2018 Tenth International Conference on Advanced Computational Intelligence (ICACI)
March 29–31, 2018, Xiamen, China

A SLAM with Simultaneous Construction of 2D and 3D maps Based on Rao-Blackwellized Particle Filters

1st Yao li, 2nd Fan Zhun, 3rd Zhu Guijie, 4th Li Wenji, 5th Li Chong, 6th Wang Yupeng, and 7th Xie Honghui
Department of Electronic Engineering
Shantou University
Guangdong, Chinese
15lyao@stu.edu.cn, zhunfan@stu.edu.cn, 16gjzhu@stu.edu.cn, liwj@stu.edu.cn, 15cli@stu.edu.cn, 15ypwang@stu.edu.cn, 14hhxie@stu

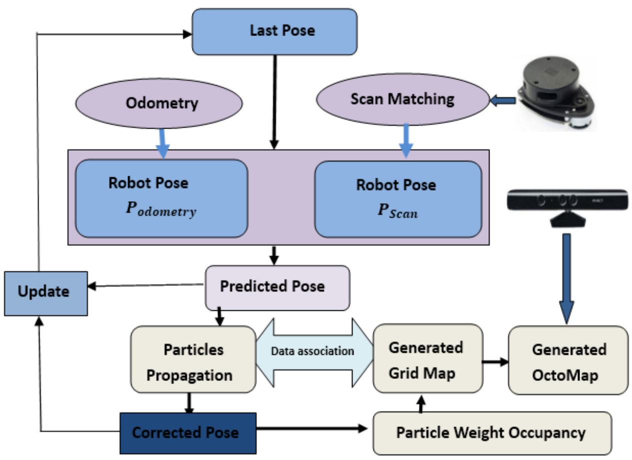

Fig. 1. An overview of our method

***Abstract*—This paper presents a SLAM (Simultaneous Localization and Mapping) method which builds 2D grid maps and generates the OctoMap based on Rao-Blackwellized particle filters. This work combines wheeled odometry and laser scan with particle filter algorithm to get the pose of the robot, and at the same time fuses the data of depth camera to generate OctoMap, OctoMap is an integrated open source framework based on octree, which is well known for its memory efficiency for the representation of 3D environments. The traditional 3D point cloud map cannot be applied in robot navigation. But OctoMap is a 3D occupancy grid mapping, which can be applied to 3D path planning of flying robots and other robots that are equipped with manipulators. In short, the experimental results demonstrate that the proposed methods can make a robot to synchronize building 2D and 3D maps very efficiently.**

***Keywords*—SLAM; rao-blackwellized particle filters; gmapping; 2D grid map; octoMap**

I. Introduction

Localizing the position of a robot and building the map of an unknown environment are the two major tasks of mobile robots[1]. Many researches focus on locating the position of a robot as well as representing the unknown environment. SLAM is about how to localize the position of a robot and to generate a 2D or 3D map of an environment, which is one of the most popular topics in robotics. There are two types of classical and matured SLAM methods, including the visual-based and the laser-based. ORB-SLAM[2] and LSD-SLAM[3] are two popular visual-based SLAM methods, which construct a 3D environment by image information. The advantage of visual-based SLAM is that the data lies in a camera has more information about object characteristics about the environment. But it needs to overcome the scale-drift issue. The laser-based SLAM can be used in large-scale scene, since the laser scanner can reach a long distance and get enough point cloud, totally it can work well at different environments. But the laser scanner can only get distance information, and cannot obtain the characteristics of objects in the environment.

Another problem in SLAM is mapping. 2D and 3D maps have different levels of abstraction about the environment[4]. The 2D occupancy grid map is very popular, and it is widely used in the robot navigation field. However, since 2D occupancy grid maps cannot provide enough space information about complicated environments, it limits the robot to perform complex tasks in three-dimensional space. Hence, building 3D maps becomes a hot filed for mobile robots. The representative work includes 3D sparse map by RGBD-SLAM[5,6]. The OctoMap[7] is a 3D occupancy grid map, which can distinct the free and the occupied space. It is essential for robot navigation.

