

Received April 10, 2020, accepted April 24, 2020, date of publication April 30, 2020, date of current version May 14, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2991465

Submap-Based Indoor Navigation System for the Fetch Robot

YONGBO CHEN¹, BRENTON LEIGHTON¹, HUISHEN ZHU², XIJUN KE², SONGTAO LIU²,
AND LIANG ZHAO¹

¹Centre for Autonomous Systems, University of Technology Sydney, Sydney, NSW 2007, Australia

²Shanghai Institute of Micro Motor (The 21st Research Institute of China Electronics Technology Group Corporation), SIMM (CETC21), Shanghai 200086, China

Corresponding author: Liang Zhao (liang.zhao@uts.edu.au)

This work was supported by the Shanghai Institute of Micro Motor (SIMM) (Develop the SLAM Algorithm for Fetch Robot in the Indoor Office and Corridor Environment).

ABSTRACT In this paper, we present a novel navigation framework for the Fetch robot in a large-scale environment based on submapping techniques. This indoor navigation system is divided into a submap mapping part and an on-line localization part. For the mapping part, in order to deal with large environments or multi-story buildings, a submap mapping framework fusing two-dimensional (2D) laser scan and 3D point cloud from RGBD sensor is proposed using Google Cartographer. Meanwhile, several image datasets with corresponding poses are created from the RGBD sensor. Thanks to the submap framework, the error is limited corresponding to the size of the map, thus localization accuracy will be improved. For the on-line localization, so as to switch the submaps, the on-line images from the RGBD sensor are used to match the database images using DeepLCD, a deep learning based library for loop closure. Based on the information from DeepLCD and odometry, adaptive Monte Carlo localization (AMCL) is reinitialized to finish the localization task. In order to validate the result accuracy, reflectors and a motion capture system are used to compute the absolute trajectory error (ATE) and the relative pose error (RPE) based on the Gaussian-Newton (GN) algorithm. Finally, the proposed framework is tested on the Fetch simulator and the real Fetch robot, including both submap mapping and on-line localization.

INDEX TERMS Indoor navigation, Fetch robot, submap, error evaluation.

I. INTRODUCTION

In recent years, indoor navigation has drawn increasing attention in the robotics research area. As the basis of a whole robot system, it provides the robot pose and the surrounding map in an unknown environment. Simultaneous localization and mapping (SLAM) is one of the commonly used methods in the indoor navigation [1]. Generally speaking, in order to solve the SLAM problem, it is divided into a front-end part, which mainly deals with the input sensor data, and a back-end part involving a non-linear optimization problem to create a consistent map or several submaps [2], [3]. Based on the different sensor types for the front-end part, the SLAM methods can be further divided into laser-based SLAM [4], [5], visual SLAM [6], sonar-based SLAM [7], and so on. For the laser-based SLAM, many systems, commonly combining with odometry and inertial measurement unit (IMU)

information, create an occupancy grid map [9] using bundle adjustment (BA) [10] or a point cloud map using iterative closest point (ICP) [11]. As an important research direction, the SLAM problem can also combine with some other areas, such as path planning [12], [13] and control [14], called as active SLAM [15] and planning under uncertainty [16], to get more accurate results. This paper aims to solve the indoor navigation problem by obtaining the occupancy grid map and then performing the localization task in the obtained map.

Based on the fusion of data from a two-dimensional (2D) laser sensor and an RGB-D camera sensor, this work focuses on the navigation system of the Fetch robot using a deep learning-based SLAM system and multiple submaps. It provides mapping and localization information for the following planning and control system in the relatively large environment. We use Google Cartographer [4] to build multiple sequential submaps with the corresponding image library, then use DeepLCD [17] and the adaptive Monte Carlo localization (AMCL) Robot Operating System (ROS) package to

The associate editor coordinating the review of this manuscript and approving it for publication was Hiram Ponce.

perform the on-line map switching and localization within multiple submaps.

As the state-of-the-art technology in the laser-based SLAM, Cartographer provides the real-time 2D/3D SLAM across multiple platforms and sensor configurations with a very good performance even in some challenging indoor environments. Similarly, AMCL, which implements the Kullback-Leibler distance (KLD)-sampling Monte Carlo localization approach, provides a 2D probabilistic localization system for the mobile robot. For the map switching, we use the DeepLCD C++ library, which is a convolutional autoencoder and widely used for online SLAM loop closure. A pre-trained model with a good image matching performance has been included in the DeepLCD C++ library, which is easy to use.

The main structure of this paper is shown as follows: In Section II, we review some literature to summarize the focused research area. We describe our focused problem and present our solution framework briefly in Section III. In Section IV, we describe the submap mapping part, including the sensor data fusion and submap mapping with the image databases. In Section V, we describe the on-line localization implementation using the deep learning based method and the AMCL method. Then, the performance evaluation methods for the proposed framework using the real Fetch robot are presented in Section VI. Following, the simulations and experiments are performed to verify the practicality of the proposed method in Section VII. Finally, in Section VIII, we make a brief conclusion for this paper.

The main contributions of this paper include:

- The novel submap mapping framework with corresponding image databases.
- The application of DeepLCD for map switching and AMCL initialization.
- The novel error evaluation methods to get the absolute trajectory error (ATE) and the relative pose error (RPE) using reflectors and a motion capture system.
- The on-board implementation for the presented method using a Fetch robot.

II. RELATED WORK

At the early stage of the SLAM technology, the research topic focuses on the filter-based method, including Extended Kalman Filter (EKF) [18], [19], Particle Filter (PF) [20], Extended Information Filters (EIF) [21] and so on. In [22], the authors present a SLAM framework, based on the Kalman Filter (KF) and FastSLAM algorithm, which operates a Rao-Blackwellized Particle Filter (RBPF) to fuse the multiple sensors, including stereo cameras, inertial measurements, and robot odometry. In [23], the authors apply the RBPF method in the SLAM task of a restaurant service robot using a depth camera and a motion base with odometry and gyroscope. In the inertial navigation systems (INS)/wireless sensors networks (WSNs)-integrated navigation system, the authors present an adaptive iterated Extended Kalman Filter (AIEKF), which uses the noise statistics estimator in

the iterated Extended Kalman Filter (IEKF) [24]. Recently, the state-of-the-art filter-based SLAM is the invariant EKF SLAM method [25], which uses a novel uncertainty representation to derive a right invariant error EKF and preserves the invariance property. In fact, this method is inspired by the deterministic nonlinear observer on Lie groups, for continuous-time systems with discrete observations [26]. The authors in [27] prove its convergence properties and compare it with the SO(3)-EKF, the Robocentric-EKF and the “First Estimates Jacobian” EKF. In [28], the authors present that its result accuracy is comparable even with the nonlinear least squares optimization based SLAM.

Because of the development of the sparse linear algebra, the SLAM technology develops from the filter-based method to the optimization-based method by exploring the sparsity of the factor graph. The optimization-based SLAM methods can roughly be divided into two categories: Ones use iterative nonlinear optimization methods, such as the Gauss-Newton method [29] or the gradient method, to get locally optimal solutions. And the other recent works show how to obtain globally optimal solutions of the SLAM problem via convex relaxations [30].

Due to the better consistency and efficiency, the optimization-based methods, especially for the pose-graph SLAM framework [31], are widely applied in the indoor navigation methods of multiple kinds of the robot systems. The authors in [32] present a mobile mapping system (MMS) of which the front-end is the map-based scan matching and back-end is the graph-based optimization. Recently, the authors in [33] develop a vertex-to-edge weighted closed-form algorithm to reduce camera drift for the dense RGB-D indoor SLAM using the factor graph optimization. Two popular state-of-the-art technologies in laser-based SLAM, Cartographer and RTAB-map [34], also use the pose-graph optimization method as their efficient back-ends.

It is easy to find that using only one kind of sensor is hard to offer the platform an accurate and robust solution. The data fusion between laser and camera is one of the popular solutions in multiple sensor SLAM. Based on pose-graph optimization, the authors in [35] present a reliable global localization approach using both laser and camera sensors with the capability of addressing the kidnapped robot problem. Based on ICP and pose-graph optimization, the authors present a novel algorithm for fusing laser range finder and RGB-D data to accurately track the position of a robot and generate large 3D maps [36]. As the state-of-the-art technology, the authors in [37] present an EKF-based multi-sensor fusion framework to achieve practical autonomous navigation of UAVs in warehouse environments based on a highly-composed low-cost sensing system, which includes a 2D laser scanner, a 1D range sensor, three cameras (forward, upward and downward), and an IMU. In [38], using the relatively simple sensing system, the authors present a lidar-inertial localization and mapping framework using the Gaussian preintegrated measurements.

