Proceedings of CECNet 2021
A.J. Tallón-Ballesteros (Ed.)

doi:10.3233/FAIA210433

Incremental 2D Grid Map Generation from RGB-D Images

Tingfeng YE [a,b], Juzhong ZHANG [b, 1], Yingcai WAN [c], Ze CUI [a] and Hongbo YANG [b]
[a] *School of Mechatronic Engineering and Automation, Shanghai University, Shanghai, China*
[b] *Suzhou Institute of Biomedical Engineering and Technology, Chinese Academy of Science, Suzhou, China*
[c] *Dept. Faculty of Robot Science and engineering, Northeastern University, Shenyang, China.*

Abstract. In this paper, we extend RGB-D SLAM to address the problem that sparse map-building RGB-D SLAM cannot directly generate maps for indoor navigation and propose a SLAM system for fast generation of indoor planar maps. The system uses RGBD images to generate positional information while converting the corresponding RGBD images into 2D planar lasers for 2D grid navigation map reconstruction of indoor scenes under the condition of limited computational resources, solving the problem that the sparse point cloud maps generated by RGB-D SLAM cannot be directly used for navigation. Meanwhile, the pose information provided by RGB-D SLAM and scan matching respectively is fused to obtain a more accurate and robust pose, which improves the accuracy of map building. Furthermore, we demonstrate the function of the proposed system on the ICL indoor dataset and evaluate the performance of different RGB-D SLAM. The method proposed in this paper can be generalized to RGB-D SLAM algorithms, and the accuracy of map building will be further improved with the development of RGB-D SLAM algorithms.

Keywords. RGB-D SLAM, layout SLAM, pose fusion, navigation, 2D grid maps

1. Introduction

SLAM (Simultaneous localization and mapping) is used to estimate the camera pose and reconstruct unknown environments, which is currently widely used in various fields, relying on sensors to achieve functions such as autonomous positioning, mapping, and path planning of the machine. For example, automatic navigation of robots, unmanned driving of cars, AR/VR technology positioning and three-dimensional reconstruction of objects.

At present, most of the lidar SLAM used to generate indoor navigation maps use lidar as the main sensor. For example, Gmapping[1], Hector SLAM[2], and Cartographer[3] can use single-line lidar to generate more accurate indoor navigation grid maps, but can only explore the environment in two dimensions. At the same time, single-line lidar equipment is large, limited by the fact that it must use a mechanical

[1] Corresponding author: Juzhong Zhang, Suzhou Institute of Biomedical Engineering and Technology Chinese Academy of Science, Suzhou, China; E-mail: jzzhang@sibet.ac.cn

motor to complete the rotation to obtain the exploration of the unknown environment. And the data obtained is with distortion [1, 2], which requires its matching SLAM algorithm to perform pre-processing to remove distortion from the data, resulting in reduced real-time and accuracy of the algorithm operation. There are also LOAM[4] which using multi-line lidar, and V-LOAM[5], LeGo-LOAM[6] based on [4] combined with other sensors and improved, which obtains spatial 3D information, also brings the problem of high cost [7] and excessive computational effort. Loam_livox [7] which uses solid-state lidar as the sensor that characteristics of laser radar have brought major challenges to the navigation and mapping of lidar. The above SLAM algorithms that use lidar as the main sensor are all limited by the defects of the laser sensor itself.

In this environment, the RGB-D sensor provides an opportunity to significantly develop the robot's indoor navigation and interaction capabilities [8]. At present, for SLAM algorithms, RGB-D cameras have been introduced and three-dimensional maps can be created in real-time, and a variety of different RGB-D SLAM algorithms have been proposed. Most of these RGB-D SLAM are used for indoor localization and object dense reconstruction [9]. Mono SLAM [10] and ORB-SLAM2 [11] based on the feature point method can directly obtain the camera's pose in space and the sparse point cloud map, but the obtained map cannot be directly used for navigation. DVO-SLAM [12] based on the direct method estimates the motion of the camera according to the gradient information of the pixels, and can construct a dense map. S-SLAM [13] and Planar-SLAM [14] based on characteristic lines and surfaces can handle low-texture, structured indoor scenes. Again, the maps created by SLAM mentioned above do not have the capability to be used for navigation.