This paper uses the Rao-Blackwellized Particle Filters(RBPF)[8] to complete SLAM in an indoor scenery, and merges data of laser scanner and Microsofts Kinect to build 2D and 3D maps. The main flow chart of this work is shown in Figure 1. The estimation of the pose of a robot involves two steps, which are prediction step and correction step. The RBPF combines the information of the encoder odometry and scan-scan matching to build an initial 2D map and OctoMap, then uses the particle filters algorithm to relocate the pose of the robot to update the 2D map and OctoMap.

The rest of the paper is organized as follows. Section II introduces the related work. Section III presents the proposed algorithm and the implemented method. Section IV includes experiments to verify the idea and compare the proposed algorithm. The section V concludes the work.

II. Related work

In the field of mobile robots, SLAM is a very basic and critical problem, and is also one of the most popular research directions in the field of robotics[9]. SLAM can actually be treated as a combined solution of localization and mapping. Localization is responsible for determining the current position of a robot in an environment, while mapping consists of collecting sensory data and storing it in a specified form for further processing. As we know, the 2D grid map is more popular in navigation, but 3D map has more information about the environment, and has gradually become a research focus in the field of robotics. In particular, OctoMap is vital for collision-free path planning and safe navigation. There are a lot of related works about 2D mapping and the building of OctoMap.

In paper[10], they proposed a method to convert a sensor-based occupancy grid to a 3D Gazebo[11] OctoMap, which creates realistic landscapes in Gazebo simulation, mainly used to simulate 3D environment by OctoMap. Li build a 3D map by mobile robot equipped with one kinect[4], the main method was iterative closest point, and combined with incremental registration to get well OctoMap, the result perform the fine registration. The paper[12] proposed that based on octree and probabilistic occupancy estimation, the mapping technique of ORB-SLAM can be extended to build 3D reconstruction by Kinect. In the paper[13], K. Kamarudin et al. proposed a method to merge the Kinect and laser scanner to improve the performance of 2D SLAM. The results show that the method is able to detect multi-sized objects and produce more accurate map which shows the benefit of merge sensor data. These papers indicate that OctoMap and the research of data fusion are more and more important for robots in unknown environment. This work is about merging sensor data and synchronize building 2D map and OctoMap with robot equipped with laser scanner and Kinect.

III. METHOD

A. The Preparation of Work

The robot operating system(ROS)[14] is a fast growing intelligent robotic application development framework with supporting the vast majority of sensors and efficient implementation for different methods of SLAM. It is a convenient operating system, which can help researchers to realize the simulation of a real world environment quickly by the gazebo and rviz[10]. In this paper, laser scanner and odometer mounted on a mobile robot are used as the main sensors to build 2D maps. Microsoft Kinect is another sensor to build 3D OctoMap. The main sensors are 'rplidar' and 'Microsoft's Kinect' as shown in Figure 2.

Fig. 2. Microsoft's Kinect sensor and 360^0 rplidar

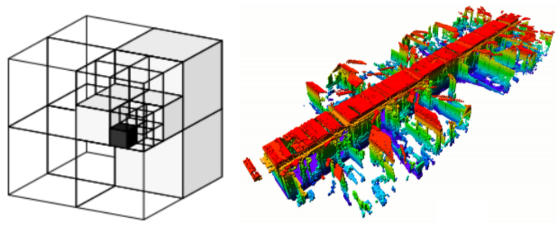

Fig. 3. An octree storing free and occupied positions and an example of OctoMap for the corridor.

B. OctoMap

Octomap is an efficient probabilistic open source framework for 3D grid maps based on octree, which is a way to store data effectively about 3D space occupancy. The main goal of OctoMap is to build an updatable, flexible, compact and complate 3D model. The advantage of the structure of OctoMap is that it is fast to query for obstacles and to compute the distance between a robot and an obstacle. An example of voxel structure and an example of 3D OctoMap for a corridor are shown in Figure 3.