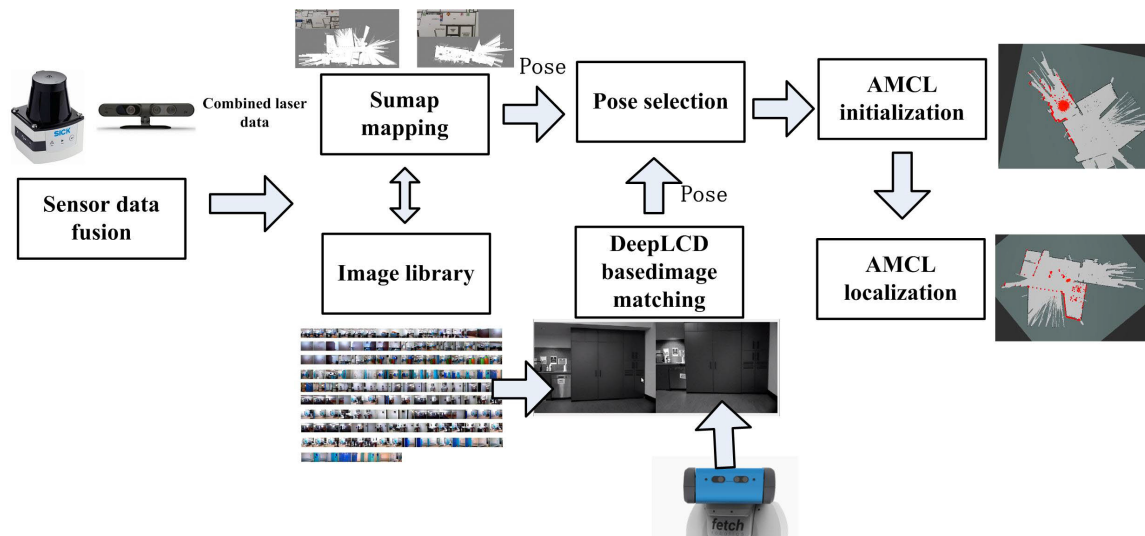


FIGURE 1. Whole solution framework.

III. PROBLEM DESCRIPTION AND SOLUTION FRAMEWORK

This paper aims to solve the indoor navigation problem for the Fetch robot in a large-scale unknown environment. The main task of the robotic indoor navigation problem is to map its surrounding environment and to estimate the robot poses during the moving process of the robot. The mapping and localization can be performed simultaneously or decoupling using the on-board sensing system. For Fetch robot, it is powered by a RGBD sensor on its head and a 2D laser on its base. In order to solve the real-world indoor navigation, we need to deal with the following challenges:

- Many robust indoor navigation systems are based on the laser system. Compared with the 3D laser, the 2D laser is much cheaper. However, the indoor environment commonly includes some stereoscopic obstacles which are hard to be detected by the 2D laser.
- Getting a consistent map with high accuracy is difficult for large-scale indoor mapping. The accuracy of the mapping will directly affect the performance of the following localization.
- Even for some small scenarios, such as the multi-storey building with the elevators, to build a single global map is difficult.
- Evaluating the accuracy of the localization result is also necessary for the real-world implementation on the Fetch robot. The navigation system is the basis of the following planning and control steps.

So as to deal with these difficulties, we present a complete and practical framework based on sensor data fusion, multiple submaps mapping, and deep learning based localization. The complete solution framework is shown in Fig. 1. We firstly combine the flattened RGBD data within the 2D laser data. By this way, we can partly map the stereoscopic obstacles using the 2D fused laser data, which can deal with the first

challenge and help the robot to avoid these obstacles. Then, we use the fused data to build several submaps as well as several image databases, which include the poses of the robot corresponding to each library image. When the robot moves into an area corresponding to the new submap, by using the DeepLCD-based image matching, we can switch the submap. Then, based on the image databases and the pose transformation, we can initialize the AMCL particles in a relatively small area inside the new submap. Moreover, using AMCL, we can perform localization within the submap easily. Thanks to the submap mapping and localization, we can easily get the highly-accurate submaps and map the multi-storey building, which means that it helps us to overcome the second and third challenges. Finally, in this paper, we also present several practical evaluation systems to test the accuracy of the localization results, which is corresponding to the last challenge.

IV. SUBMAPPING BASED ON SENSOR FUSION AND IMAGE LIBRARY

A. SENSOR DATA FUSION

In the indoor environment, because of the environment complexity in the vertical direction, using only a 2D laser is very dangerous for navigation. For many obstacles, including desks, chairs, beds and so on, only a small part will be detected by the laser scan, which makes the mapped obstacle areas seem to be obstacle-free. Different from the 2D laser, the RGBD camera sensor can offer a stereoscopic version for the robot navigation with a limited range. There is a SICK TIM571 laser and a Primesense Carmine 1.09 short-range RGBD sensor locating at the bottom and head of the Fetch robot, respectively. We can easily read their output data from their corresponding ros-topics ‘base_scan’ and ‘head_camera/depth_downsampled/points’.

So as to solve these problems, we aim to merge the data from the 2D laser and the RGBD camera together for

the mapping. The basic idea for sensor data fusion is to transform the 3D point cloud from the RGB-D camera into the frame of the laser, remove points that belong to the floor, convert the point cloud into a laser scan, and then combine the generated laser scan with the actual laser scan. The transform process can be written as:

$$X_b = R_b^c p_c + T_b^c \tag{1}$$

where X_b is the transformed 3D point in the laser frame, R_b^c , and T_b^c are the rotation and translation between camera and laser frame, which are saved in the tf rostopic, p_c is the original 3D point from the RGBD camera.

The 3D point cloud from the RGBD camera commonly includes the points on the floor, which may make the environment lack of free-space for the general laser mapping tool. In order to solve this problem, any transformed points lower than the level of the laser scanner will be deleted. We achieve this goal by transforming the points into the laser scanner frame and filtering any points with Z-axis values less than zero based on Point Cloud Library (PCL). The obtained 3D point cloud can be flattened into a pseudo laser scan so that it can be combined with the real laser scan. This operation created with the same field of view and step angle as the real laser scan, and these two are combined by saving the shorter range value into the combined laser scan data. The fusion process can be written as:

$$\rho = \begin{cases} \rho_1 & \text{dist}(\rho_1, x_v) \leq \text{dist}(\rho_2, x_v) \\ \rho_2 & \text{dist}(\rho_1, x_v) > \text{dist}(\rho_2, x_v) \end{cases} \tag{2}$$

where ρ is the selected fusion points, ρ_1 and ρ_2 are respectively the hit points of the laser point and the flattened RGBD camera point. $\text{dist}(\star, x_v)$ means the distance between \star and the robot ‘base_link’ position. Fig. 2 shows this process and the fusion data. Finally, the generated data are published as the ROS topic ‘base_scan_combined’, and are subscribed by Cartographer for creating a 2D occupancy map.

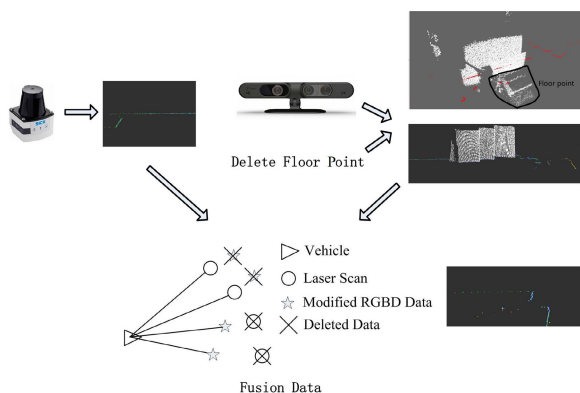


FIGURE 2. Data fusion operations and results.

B. SUBMAP MAPPING AND IMAGE LIBRARY

After obtaining the merged data, we will use it with Google Cartographer to build the 2D occupancy grid map. It is easy

to know that if the mapping area is too big, with the growth of the map area, the consistency of the SLAM map will reduce greatly. So we prefer to use the submap to solve this problem and the obtained maps will have better accuracy compared with the global one [39].

When the Fetch robot moves more than L_F (defined by user), we will build a new local submap based on the last pose of the previous submap. The last poses in the different maps will be saved as a list $\{P_1^2, P_2^3, P_3^4, \dots, P_{n-1}^n\}$. In every submap, we use the Google cartographer to subscribe to the ROS topic ‘base_scan_combined’. As shown in Fig. 3, during these processes, we meanwhile subscribe to the RGBD images from the head camera when the robot meets some thresholds, such as the rotated angles θ_r and the moved distance l_r . These images will be saved as an image database for the submap switching during localization. At the same time, the local poses and the submap number corresponding to the images will also be recorded.

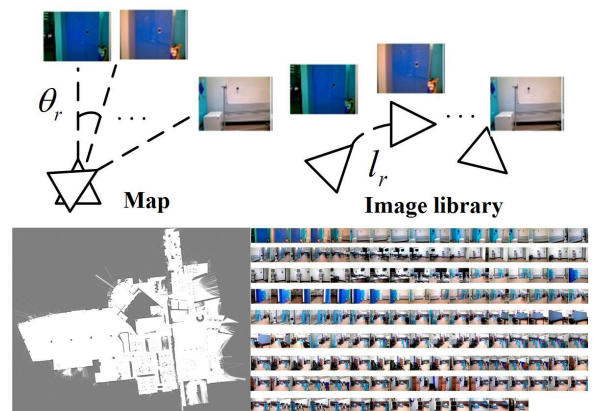


FIGURE 3. Submap mapping and image library.