In this paper, we exploit the ability of RGB-D SLAM to output depth maps as well as camera poses in real time, combined with algorithms for navigation grid map building of 2D LiDAR SLAM, to design a robust RGB-D SLAM processing system specifically for building maps of indoor environments. We first obtain the depth map and the camera pose corresponding to the image information through RGB-D SLAM, then convert the depth map into laser information, and pass the camera pose as the predicted pose to SLAM to obtain the indoor two-dimensional grid map. The grid map is different from [1][2][3]. We pass the camera pose obtained by RGB-D SLAM as the predicted pose into SLAM, which is a more robust prediction. At the same time, we use the loop closure detection provided by itself to improve the real-time performance of the algorithm. In terms of equipment hardware, the system we propose can run in real-time on depth-sensing equipment, reducing the size and cost of the equipment. We have filled the functional gap of sparse RGB-D SLAM in the field of navigation by significantly reducing the computational effort. At last, we evaluated the mapping performance of the two-dimensional grid map obtained by the system using different RGB-D SLAM in the ICL dataset and showed the stability of our system in different RGB-D systems. With the improvement of RGB-D SLAM, there are better map building results.

2. Related Work

There are many related SLAM algorithms that can be used for indoor navigation, which can be roughly divided into two categories, visual SLAM and lidar SLAM. We will mainly summarize the RGB-D SLAM systems and the two-dimensional laser algorithm used in laser SLAM methods, respectively.

2.1. Two-dimensional laser SLAM

Two-dimensional laser SLAM is relatively mature in theory and practice and has been widely used in scientific research and industrial fields. Gmapping is based on the Fast-SLAM [15] scheme and is equipped with a two-dimensional algorithm based on particle filtering of the lidar sensor. The core idea of particle filtering is to randomly sample and estimate the map through selective resampling of particles [16]. Its main contribution is to improve the proposal distribution and selective resampling. However, due to the lack of closed-loop detection, it can only be used in simple, Maintain reliable accuracy in low-feature indoor environments. [17] proposed the corrective gradient refinement algorithm, which is a new method to improve the positioning based on the particle filter, which extends the traditional particle filter algorithm and is more universal. [18] proposed an improved algorithm based on information fusion, combining the odometer and inertial measurement unit, using Kalman filter for information fusion, assisting the robot in mapping, and improving the robot's positioning and mapping performance in degraded environments. Cartographer [3] based on graph optimization is different from [2] whose back-end adopts a closed-loop detection link based on branch and bound method. Cartographer can eliminate errors in robot motion and map large-area scenes. However, its optimization is more computationally intensive, and its real-time performance cannot be guaranteed.

2.2. RGB-D SLAM

Visual SLAM can be divided into sparse and dense maps according to the characteristics of mapping. Semi-dense and dense map can be directly used for indoor navigation maps, while sparse map cannot provide enough information for robots to navigate. Kinect Fusion [19] achieves real-time operation on the GPU. The system uses voxel grids to build maps, and does not restrict the cumulative error generated by the motion, so the application range is small, and when the environment is mainly composed of parallel planes, ICP Will fail. [20] proposed the Kintinuous system and added loop detection to eliminate accumulated errors, which improved the space expansion on the basis of [19]. Different from Kintinuou, ElasticFusion [21] uses a direct optimization method for map points in order to improve accuracy, and densely reconstructs and repositions the three-dimensional environment through the surfel model, making full use of color and depth information, but the scope of the map is However, it is only suitable for reconstruction of room-sized scenes. [11] improved on the basis of ORB-SLAM [22], adding two modes of binocular SLAM and RGB-D SLAM. [11] adopts monocular and binocular beam adjustment optimization (BA), which improves the accuracy but can only create relatively simple 3D point clouds. Planer-SLAM [14] can handle low-structure, textured indoor scenes, and generate indoor three-dimensional plane environments from sparse point clouds. Dense RGBD-SLAM reconstruction can achieve good pose estimation and high-quality scene representation, but this requires high computational cost and complex equipment, such as high-performance GPU, which limits the platform used in different devices.

3. System overview

The schematic diagram of our proposed system is shown in Figure 1. Our work is inspired by RGB-D SLAM that generates sparse point cloud maps, such as ORB-SLAM 2 [11]. The above mentioned RGB-D SLAM can operate in an indoor environment and is also robust to strenuous exercise. However, the point cloud map generated by this system is sparse and cannot be directly used for indoor navigation. We can use the depth information in RGB-D SLAM, convert it into lidar data (See next subsection), and add its output pose information to build an indoor two-dimensional grid map. Our system first uses the pose provided by RGB-D SLAM as the initial pose, and then predicts the next new pose based on scan matching. These two poses obtain a new estimated pose through pose fusion. Insert the converted lidar data from the corresponding depth map into the final pose. After inserting a certain number of frames of lidar information, a submap is generated. We refer to the processing algorithm in [3], and on this basis, combined with the loop detection that comes with RGB-D SLAM, which reduces the cumulative error of previous submaps and improves the accuracy of mapping