IV. SYSTEM ARCHITECTURE

A. The Theory of RBPF

Considering the SLAM problem, given a mobile robot with unknown pose in an unknown environment, the robot needs to locate itself and update a consistent environment map at the same time. When the initial position of the robot is specified, the robot can construct local maps with the observation of the environmental characteristic. There is a Bayes network graph[9] depicting the basic SLAM problem as shown in figure 4. $z_{1,t}$ is the observation sequence, which represents the observation information from the laser scanner. $u_{1,t}$ represents the input sequence of the odometry, the robot state $x_{1,t}$ represents the position of the robot at time t. In the process of the robot's localization, the robot uses the laser scanner information to update its posture status, and uses the observation information for the state correction. At each pose x_t, it observes nearby landmarks in the map $m = \{m_1, m_2, m_3 \ldots\}$.

The grid mapping with RBPF has shown to be an effective method to solve the SLAM problem. The main idea about the algorithm in this paper is to use particle filters to estimate the path posterior for robot positioning. This particle filter works in analogy to Monte-Carlo Localization[15]. In addition, the map landmark locations are estimated by using Extend Kalman

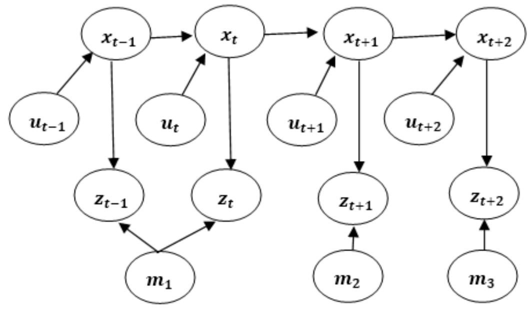

Fig. 4. The SLAM problem depicted as bayes network graph

	robot path	feature 1	feature 2	...	feature N
Particle $k=1$	$x_{1:t}^{[1]} = \{(x\ y\ \theta)^T\}_{1:t}^{[1]}$	$\mu_1^{[1]}, \Sigma_1^{[1]}$	$\mu_2^{[1]}, \Sigma_2^{[1]}$	...	$\mu_N^{[1]}, \Sigma_N^{[1]}$
Particle $k=2$	$x_{1:t}^{[2]} = \{(x\ y\ \theta)^T\}_{1:t}^{[2]}$	$\mu_1^{[2]}, \Sigma_1^{[2]}$	$\mu_2^{[2]}, \Sigma_2^{[2]}$	...	$\mu_N^{[2]}, \Sigma_N^{[2]}$
		$\vdots$			
Particle $k=M$	$x_{1:t}^{[M]} = \{(x\ y\ \theta)^T\}_{1:t}^{[M]}$	$\mu_1^{[M]}, \Sigma_1^{[M]}$	$\mu_2^{[M]}, \Sigma_2^{[M]}$	...	$\mu_N^{[M]}, \Sigma_N^{[M]}$

Fig. 5. Particles are composed of a path estimate and a set of features

Filters (EKF) to update the mean $\mu_{i,t}^{[k]}$ and the covariance$\sum_{i,t}^{[k]}$ of landmarks in the map, it's mainly to construct environment maps. The formula of RBPF is defined as follows:

$$\begin{aligned} p(x_{1:t}, m|z_{1:t}, u_{1:t-1}) \quad &= \\ p(m|x_{1:t}, z_{1:t}) \quad &* \quad p(x_{1:t}|z_{1:t}, u_{1:t-1}) \end{aligned}$$

In this formula, the left $p(x_{1:t}, m|z_{1:t}, u_{1:t-1})$ is to estimate the joint posterior about the map m and the trajectory $x_{1:t}$ of the robot. $p(x_{1:t}|z_{1:t}, u_{1:t-1})$ is the probabilistic model of localization and the $p(m|x_{1:t}, z_{1:t})$ is mapping probabilistic model. As a result of $x_{1:t}$ and $z_{1:t}$ are known in advance, $p(m|x_{1:t}, z_{1:t})$ can be estimated with known poses, which is described in the following part B. The posterior $p(x_{1:t}|z_{1:t}, u_{1:t-1})$ is computed by the particle filter. The sampling importance resampling filter algorithm is used for localization to estimate posterior, which can be summarized by the following part C. Totally, each particle contains an estimated pose of the robot, and obtains the robot accurate positioning, and each particle builds a local map for global map.