In this process, we encounter an issue for the fused laser scan, due to the significant differences of the ranges and fields of view of the two sensors (RGB-D camera: 3 m maximum range and 54° horizontal field of view; Laser scanner: 25 m range and 220° horizontal field of view). We find that the occupancy grids will be quickly erased once the obstacles are out of range of the RGB-D camera. In order to solve this problem, we successively build two occupancy grid maps to keep the point cloud obstacles by adjusting the parameter ‘insert_free_space’. One map containing all obstacles but no unoccupied cells and the other one with ‘insert_free_space’ setting to true. Finally, the two maps are combined to produce a map that contains both free space and all obstacles. For every pixel of these two maps, the combination process can be written as:

$$\psi = \begin{cases} \psi_1 & \psi_1 \geq H_{r1} \text{ and } \psi_2 < H_{r2} \\ \psi_2 & \text{else} \end{cases} \tag{3}$$

where ψ , ψ_1 , and ψ_2 are respectively the grayscale of the corresponding pixels in the combined map, the map (“insert_free_space”= false), and the map

(“insert_free_space”= true); H_{r1} and H_{r2} are the grayscale thresholds. Commonly, H_{r1} is close to 1, which means that the pixel is occupied, and H_{r2} is close to 0, which means that the pixel is obstacle-free. By this way, the missing occupancy grids corresponding to the 3D RGBD data will be added into the combined map, as shown in Fig. 4.

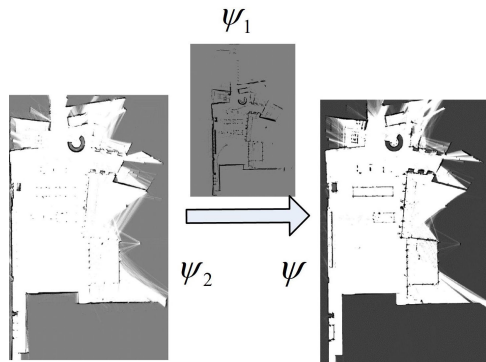


FIGURE 4. Map fusion combining the RGBD data and the laser data.

V. SUBMAP LOCALIZATION BASED ON DEEP LEARNING

By the above steps, several continuous submaps are built with an image database. Our following steps aim to deal with the submap switching and localization in multiple submaps.

A. SUBMAP SWITCHING BASED ON DeepLCD

Thanks to the image database, we use the deep learning-based method (DeepLCD) to deal with the map switching. DeepLCD is a C++ library for the on-line SLAM loop closure. Our submap switching method uses the image matching, similar to the loop closure, to identify the number of the submaps. Based on the pre-trained model in DeepLCD, we can match the real-time image from the head camera with the image database. In this process, every image will be scored by the DeepLCD network. If more than N_t obtained camera images are continuously matched with the image database, the submap will be switched and the map for localization will be set as the corresponding submap.

As shown in Fig. 5, after enough matching between the camera images with the image database, the last matched image will be selected to output its corresponding pose (red one) p_v^1 .

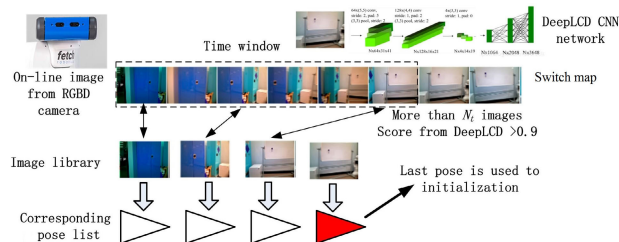


FIGURE 5. Submap switching and localization using the DeepLCD network.

B. LOCALIZATION INITIALIZATION USING IMAGE LIBRARY AND POSE TRANSFORMATION

After the map switching, in order to use the AMCL method, we need to reinitialize the localization by sampling some particles in some areas. However, the rough global sampling is not robust and is also slow for the particles converging, so it is common to sample in a small area. This section aims to find a relatively accurate area for the AMCL initialization by the image libraries and the pose transformation.

Besides this initialization pose $p_v^1 = (x_v^1, \theta_v^1)$, we can also obtain a transformed pose $p_t^1 = (x_t^1, \theta_t^1)$ using the sequential last poses $\{P_1^2, P_2^3, \dots, P_{j-1}^j = (X_p^j, \theta_p^j), \dots, P_{n-1}^n\}$. Using the AMCL, we can get the localization result of the robot: $p_j^{amcl} = (x_j^{amcl}, \theta_j^{amcl})$ in the j -th submap. Because the last pose of the previous map is the same as the first pose of the latter map, this pose is switched by the $j + 1$ -th submap, we can use the following equation:

$$\begin{aligned} x_t^{j+1} &= R_p^j \top (x_{j+1}^{amcl} - X_p^j), \\ R_p^j &= \begin{bmatrix} \cos(\theta_p^j) & -\sin(\theta_p^j) \\ \sin(\theta_p^j) & \cos(\theta_p^j) \end{bmatrix}, \\ \theta_t^{j+1} &= \theta_j^{amcl} - \theta_p^j. \end{aligned} \quad (4)$$

Even though the DeepLCD algorithm commonly obtains the good matched pose p_v^1 , it is still hard to guarantee that we can get the robust switched poses. So, in order to get the more robust result, we combine the above two poses and initialized the area for the AMCL algorithm by the following equation:

$$x_c = \begin{cases} x_v^j & \|x_v^j - x_t^j\| > C \\ x_t^j & \|x_v^j - x_t^j\| \leq C \end{cases} \quad (5)$$

where x_c is the center position of the initialized area for the AMCL algorithm, C is the index which means that when these two poses get closed to each other, we will use the coordinate of the transformed pose x_t^j , and otherwise, we will believe the matched pose p_v^j .

Based on the obtained center point $x_c = (x_x^c, x_y^c)$, we can define a big squared area, of which 4 vertexes are $(x_x^c - 2, x_y^c - 2)$, $(x_x^c - 2, x_y^c + 2)$, $(x_x^c + 2, x_y^c - 2)$, and $(x_x^c + 2, x_y^c + 2)$, for the initialization of the AMCL method. The particles will be spread randomly in this squared area. Then, using the AMCL, the particles will converge after matching the laser scan with the mapped environment.

VI. PERFORMANCE EVALUATION

In order to verify the accuracy of the localization results in the submaps, we present two ways to evaluate the relative position error and the absolute pose error respectively based on the reflectors and the motion capture system. The laser reflector is a reflective material, which can increase the intensity of the laser. By reading the intensity and angle of the laser, we can compute the reflector coordinate based on the laser frame. The motion capture system is a tool to obtain the marks trajectory and the poses of the rigid body in its defined

coordinate system. We use the industry product from Opti-Track company, which can help us to get the coordinates of the marks at a very high rate (more than 40 Hz). The relative position error is defined as the distance error of two poses, of which distance values are obtained by the AMCL localization results and the other localization results. The absolute pose error is defined as the absolute coordinate errors between the estimated poses based on the AMCL method and the ground truth based on the simulation or the highly accurate localization results using the motion capture system.

A. RELATIVE POSITION ERROR EVALUATION BASED ON REFLECTORS

For the relative position error evaluation, the reflectors, which can increase the intensity of the laser scan, are applied. As shown in Fig. 6, two reflectors are set using a tape measure. Because of the intensity change of the laser on the edge of the reflector, we can easily identify the order of the laser ray in a scan and further compute its coordinate based on the ‘laser_link’ frame of the robot. As shown in Fig. 6, the coordinates of the left edges of two reflectors are $e_k = (L_k \cos(\theta_k), L_k \sin(\theta_k))$ and $e_m = (L_m \cos(\theta_m), L_m \sin(\theta_m))$. We can see the reflectors as the poses by also identifying the coordinates of two edges of the reflector, which can be written as: (e_k, R_k) and (e_m, R_m) .

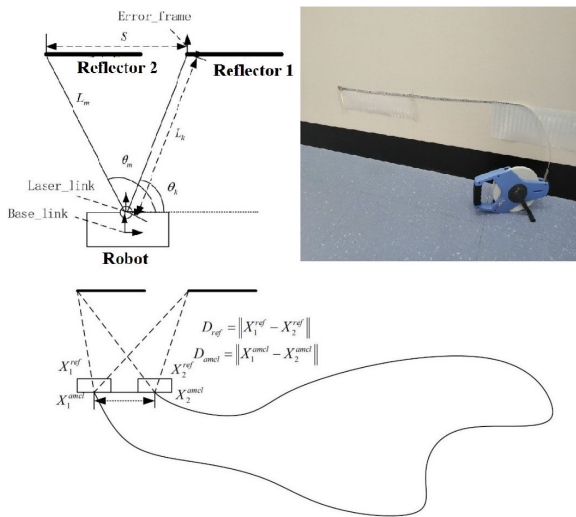


FIGURE 6. Related position error evaluation based on two reflectors.

