 2016.
2. W. Hess, K. Damon, R. Holger, and A. Daniel, Real-time loop closure in 2D LIDAR SLAM. In 2016 IEEE International Conference on Robotics and Automation (ICRA), pp. 1271-1278. IEEE, 2016.
3. R. Mur-Artal, and J.D. Tards, Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. IEEE Transactions on Robotics, 33(5), pp.1255-1262, 2017.
4. J. Strom, and O. Edwin, Occupancy grid rasterization in large environments for teams of robots. In 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4271-4276. IEEE, 2011.
5. A real-time 2D and 3D ICP-based SLAM system. http://wiki.ros.org/ethzasl_icp_mapping
6. M. W. M. Gamini Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, A Solution to the Simultaneous Localization and Map Building (SLAM) Problem, IEEE Transactions on Robotics and automation, 17(3), 2001.
7. S. Thrun, D. Fox, W. Burgard, and F. Dellaert, Robust monte carlo localization for mobile robots, Artificial intelligence, 128(1-2), pp. 99141, 2001
8. C. Weyers and G. Peterson, Improving Occupancy Grid FastSLAM by Integrating Navigation Sensors, IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 859-864, 2011.
9. J. Zhang, Y. Ou, G. Jiang and Y. Zhou, An Approach to Restaurant Service Robot SLAM, Proceedings of the IEEE International Conference on Robotics and Biomimetics, pp. 2122-2127, 2016.
10. G. Hu, S. Huang and G. Dissanayake, Evaluation of Pose Only SLAM, IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3732-3738, 2010.
11. Y. C. Lee and S.H. Park, 3D map building method with mobile mapping system in indoor environments. In 2013 16th International Conference on Advanced Robotics (ICAR), pp. 1-7. IEEE.
12. Z. Su, X. Zhou, T. Cheng, H. Zhang, B. Xu, and W. Chen, Global Localization of a Mobile Robot Using Lidar and Visual Features, Proceedings of the IEEE International Conference on Robotics and Biomimetics, pp. 2377-2383, 2017.
13. A. Ratter and C. Sammut, Fused 2D/3D Position Tracking for Robust SLAM on Mobile Robots , IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1962-1969, 2015.
14. Fetch Robotics. https://docs.fetchrobotics.com/
15. Robot Operating System (ROS). https://docs.fetchrobotics.com/
16. S. Thrun, W. Burgard, D. Fox, Probabilistic robotics. MIT press. (2005)