B. Map Update with Known Poses

Whether it's a 2D grid map or an OctoMap, it's both occupancy grid map. The algorithm of occupancy grid mapping is defined as follows base on the given data to compute the posterior over maps:

$p(m|z_{1:t}, x_{1:t}, c_{1:t}) = \prod_{n-1}^{N} p(m_n|z_{1:t}, x_{1:t}, c_{1:t})$

Formally, $c_{1:t}$ are the correspondences of measurements and landmarks in the map, m represents the map, $z_{1:t}$ is a series of measurement, and $x_{1:t}$ is the path of the robot. Given the robot path, landmarks are conditionally independent, and the particles are sampled from the motion model, each particle has many landmarks $c_{1:t}$ in local maps. Then the maps are updated and used in the global map.

The occupancy grid map is used to divide the space into finite grid cells. $m = \sum_i m_i$, each m_i has a binary occupancy probability value, '1' represents occupied and '0' represents idle. The notation $p(m_i)$ refers to a probability that a grid cell is occupied. This approach divides the problem of estimating the map into a collection of separate problems: $p(m_i|z_{1:t}, x_{1:t})$, the whole map is approximated as $p(m|z_{1:t}, x_{1:t}) = \prod p(m_i|z_{1:t}, x_{1:t})$, therefore, the estimation of the occupancy probability of each grid cell is a static binary estimation problem. The probability by the log-odds representation of occupancy for each grid cell: $l_{t,i} = \log \frac{p(m_i|z_{1:t}, x_{1:t})}{1-p(m_i|z_{1:t}, x_{1:t})}$. The more certain it is occupied for the larger value of grid cell, otherwise, it is a free state. $l_{t,i}$ gives the probability about a grid cell is occupied or idle statue with the sensor measurement z_t at location x_t. Especially about the robot navigates on a flat surface, the 2D occupancy map is widely used. Certainly occupancy grid techniques can generalize to 3D representations[9].

C. The Process of Localization

In the step of localization, each particle carries along with a local map about the environment, there are following four steps about the process:

(a) Sampling from the motion model: the pose of a robot is predicted by sampling from a proposal distribution π, and is denoted as $x_t^{[k]}$ for the k-th particle at time t. Normally, The motion probability model of the odometry generally adopts the proposed distribution. Drawing a sample according to the motion posterior: $x_t^{[k]} \sim p(x_t|x_{t-1}^{[k]}, u_t)$. The bracketed notation $[k]$ indicates the index of the particle. the pose x_t is associated with the k-th particle. $\mu_t^{[k]}$ and $\sum_t^{[k]}$ are the mean and variance values of k-th landmark location. A set of M particles is shown in Figure 5.

(b) Importance weighting: the importance weight for the k-th particle w_t^k, which represents the proportion between the target distribution $p(x_{1:t}^{(k)})|z_{1:t}, u_{1:t-1})$ and the proposal distribution π mentioned above is defined as follows:

$w_t^k = \frac{p(x_{1:t}^{(k)}|z_{1:t}, u_{1:t-1})}{\pi(x_{1:t}^{(k)}|z_{1:t}, u_{1:t-1})}$

As we know, there are many improved methods about the importance weighting[16].

(c) Resampling: the relatively low importance weight particles will be replaced by other high weight particles or eliminated. It is a very important step. Only a limited number of important particles will be used to approximate the continuous distribution. Some resampling algorithms and analyses are presented in [17].