If the rotation and translation of the ‘base_link’ frame based on the ‘laser_link’ frame is (T_l^b, R_l^b) , we can get that the poses corresponding to the reflectors based on the pose of the robot (original point of the ‘base_link’ frame) are respectively $(e_k^{f1} = R_l^{b\top}(e_k - T_l^b), R_k^{f1} = R_l^b\top R_k)$ and $(e_k^{f2} = R_l^{b\top}(e_m - T_l^b), R_k^{f2} = R_l^b\top R_m)$. Setting the left edge of the reflector 1 as the original point, based on these two reflectors, we can further estimate the coordinate of the robot

pose X^{ref} by the following equations:

$$\begin{aligned} X^{ref} &= (x_r^{f1} + x_r^{f2})/2 \\ x_r^{f1} &= R_k^{f1\top} (0 - e_k^{f1}) \\ x_r^{f2} &= R_k^{f2\top} (0 - e_k^{f2}) - S \end{aligned} \quad (6)$$

where S is the distance between two left edges of the reflector. This equation aims to combine the coordinates of the robot obtained using two reflectors by the weighted way to improve the accuracy of the coordinate.

As shown in the last figure of Fig. 6, we can move the robot to get two poses and their corresponding distances using the AMCL method and the reflectors. The uncertainty level of the distance results using the reflector $D_{ref} = \|X_1^{ref} - X_2^{ref}\|_2$ is about 1cm, which can be seen as the ground truth. The other estimated distance by the AMCL method is defined as: $D_{amcl} = \|X_1^{amcl} - X_2^{amcl}\|_2$. Then, we can define the relative position error by $|D_{amcl} - D_{ref}|$, of which meaning is to obtain the distance error between two poses using the localization result and the ground truth.

B. ABSOLUTE POSE ERROR AND RELATIVE POSITION ERROR BASED ON MOTION CAPTURE SYSTEM

As a more accurate index, the absolute pose error requires the more accurate localization system and the synchronization method to get the better ground truth result. So we use the motion capture system to get the ground truth data. For the record, we can also simply calculate the obtained absolute results to get the relative position error.

1) COMPUTATION OF THE base_link FRAME

The motion capture can get the trajectories of the marks with 1mm accuracy. We lay 9 marks on the pedestal of the Fetch robot, as shown in Fig. 7. Using these 9 marks, we aim to estimate the accurate trajectory for the base_link frame of the Fetch robot. Because the rotation center is the same as the base_link frame of the Fetch robot, we firstly control the Fetch robot to move with several circles without any translation under the record of the motion capture system. By this way, theoretically, the trajectories of the marks

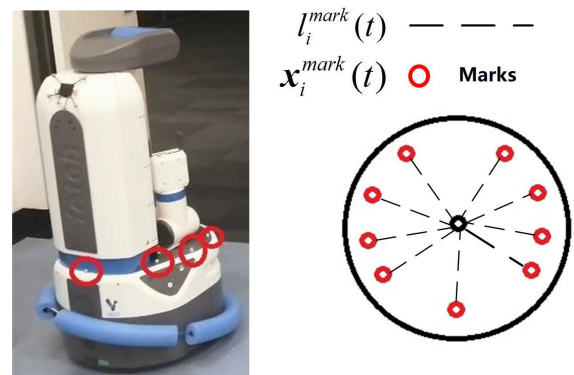


FIGURE 7. 9 marks on the Fetch robot.

$\mathbf{x}_i^{mark}(t) = (x_i^{mark}(t), y_i^{mark}(t))^T$ will locate at several concentric circles. However, because of the noises, we will use the least squares method to estimate the real trajectory of the base_link frame.

As shown in Fig. 7, assuming the trajectories of the marks are $\mathbf{x}_i^{mark}(t)$, $i = 1, 2, \dots, 9$, the rotation center of the robot is \mathbf{x}_i^g and their corresponding radius are l_i^{mark} , $i = 1, 2, \dots, 9$, we can build the following optimization problem:

$$\min_{\mathbf{x}_i^g, l_i^{mark}} \sum_t \left(\|\mathbf{x}_i^{mark}(t) - \mathbf{x}_i^g\|_2^2 - l_i^{mark^2} \right)^2 \quad (7)$$

where $t \in \mathbb{Z}$ means the way-point number of the marks.

In fact, the above optimization problem has a closed form solution, which is obtained by setting the derivation as 0. The obtained $\mathbf{x}^g = (x^g, y^g)$ and l_i^{mark} meet:

$$\begin{aligned} x^g &= -\frac{v_H v_D - v_E v_G}{2(v_C v_G - v_D^2)}, \\ y^g &= \frac{v_H v_C - v_E v_D}{2(v_C v_G - v_D^2)}, \\ l_i^{mark} &= -\frac{\sum_t (x_i^{mark}(t)^2 + y_i^{mark}(t)^2) +}{T}, \\ &\quad \frac{x^g \sum_t x_i^{mark}(t) + y^g \sum_t y_i^{mark}(t)}{T}, \\ v_C &= T \sum_t x_i^{mark}(t)^2 - \sum_t x_i^{mark}(t) \sum_t x_i^{mark}(t) \\ v_D &= T \sum_t x_i^{mark}(t) y_i^{mark}(t) - \sum_t x_i^{mark}(t) \sum_t y_i^{mark}(t) \\ v_E &= T \sum_t x_i^{mark}(t)^3 + T \sum_t x_i^{mark}(t) y_i^{mark}(t)^2 \\ &\quad - \sum_t (x_i^{mark}(t)^2 + y_i^{mark}(t)^2) \sum_t x_i^{mark}(t) \\ v_G &= T \sum_t y_i^{mark}(t)^2 - \sum_t y_i^{mark}(t) \sum_t y_i^{mark}(t) \\ v_H &= T \sum_t x_i^{mark}(t)^2 y_i^{mark}(t) + T \sum_t y_i^{mark}(t)^3 \\ &\quad - \sum_t (x_i^{mark}(t)^2 + y_i^{mark}(t)^2) \sum_t y_i^{mark}(t) \end{aligned} \quad (8)$$

Then, we can obtain all radius of these circle trajectories l_i^{mark} by (8). In the new experiments, we begin to move the robot by the random way instead of the circles. Because the relative positions of the rotation center and the marks are solid, in the moving process of the robot, we can use the relative distances to estimate the robot ‘base_link’ frame using the trajectories of the marks. Assuming the trajectories of the marks are $\mathbf{x}_i^{mark}(t)$, $i = 1, 2, \dots, 9$, with the known of the radius l_i^{mark} , we can get the robot trajectory of the robot ‘base_link’ frame $\mathbf{x}^{ref}(t)$, which will be regarded as the ground truth, by the following optimization problem.

$$\min_{\mathbf{x}_i^{ref}(t)} \sum_t \sum_{i=1}^9 \left(\|\mathbf{x}_i^{mark}(t) - \mathbf{x}^{ref}(t)\|_2 - l_i^{mark} \right)^2 \quad (9)$$

It is noted that, for this new optimization, its ‘base_link’ frame $\mathbf{x}^{ref}(t)$ at t -th step is just related to the coordinates of the marks at t -th step. It is independent of the other steps. We can solve this weighted non-linear least squares problem by the Gaussian-Newton (GN) method.

Let’s consider the following problem: For a measurement vector $\mathbf{z} = F(\mathbf{x}) + \mathbf{W}$, which \mathbf{W} meets $\mathbf{W} \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_z)$, let’s consider the following problem: To find $\mathbf{x} \in \mathbb{R}^n$ to minimize $[\mathbf{z} - F(\mathbf{x})]^T \mathbf{P}_z^{-1} [\mathbf{z} - F(\mathbf{x})]$. Its iteration solution at k -th step is shown as follows:

$$\mathbf{x}_k = (\mathbf{J}_k^T \mathbf{P}_z^{-1} \mathbf{J}_k)^{-1} \mathbf{J}_k^T \mathbf{P}_z^{-1} (\mathbf{z} - F(\mathbf{x}_k) + \mathbf{J}_k \mathbf{x}_k) \quad (10)$$

where \mathbf{J}_k is the Jacobian matrix corresponding to the vector \mathbf{x}_k at k -th step. Based on (10) and $\mathbf{P}_z = \mathbf{I}$, we can obtain the solution $\mathbf{x}^{ref}(t)$.

2) SYNCHRONIZATION

Even though we can get the ground truth data by the above section, because of the different coordinate systems, the different time-stamp, and the different frequency, we still need to deal with the synchronization problem. The goal of the synchronization is to make that the absolute pose error is close to 0, so we can formulate the following problem:

$$\min_{\mathbf{R}^{ref}, \mathbf{T}^{ref}, \Delta t} \sum_t \left(f_t(\mathbf{R}^{ref}, \mathbf{T}^{ref}, \Delta t, \mathbf{x}^{ref}(t_1)) - \mathbf{x}^{amcl}(t) \right) \quad (11)$$

where \mathbf{R}^{ref} and \mathbf{T}^{ref} are translation and rotation between the coordinate system of the AMCL result and the coordinate system of the motion capture system, Δt is the difference of the time axes between two coordinate systems, $f_t(\mathbf{R}^{ref}, \mathbf{T}^{ref}, \Delta t, \star)$ means to transform the trajectory \star by $(\mathbf{R}^{ref}, \mathbf{T}^{ref})$ and synchronize the trajectory by the time difference Δt . $f_t : t_1 \rightarrow t$ also includes the interpolation of the time because of the different frequency.

We need a good initial value for this optimization problem and optimize it by the sequential quadratic programming (SQP) method. The initialization steps for three parameters $\mathbf{R}^{ref}, \mathbf{T}^{ref}, \Delta t$ are shown as follows:

For the rotation and translation of the whole coordinate system $\mathbf{R}^{ref}, \mathbf{T}^{ref}$, we use the iterative closest point (ICP), which is an algorithm employed to minimize the difference between two clouds of points. We can see two trajectories, including the AMCL result $\mathbf{x}^{amcl}(t)$ and the motion capture result $\mathbf{x}^{ref}(t)$, as two clouds of points. And then, ICP can easily get a good initial value of the coordinate transformation $\mathbf{R}^{ref}, \mathbf{T}^{ref}$ between two trajectories. Assuming the AMCL result and the motion capture result have the same number of points T , the formulation of the ICP method can be written as:

$$\min_{\mathbf{R}^{ref}, \mathbf{T}^{ref}} \sum_{t=1}^T \left\| \mathbf{x}^{amcl}(t) - \mathbf{R}^{ref} \mathbf{x}^{ref}(t) - \mathbf{T}^{ref} \right\|_2^2 \quad (12)$$

In fact, it is noted that the normal ICP method does not require the same point number. The ICP result can be seen as the initialization of the above problem (11).

For the time difference Δt , we also need to solve the interpolation problem. The frequencies of the AMCL result and the ground truth are different. Commonly, the frequency of the ground truth data is much higher, so we use the principle of contiguity to find the nearest data in the ground truth data $x^{ref}(t)$ after applying the ICP transformation result. Then all the potential time differences $\Delta t \in (t_a, t_b)$ are tested $t_a + kt_1, k = 1, 2, \dots, [(t_b - t_a)/t_1]$ exhaustively with an interval $t_1 = 0.1s$, where $[(t_b - t_a)/t_1]$ means the maximal integer smaller than $(t_b - t_a)/t_1$. The objective function of the exhaustive method is to minimize the absolute pose error after applying the interpolation operation and the contiguity law. By this way, we can easily get the sub-optimal time difference Δt_{sub} as the initialization for the following optimization on the absolute pose error. This process is shown in Fig. 8.

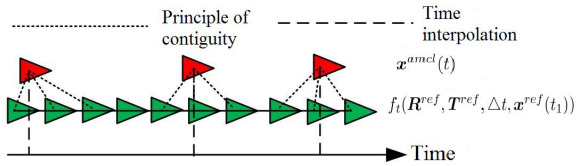


FIGURE 8. Time synchronization.

3) ABSOLUTE POSE ERROR AND RELATIVE POSITION ERROR EVALUATIONS

Based on the initialized values $\Delta t_{sub}, R_{sub}^{ref}, T_{sub}^{ref}$, we can solve the equation (11) and get the optimal solution $\Delta t_{opt}, R_{opt}^{ref}, T_{opt}^{ref}$ by the following framework, as shown in Fig. 9:

Based on this framework, we can get the corresponding poses using the AMCL results and the motion capture system. Then, according to the definition of the absolute pose error, we can compute the absolute coordinate error of each pose along the whole trajectory. Then, we can also pick out several poses to compute the relative position error easily.

VII. SIMULATIONS AND EXPERIMENTS

In this section, we present several simulations and experiments using the Fetch simulator and the real Fetch platform to verify our proposed framework.

A. SIMULATION

1) SIMULATION IN A SMALL OFFICE ENVIRONMENT

We apply the Fetch simulator in a small indoor environment to test the data fusion result and the AMCL result in only one submap. The results are shown in Fig. 10. The top-right small sub-figure is the simulation environment with multiple 3D obstacles, including the Fetch simulator, a desk, a human, a cupboard, and several colorful objects.

As shown in Fig. 10, benefiting from the fusion of the camera data and the laser data, we can build a 2D occupancy grid map with some stereoscopic obstacles, which are hard to be detected by the 2D laser. We also test the AMCL in this environment. The small red vectors around the robot show the

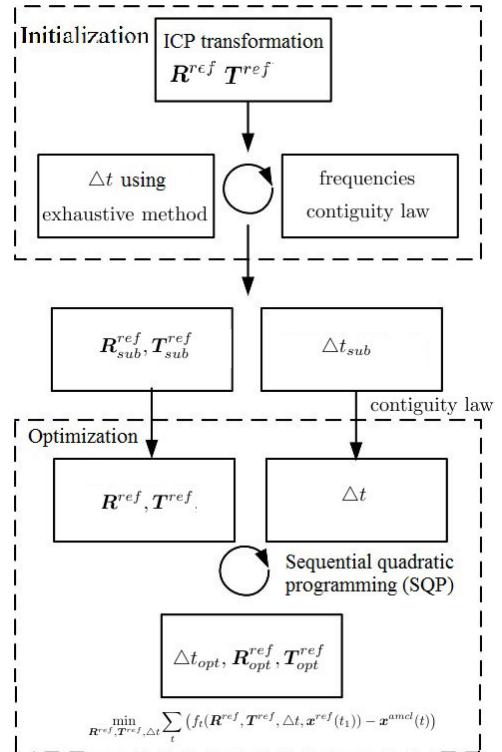


FIGURE 9. Synchronization based on the optimization framework.

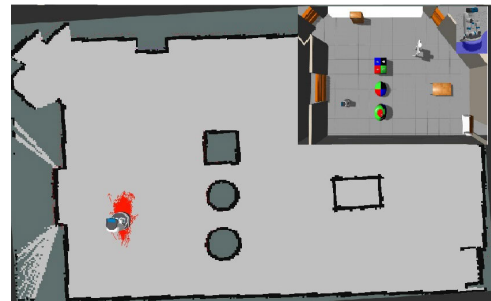


FIGURE 10. Mapping and localization result in a small simulation environment.

estimated pose and its covariance. The results indicate that the absolute pose error is smaller than 10cm compared with ground truth data, which means that it can reach a good result accuracy.

2) SIMULATION IN A LARGE-SCALE ‘Willow’ ENVIRONMENT

In order to further test the mapping and localization performance of the presented framework, we test the Fetch simulator in a relatively large simulation called ‘Willow’. The mapping and localization results are shown in Fig. 11 and Fig. 12. In Fig. 11, the left-down sub-figure shows the simulated environment, which is an office building, and the large one is the built map. In the left-hand side of Fig. 12, we present the compared trajectories, which include the black ground truth data, the blue AMCL estimated data, and the red

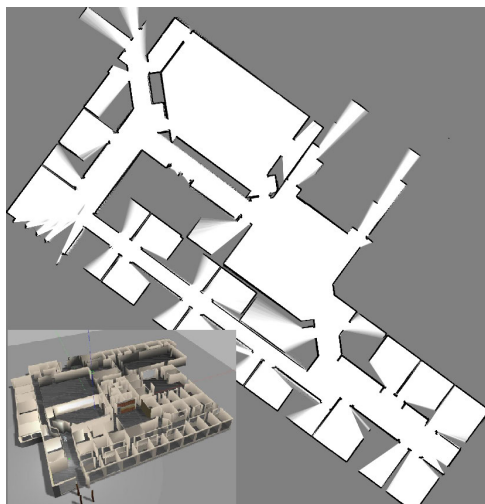


FIGURE 11. Map for the 'Willow' environment.

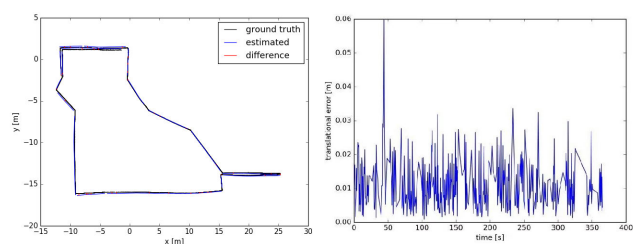


FIGURE 12. Absolute pose error of the AMCL-based localization for simulation.

difference data. The right one shows the absolute pose errors varying with the simulation time.

Based on Fig. 11 and Fig. 12, we can get that the presented framework can get an accurate mapping and localization result in one submap. The largest absolute pose error is 6cm and the mean absolute pose error is smaller than 2cm. These indexes show that the localization accuracy is high in this relatively large-scale simulation environment. Meanwhile, the error figure also shows that the level of the error is consistent and limited.

B. EXPERIMENTS

1) MAPPING USING MULTIPLE SUBMAPS AND BUILT IMAGE DATABASE

In this sub-section, we will build multiple submaps in a real indoor environment, which is in University of Technology Sydney (UTS) Tech Lab. The area of the mapping space is more than 600 m². Meanwhile, we will build a corresponding database during its moving process.