(d) The last step is map estimating, the estimation $p(m|x_{1:t}, z_{1:t})$ is to compute the corresponding local map for each particle pose, which is used to update the global map by adding the transform local grid maps based on the sample of trajectory $x_{1:t}$ and the history of observations $z_{1:t}$.

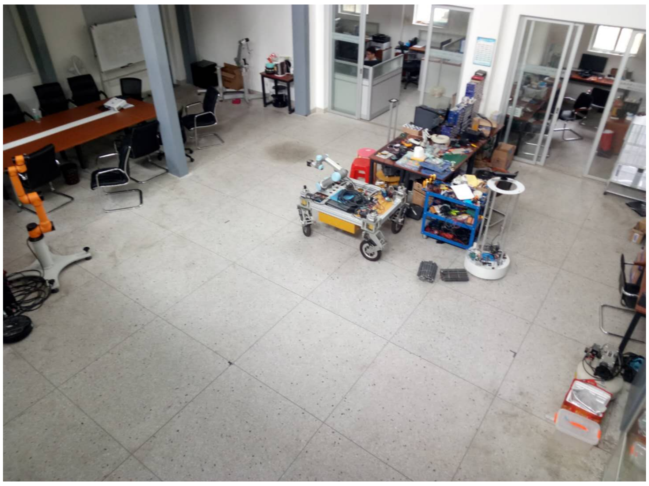

Fig. 6. A complex laboratory scene

D. Loop Closure

In loop closure, a robot moves through an unknown terrain. It encounters landmarks seen previously at some points. It is particularly important to maintain the correlation in a SLAM algorithm. RBPF maintains the correlation through its diversity persevation in the particle sets. Thus, the loops are terminated based on the number of particles. And the sample set with better diversity results in a better performance of loop closing.

V. RESULTS AND COMPARISON

A. Robot Platform and the Test Environment

We adopt the TurtleBot mobile robot as the experimental platform, which equips with a Rplidar sensor, a Kinect sensor, and inertial measurement unit and a laptop with Linux operating system. The ROS Indigo is used.

B. Experimental Results

The resulting 2D grid maps and OctoMap present occupancy grids which can be visualized in RVIZ(ROS visualization tools). Generally, a 2D grid map has three possible pixel values of black for occupied space, white for free space and gray for vaguely defined space with uncertain situation. The resolution of OctoMap is 0.05m, as shown in Figure 7. The color tends to be red with a higher distance to the ground, which represents an infeasible area in 3D environments.

Figure 6 is a complex indoor laboratory environment, and Figure 7 shows the 2D grid maps and OctoMap in RVIZ of Figure 6, where there are a variety of chairs, tables, experimental equipments, cubicles and so on. However, this method still has very good results, and can see the location distribution of the original environment in this map. This method can also be scaled to a larger space, as shown in Figure 8 with a corridor environment.

C. Error Example Comparisons

When the system works for a long time, the processing speed of the system may gradually decline. There are two main reasons, the first one is that the map becomes larger and larger. Another is that the local map of each particle needs to be constantly matched with the global map. Occasionally, the robot may lose location information in the map, resulting a mapping error. In that case, the robot may hit a wall, as shown in Figure 9. In this case, we can change the number of particles sampled from the state of the robot in different environments to improve its efficiency.