We sequentially build three maps and its image database, as shown in Fig. 13a, Fig. 13b, Fig. 13c, and Fig. 13d. The small figures on the left-hand side of the maps are their corresponding topographic maps. We can simply find that every built submap can well map the surrounding environment with the flatten obstacles. These highly-accurate submaps help the robot to obtain the good localization result. We also show the global map if we do not use the multiple submaps,

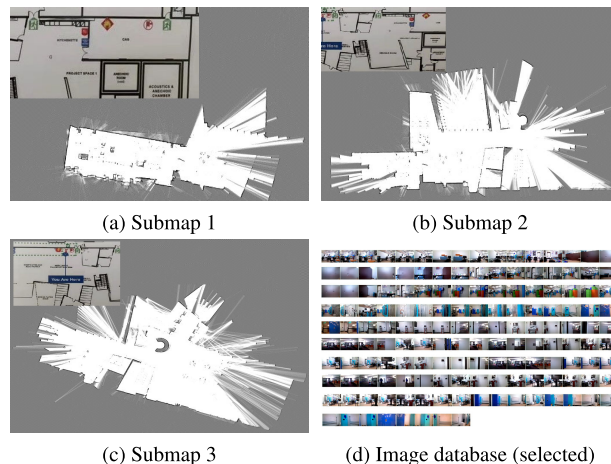


FIGURE 13. Submaps and image database.

shown in Fig. 14 (red: Submap 1; yellow: Submap 2; blue: Submap 3).

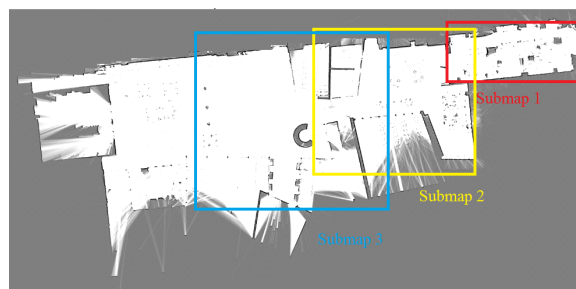


FIGURE 14. On-board global map and the location of the multiple submaps.

The above result shows that, based on our framework, we can build multiple submaps with an image database within several minutes.

2) LOCALIZATION IN MULTIPLE SUBMAPS

Using the built map and the image database, based on our presented framework, we can easily switch map and perform the localization process in the moving process of the Fetch robot.

Our method can switch the map by using DeepLCD-based image matching. And then initialize the AMCL particles within the switching area. The on-line map matching results are shown in Fig. 15 and Fig. 16. In every sub-figure, we can find two images. In Fig. 15, when the on-board camera gets the image and it is matched with the image library, our submap will be switched and the AMCL method will be applied in the new submap. We also output a result, shown in Fig. 17, to show that, after the map switching, the particles will converge successfully based on the given switching area.

The photos of the map switching and localization using the Fetch robot are shown in Fig. 18. The localization trajectories in submaps are shown in Fig. 19.

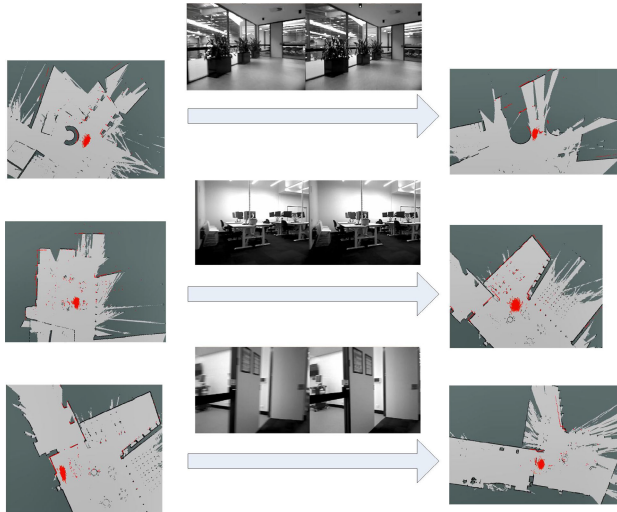


FIGURE 15. Map switching (Two images locating in the middle of the figure are the matched image. The left one is the image from the image database and the right one is from the RGBD camera.)

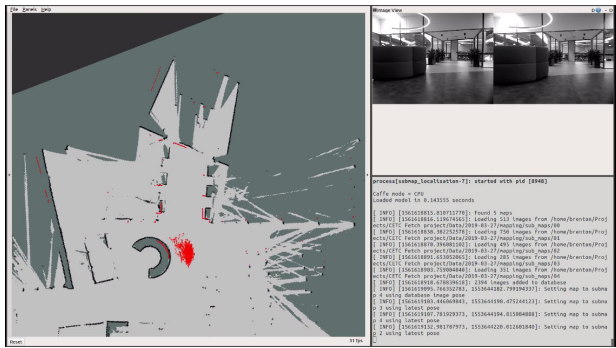


FIGURE 16. Map switching and localization using the AMCL.



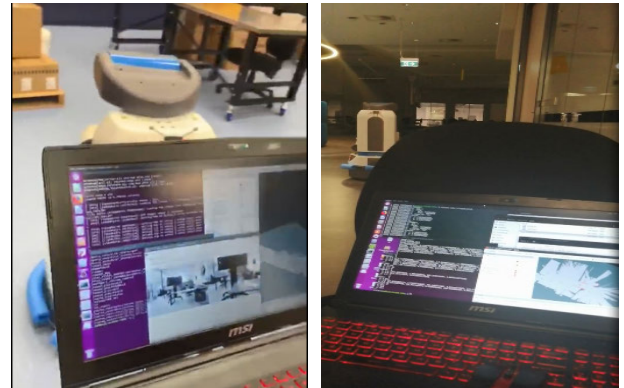
FIGURE 17. Particles convergence after map switching.

These results show that our framework can perform the on-line map switching and localization in the real-world environment successfully.

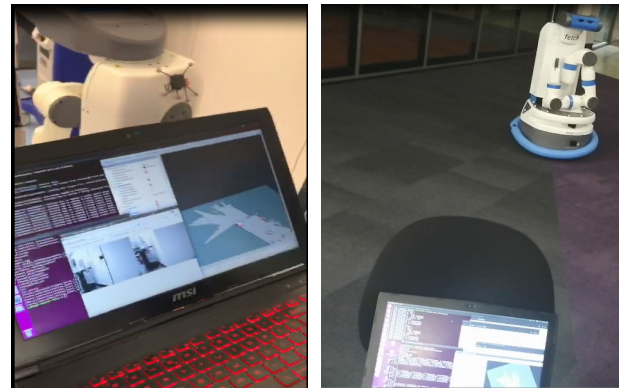
3) ERROR EVALUATION FOR THE EXPERIMENTS USING REFLECTORS

Firstly, we use two reflectors to get the relative position error. The experiment process is shown in Fig. 20. The reflectors are pasted on the wall based on the height of the 2D laser. Then, the robot moves following the trajectory similar to Fig. 6. Finally, we can output the estimated poses based on the coordinates of the reflectors.

Based on the reflectors and method presented in Section VI-A, in 3 experiments, our relative position errors

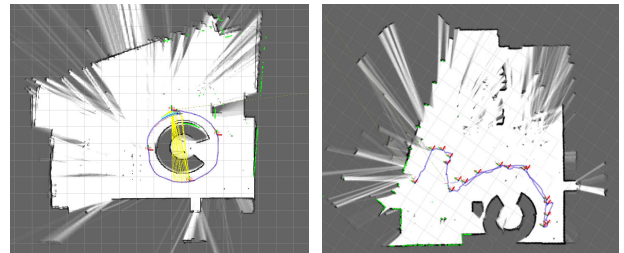


(a) Submap switching from sub-mao 2 to 1 (b) Submap switching from sub-mao 2 to 3



(c) Submap switching from sub-mao 2 to 1 (d) Submap switching from sub-mao 3 to 2

FIGURE 18. On-board implementation for the map switching and localization.



(a) localization trajectory 1 (b) localization trajectory 2

FIGURE 19. On-line localization trajectory using the Fetch robot.

using the AMCL method in one of the submaps are 4.71cm, 4.19cm, and 3.58cm, respectively. It is easy to find that our framework in one of multiple submaps can reach a good localization performance. The results also show that the low-cost reflector-based evaluation is a simple and practical way to evaluate the localization accuracy of the Fetch robot.

4) ERROR EVALUATION USING THE MOTION CAPTURE SYSTEM

Then, in order to get the absolute pose error using the motion capture system, we need to estimate the distances between the marks and the robot rotation center.

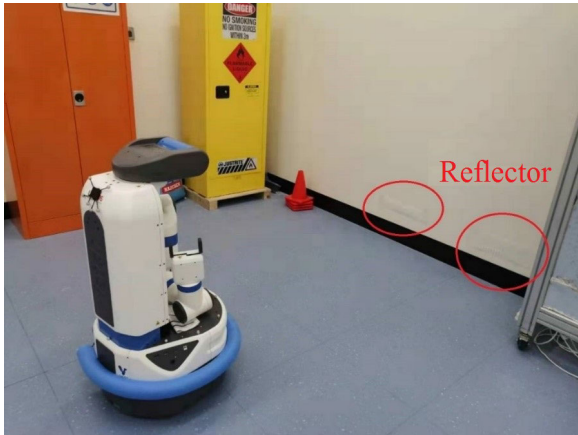


FIGURE 20. Experiment to get the relative position error using the reflectors.

We firstly implement an experiment that makes the robot only perform the rotation action without any translation. By this way, we can record the trajectories of all marks. Using the method mentioned in (8), we can get that the distances between the marks and the center point are respectively 0.2149m, 0.2546m, 0.2249m, 0.2501m, 0.2293m, 0.2552m, 0.2591m, 0.2635, and 0.2609m. The results are shown in Fig. 21 and Fig. 22.

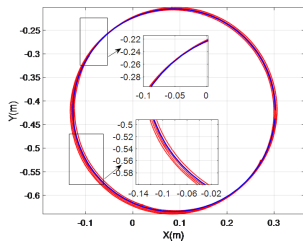


FIGURE 21. Estimated circle for mark 1.

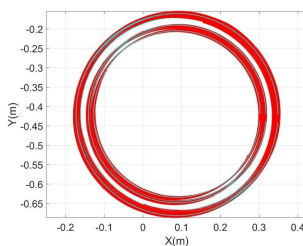


FIGURE 22. Estimated circles for all marks.

Based on the motion capture system, the trajectories of all marks (blue lines) are captured at a very high rate (more than 40Hz). Using these distances, we can get the ground truth data and the absolute pose error of the experiment 1 based on the trajectories of 9 marks using the synchronization technologies, as shown in Fig. 23 and Fig. 24. In the upper figure of Fig. 24, we show the compared result of the estimated poses and the ground truth trajectory after using the transformation. Their ATEs varying with the simulation time are shown in the lower one after applying the synchronization result.

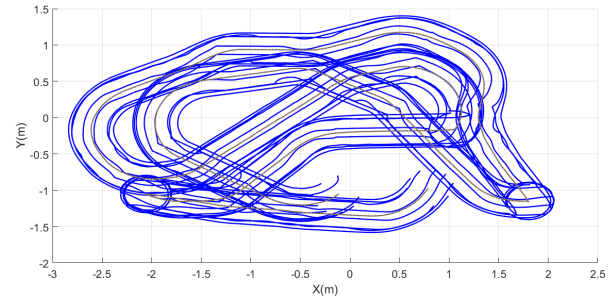


FIGURE 23. Ground truth data (gray line) based on 9 marks (blue lines).

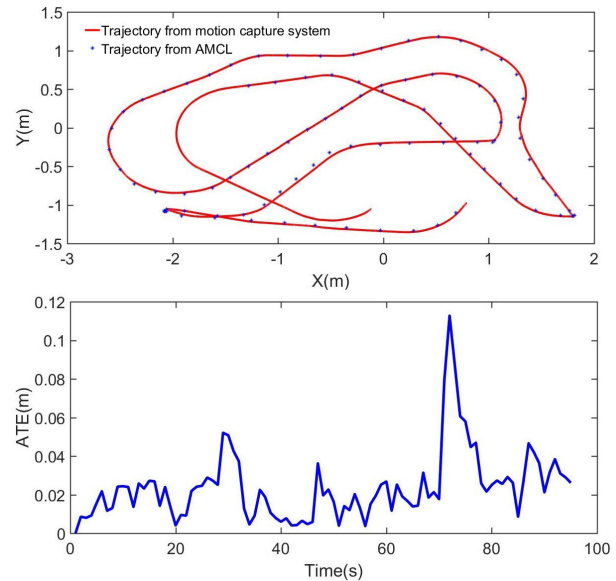


FIGURE 24. Ground truth data (red dense trajectory) and AMCL results (blue sparse trajectory) based on 9 marks and the absolute pose error after using the synchronization technologies.

Based on the results, we can see that the synchronization technologies help to match the trajectories and compute the absolute pose error accurately. It is easy to find that the maximal absolute pose error will be smaller than 12cm and its mean error is about 2.38cm, which means that our framework can get a good localization result in a submap. We also repeat multiple times to perform the ATE experiments using the motion capture, named experiment 2-4. The mean ATEs of the rest of these experiments (Experiment 2-4) are respectively 1.77cm, 2.49cm, and 2.01cm. The final statistical results using the box chart are shown in the following Fig. 25. For the box chart, the lines from the top to the bottom are respectively the minimal, one fourth, half, three fourth, and maximal ATE values except the outliers. The outliers are shown in the start points. After using the synchronization technology, all experiments based on the motion capture system verify that the accuracy of the AMCL localization results in submaps can meet the requirement of the indoor navigation system of the Fetch robot, which is commonly set about 10 to 15 cm.

Of course, we can also output the relative position error using the motion capture system. An example to obtain the

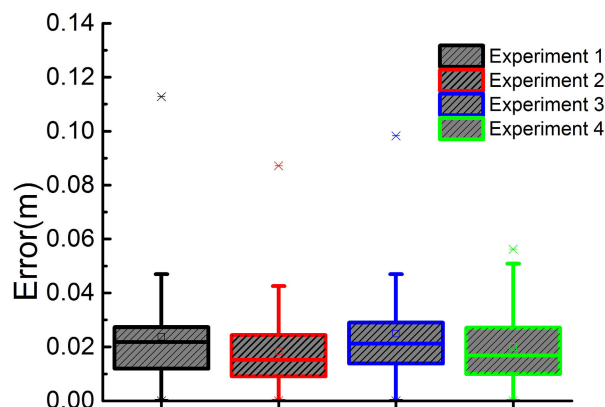


FIGURE 25. Box chart for ATE in different experiments.

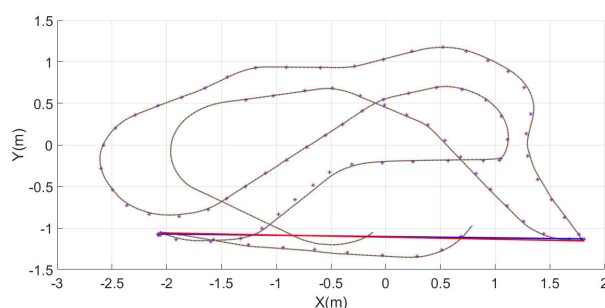


FIGURE 26. An example to obtain the relative position error.

relative position error based on the synchronization results is shown in Fig. 26. In the result, we first obtain the synchronized trajectories, and then pick out several lines (red and blue lines) based on the poses of the AMCL results and the motion capture data. Finally, we compute the difference of the lengths of the lines and obtain the relative position error. The RPE of this example is 1.47cm. We test multiple times to obtain the RPEs for the different trajectories. The RPEs are respectively 2.11cm, 0.98cm, and 1.27cm. The obtained RPEs further prove the accuracy of the localization results. All these results also prove that this motion capture based evaluation system is practical and suitable for the Fetch robot.

VIII. CONCLUSION

This paper presents a practical indoor navigation system for the Fetch robot based on submaps and DeepLCD using Cartographer and AMCL. The navigation system is divided into the mapping part and the on-line localization part. Several sequential submaps are mapped by the fusion data from the 2D laser scan and the RGBD camera sensor. Meanwhile, the image library is also built corresponding to the robot trajectory for the following image matching. Based on the deep learning based image matching results, the AMCL method is initialized and performed in every submap to achieve the highly-accurate localization result. The novel evaluation methods for the indoor navigation are presented and applied using the reflectors and the motion capture system. The simulation and experimental results, using the Fetch robot,

demonstrate that the presented framework can complete the real-world mapping and the localization tasks with a robust and accurate performance.

REFERENCES

- [1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.
- [2] Y. Chen, S. Huang, R. Fitch, L. Zhao, H. Yu, and D. Yang, "On-line 3D active pose-graph SLAM based on key poses using graph topology and submaps," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, Montreal, QC, Canada, Canada, May 2019, pp. 169–175.
- [3] S. Huang, Z. Wang, and G. Dissanayake, "Sparse local submap joining filter for building large-scale maps," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1121–1130, Oct. 2008.
- [4] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LIDAR SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Stockholm, Sweden, May 2016, pp. 1271–1278.
- [5] J. Zhao, S. Huang, L. Zhao, Y. Chen, and X. Luo, "Conic feature based simultaneous localization and mapping in open environment via 2D lidar," *IEEE Access*, vol. 7, pp. 173703–173718, 2019.
- [6] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.
- [7] M. F. Fallon, J. Folkesson, H. McClelland, and J. J. Leonard, "Relocating underwater features autonomously using sonar-based SLAM," *IEEE J. Ocean. Eng.*, vol. 38, no. 3, pp. 500–513, Jul. 2013.
- [8] M. Li, H. Zhu, S. You, L. Wang, and C. Tang, "Efficient laser-based 3D SLAM for coal mine rescue robots," *IEEE Access*, vol. 7, pp. 14124–14138, 2019.
- [9] J. Strom and E. Olson, "Occupancy grid rasterization in large environments for teams of robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Francisco, CA, USA, Sep. 2011, pp. 4271–4276.
- [10] H. Strasdat, J. M. M. Montiel, and A. J. Davison, "Real-time monocular SLAM: Why filter?" in *Proc. IEEE Int. Conf. Robot. Autom.*, Anchorage, AK, USA, May 2010, pp. 2657–2664.
- [11] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing ICP variants on real-world data sets," *Auto. Robots*, vol. 34, no. 3, pp. 133–148, Apr. 2013.
- [12] Y.-B. Chen, G.-C. Luo, Y.-S. Mei, J.-Q. Yu, and X.-L. Su, "UAV path planning using artificial potential field method updated by optimal control theory," *Int. J. Syst. Sci.*, vol. 47, no. 6, pp. 1407–1420, Apr. 2016.
- [13] Y. Chen, L. Zhao, K. M. B. Lee, C. Yoo, S. Huang, and R. Fitch, "Broadcast your weaknesses: Cooperative active pose-graph SLAM for multiple robots," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2200–2207, Apr. 2020.
- [14] S. Frintrop and P. Jensfelt, "Attentional landmarks and active gaze control for visual SLAM," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1054–1065, Oct. 2008.
- [15] Y. Chen, S. Huang, and R. Fitch, "Active SLAM for mobile robots with area coverage and obstacle avoidance," *IEEE/ASME Trans. Mechatronics*, early access, Jan. 1, 2020, doi: 10.1109/TMECH.2019.2963439.
- [16] H. Tahir, M. N. Syed, and U. Baroudi, "Heuristic approach for real-time multi-agent trajectory planning under uncertainty," *IEEE Access*, vol. 8, pp. 3812–3826, 2020.
- [17] N. Merrill and G. Huang, "Lightweight unsupervised deep loop closure," in *Proc. Robot., Sci. Syst. (RSS)*, Cardiff, Wales, U.K., 2018, pp. 1–10, [Online]. Available: arXiv:1805.07703v2
- [18] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Trans. Robot. Autom.*, vol. 17, no. 3, pp. 229–241, Jun. 2001.
- [19] J. Park, S. Lee, and J. Park, "Correction robot pose for SLAM based on extended Kalman filter in a rough surface environment," *Int. J. Adv. Robot. Syst.*, vol. 6, no. 2, pp. 67–72, 2009.
- [20] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust Monte Carlo localization for mobile robots," *Artif. Intell.*, vol. 128, nos. 1–2, pp. 99–141, May 2001.
- [21] M. R. Walter, R. M. Eustice, and J. J. Leonard, "Exactly sparse extended information filters for feature-based SLAM," *Int. J. Robot. Res.*, vol. 26, no. 4, pp. 335–359, Apr. 2007.