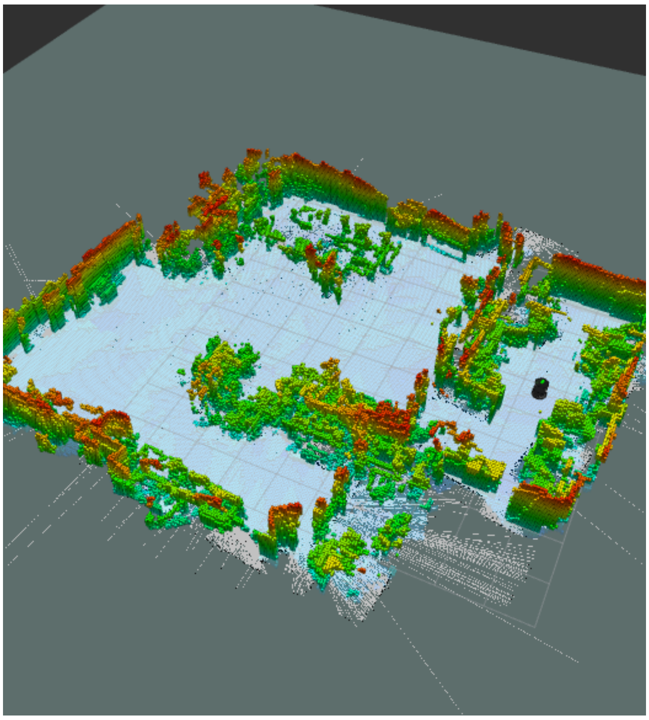

Fig. 7. 2D grid map and OctoMap in RVIZ

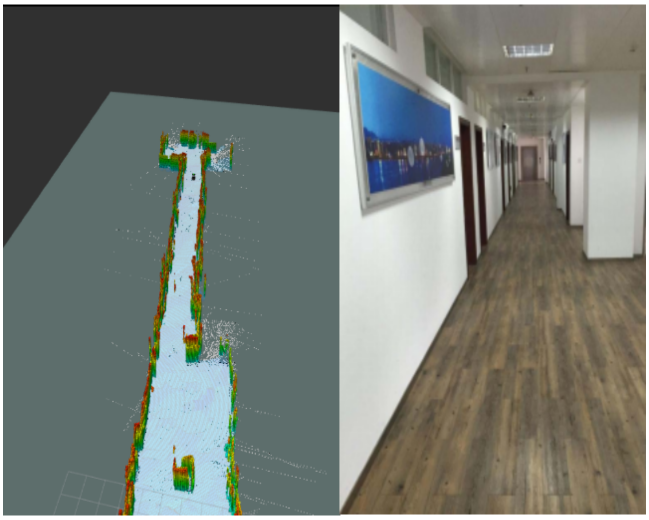

Fig. 8. Left is the 2D grid map and OctoMap of a corridor, right is the real corridor