- [22] C. Weyers and G. Peterson, "Improving occupancy grid FastSLAM by integrating navigation sensors," in *Proc. IEEE/R SJ Int. Conf. Intell. Robots Syst.*, San Francisco, CA, USA, Sep. 2011, pp. 859–864.
- [23] J. Zhang, Y. Ou, G. Jiang, and Y. Zhou, "An approach to restaurant service robot SLAM," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Qingdao, China, Dec. 2016, pp. 2122–2127.
- [24] Y. Xu, X. Chen, and Q. Li, "Adaptive iterated extended kalman filter and its application to autonomous integrated navigation for indoor robot," *Sci. World J.*, vol. 2014, Feb. 2014, Art. no. 138548.
- [25] K. Wu, T. Zhang, D. Su, S. Huang, and G. Dissanayake, "An invariant-EKF VINS algorithm for improving consistency," in *Proc. IEEE/R SJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 1578–1585.
- [26] A. Barrau and S. Bonnabel, "The invariant extended Kalman filter as a stable observer," *IEEE Trans. Autom. Control*, vol. 62, no. 4, pp. 1797–1812, Apr. 2017.
- [27] T. Zhang, K. Wu, J. Song, S. Huang, and G. Dissanayake, "Convergence and consistency analysis for a 3-D invariant-EKF SLAM," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 733–740, Apr. 2017.
- [28] Y. Zhang, T. Zhang, and S. Huang, "Comparison of EKF based SLAM and optimization based SLAM algorithms," in *Proc. 13th IEEE Conf. Ind. Electron. Appl. (ICIEA)*, May 2018, pp. 1308–1313.
- [29] V. Ila, L. Polok, M. Solony, and P. Svoboda, "SLAM++-A highly efficient and temporally scalable incremental SLAM framework," *Int. J. Robot. Res.*, vol. 36, no. 3, pp. 416–446, 2017.
- [30] D. M. Rosen, L. Carlone, A. S. Bandeira, and J. J. Leonard, "SE-sync: A certifiably correct algorithm for synchronization over the special Euclidean group," *Int. J. Robot. Res.*, vol. 38, nos. 2–3, pp. 95–125, Mar. 2019.
- [31] G. Hu, S. Huang, and G. Dissanayake, "Evaluation of pose only SLAM," in *Proc. IEEE/R SJ Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 3732–3738.
- [32] Y.-C. Lee and S.-H. Park, "3D map building method with mobile mapping system in indoor environments," in *Proc. 16th Int. Conf. Adv. Robot. (ICAR)*, Nov. 2013, pp. 1–7.
- [33] S. Tang, Y. Li, Z. Yuan, X. Li, R. Guo, Y. Zhang, and W. Wang, "A Vertex-to-Edge weighted closed-form method for dense RGB-D indoor SLAM," *IEEE Access*, vol. 7, pp. 32019–32029, 2019.
- [34] M. Labbé and F. Michaud, "RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *J. Field Robot.*, vol. 36, no. 2, pp. 416–446, 2019.
- [35] Z. Su, X. Zhou, T. Cheng, H. Zhang, B. Xu, and W. Chen, "Global localization of a mobile robot using lidar and visual features," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2017, pp. 2377–2383.
- [36] A. Ratter and C. Sammut, "Fused 2D/3D position tracking for robust SLAM on mobile robots," in *Proc. IEEE/R SJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 1962–1969.
- [37] W. Kwon, J. H. Park, M. Lee, J. Her, S. Kim, and J. Seo, "Robust autonomous navigation of unmanned aerial vehicles (UAVs) for warehouses' inventory application," *IEEE Robot. Automat. Lett.*, vol. 5, no. 1, pp. 243–249, Jan. 2020.
- [38] C. Le Gentil, T. Vidal-Calleja, and S. Huang, "Gaussian process preintegration for inertial-aided state estimation," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2108–2114, Apr. 2020.
- [39] L. Zhao, S. Huang, and G. Dissanayake, "Linear SLAM: Linearising the SLAM problems using submap joining," *Automatica*, vol. 100, pp. 231–246, Feb. 2019
- [40] (2019). *Fetch Robotics*. [Online]. Available: <https://docs.fetchrobotics.com/>
- [41] (2019). *Robot Operating System (ROS)*. [Online]. Available: <https://docs.fetchrobotics.com/>
- [42] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA, USA: MIT Press, 2005.



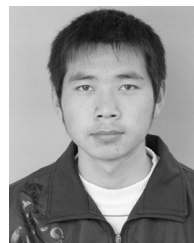
BRENTON LEIGHTON received the B.S. degree in mechanical and mechatronic engineering and the two master's degrees in computer control engineering and research engineering from the University of Technology Sydney, Sydney, China, in 2011, 2015, and 2019, respectively. He is currently a Research Assistant with the University of Technology Sydney. His research interest is in simultaneous localization and mapping.



HUI SHEN ZHU received the bachelor's and master's degree in engineering from the Beijing Institute of Technology (BIT), Beijing, China, in 2016 and 2018, respectively. He is currently an Engineer with the 21st Institute of China Electronic Technology Group Corporation, Shanghai, China. His research interests include mobile robots simultaneous localization and mapping (SLAM) and robot navigation.



XIJUN KE received the B.S. degree in material science from China Three Gorges University, Yichang, China, in 2011, and the Ph.D. degree in mechanical engineering from Shanghai Jiao Tong University, Shanghai, China, in 2017. He is currently an Engineer with the Shanghai Institute of Micro Motor (The 21st Research Institute of China Electronics Technology Group Corporation), SIMM (CETC21), Shanghai. His research interest is in applied robot research.

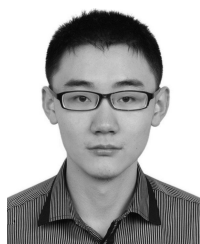


SONGTAO LIU received the master's degree in mechanical engineering from Shanghai Jiao Tong University (SJTU), Shanghai, China, in 2018. He is currently an Engineer with the 21st Research Institute of China Electronics Technology Group Corporation. His research interests include robot localization and path planning.



LIANG ZHAO received the Ph.D. degree in photogrammetry and remote sensing from the School of Earth and Space Science, Institute of Remote Sensing and Geographic Information System, Peking University, Beijing China, in January 2013.

From 2012 to 2014, he was a Postdoctoral Research Fellow with the Centre for Autonomous Systems (CAS), Faculty of Engineering and Information Technology, University of Technology Sydney (UTS), Sydney, Australia. From 2014 to 2016, he worked as a Postdoctoral Research Associate with the Hamlyn Centre for Robotic Surgery, Department of Computing, Faculty of Engineering, Imperial College London, U.K. Since July 2016, he has been a Senior Lecturer with CAS, UTS. His current research interests include surgical robotics, simultaneous localization and mapping (SLAM), monocular SLAM (or structure-from-motion), aerial photogrammetry, optimization techniques in mobile robot localization and mapping, and image guide robotic surgery.



YONGBO CHEN was born in Putian, Fujian, China, in 1990. He received the bachelor's degree in engineering from the Beijing Institute of Technology (BIT), Beijing, China, in 2012. He is currently pursuing the Ph.D. degree with the Centre for Autonomous Systems, Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, NSW, Australia. His research interests include mobile robots simultaneous localization and mapping (SLAM) and UAV motion/mission planning.