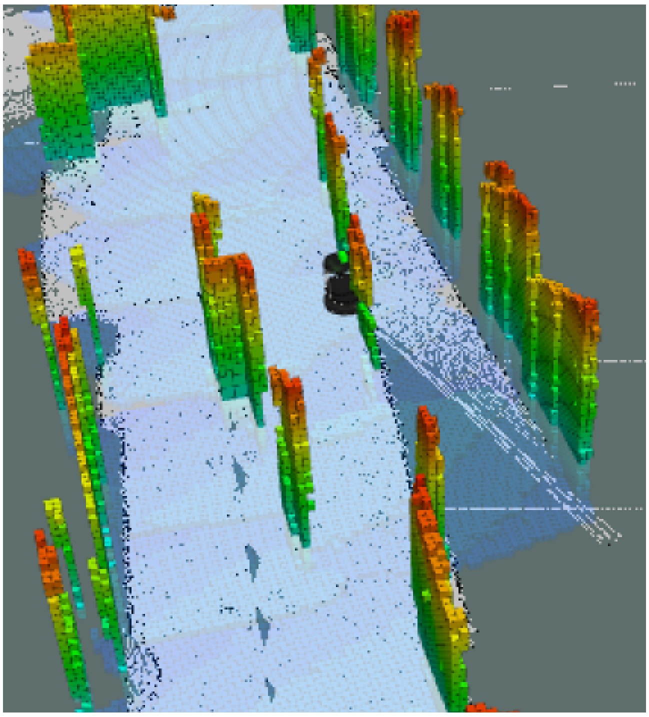

Fig. 9. An error map causes the robot hits the wall

VI. Conclusions

This paper combines laser scanner with depth camera to build 2D grid map and OctoMap simultaneously. SLAM with laser scanner has high precise localizing information, while the depth camera can get more characteristics of objects compared with the laser scanner in the 3D environment. Moreover, while 2D grid maps can be used in path planning of mobile robot on the floor, while the OctoMap can be used in the collision detection of the robot arm, and path planning in 3D environment. In our future work, we plan to combine the two maps for path planning and the collision detection for manipulators equipped on mobile robots.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant (61175073, 61300159, 61332002, 51375287) , the Guangdong Key Laboratory of Digital signal and Image Processing, the Science and Technology Planning Project of Guangdong Province (2013B011304002) and the Project of Educational Commission of Guangdong Province, China 2015KGJHZ014).

References

[1] B. L. E. A. Balasuriya, B. A. H. Chathuranga, B. H. Jayasundara, N. R. A. C. Napagoda, S. P. Kumarawadu, D. P. Chandima, and A. G. B. P. Jayasekara, "Outdoor robot navigation using gmapping based slam algorithm," *Moratuwa Engineering Research Conference*, pp. 403-408, 2016.

[2] R. Mur-Artal, J. M. M. Montiel, and J. D. Tards, "ORBSLAM: A versatile and accurate monocular slam system," *Moratuwa IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147-1163, 2015.

[3] J. Engel, T. Schops, and D. Cremers, "LSD-SLAM: Largescale direct monocular slam," *Computer Vision-ECCV 2014*, pp. 834-849, 2014.

[4] J. W. Li, D. F. Zheng, Z. H. Guan, C. Y. Chen, X. W. Jiang, and X. H. Zhang, "Indoor 3d scene reconstruction for mobile robots using microsoft kinect sensor" *Chinese Control Conference*, pp. 6324-6328, 2016.

[5] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, and A. Davison, "Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera," *ACM Symposium on User Interface Software and Technology*, pp. 559-568, 2011.

[6] C. Kerl, J. Sturm, and D. Cremers, "Dense visual slam for rgbd cameras," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2100-2106, 2014.

[7] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots, 2013*, [Online]. Available: http://octomap.github.com.

[8] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Transactions on Robotics*, pp. 34-46, 2007.

[9] S. Thrun, "Probabilistic robotics," *MIT Press*, 2006.

[10] K. K. D. K. U. Weerasinghe, L. C. J. Silva, B. M. S. S. Basnayake, S. D. M. Sandanayaka, S. P. Kumarawadu, D. P. Chandima, and A. G. B. P. Jayasekara, "Mapping and path planning for long distance autonomous navigation using multisensory data," *Electrical Engineering Conference*, pp. 1-6, 2016.

[11] open source, "Gazebo online tutorial," 2013. [Online]. Available: http://gazebosim.org/.

[12] Q. Lv, H. Lin, G. Wang, H. Wei, and Y. Wang, "ORB-SLAMbased tracing and 3d reconstruction for robot using kinect 2.0," *Control and Decision Conference*, pp. 3319-3324, 2017.

[13] K. Kamarudin, S. M. Mamduh, A. S. A. Yeon, R. Visvanathan, A. Y. M. Shakaff, A. Zakaria, L. M. Kamarudin, and N. A. Rahim, "Improving performance of 2d slam methods by complementing kinect with laser scanner," *IEEE International Symposium on Robotics and Intelligent Sensors*, pp. 278-283, 2016.

[14] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source robot operating system," 2009.

[15] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, "Monte carlo localization: Efficient position estimation for mobile robots," *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, pp. 343-349, 1999.

[16] M. Montemerlo, S. Thrun, D. Roller, and B. Wegbreit, "Fastslam 2.0: an improved particle filtering algorithm for simultaneous localization and mapping that provably converges," *International Joint Conference on Artificial Intelligence*, pp. 1151-1156, 2003.

[17] M. Bolic, P. M. Djuric, and S. Hong, "Resampling algorithms for particle filters: A computational complexity perspective," *EURASIP Journal on Advances in Signal Processing*, vol. 2004, no. 15, pp. 1-11, 2004.