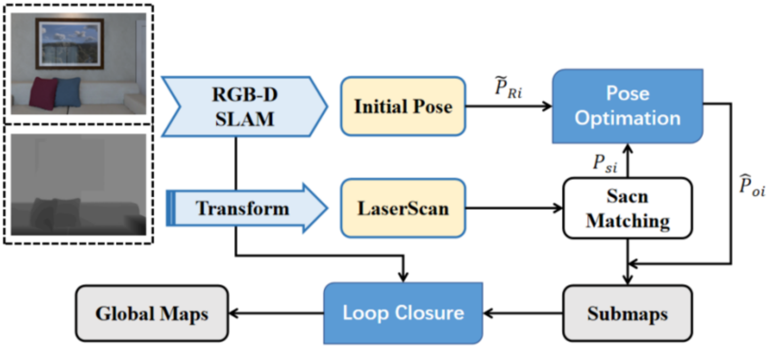

Figure 1. Main system pipeline

3.1. Depth map converted to laser scan

We first convert each pixel in the depth image (commonly the pixel in the middle of the image) into laser data. Refer to the pinhole camera model, the conversion matrix from a point M(x, y, z) in the world coordinate system to a pixel point m(u, v) in the camera coordinate system can be derived from a camera internal parameter matrix and an external parameter matrix, as shown in Equation 1.

$$z_C \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{bmatrix} f/_{dx} & 0 & u_0 \\ 0 & f/_{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} [R\ T] = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{1}$$

where u, v are any coordinate points under the pixel coordinate system, u_0 , v_0 are the central coordinates of the image. x, y, z represent the 3D coordinate points under the world coordinate system. z_c represents the z-axis value of the camera coordinates, the distance from the target to the camera. R, T are denoted as the unit matrices of 3×3 rotation and 3×1 translation in the external reference matrix respectively. Since the ICL simulation set and the camera parameters it provides are used, distortions can be disregarded. We then project the point cloud data obtained above according to Equation 3 based on the pose information obtained from RGB-D SLAM, thus converting a point cloud derived from non-horizontal depth map information into a horizontal frame of point cloud data. We finally obtain the coordinates of the laser point cloud as follow:

$$\begin{cases} x = q_{pose}(z_C \cdot (u - u_0) \cdot dx \,/\, f) \cdot q_{pose}^{-1} \\ y = q_{pose}(z_C \cdot (v - v_0) \cdot dy \,/\, f) \cdot q_{pose}^{-1} \\ \quad z = q_{pose} \cdot z_C \cdot q_{pose}^{-1} \end{cases} \tag{2}$$

where q_{pose} represents the pose information obtained by RGB-D SLAM.

3.2. Pose Optimization

Our system undergoes a pose optimization to process the initial pose from RGB-D SLAM and obtain the final pose. This optimization step starts with the calculation of a new pose by scan matching between the current frame of the LIDAR and the submaps, then the two poses are fused by Kalman filtering to obtain a more robust pose:

$$\hat{P}_{Oi} = \tilde{P}_{Ri} + K(P_{Si} - C_i\tilde{P}_{Ri}) \tag{3}$$

where $\hat{P}_{Oi}$ represents the fused poses, $\tilde{P}_{Ri}$ represents the predicted poses obtained from RGB-D SLAM, P_{Si} represents the observed poses obtained from the scan matching, Kalman gain K depends on the noise of the sensor, and the higher the camera noise, the higher the value of K.

Referring to the scan matcher in [3], scan matching works on the principle of finding the optimal probability value of the scanned points in a grid-based submap. In processing laser data and submap matching, a violent search matching algorithm is used to match the current laser frame traversing the historical grid map to obtain the position with the highest correlation, i.e., the position with the highest confidence. We subjected the grid map to bicubic interpolation and then converted the laser frame to map correlation matching to the least squares problem as follows:

$$\underset{f}{\operatorname{argmin}} \sum_{k=1}^{K} \left(1 - M_s(T_f h_k)\right)^2 \tag{4}$$

where h_k is the laser frame transformed into the submap by T_f and M_s is the probability-valued bicubic interpolation smoothing filter in the submap. The mathematical optimization of this smoothing function is usually higher resolution than the grid and has better accuracy. However, because we use local optimization, a good initial pose is required that using the pose given by RGB-D SLAM will be a good choice.

3.3. Loop Closure Detection

The map created by SLAM is divided into two main parts. The first part is a local landmark map called a submap, which is made up of a certain number of laser data. The second part is a global map built from a combination of accumulated submaps. The map building process as described above accumulates errors, which are small for a few tens of consecutive laser data, but not negligible when building larger scenes or rotating scans of an interior. Laser SLAM requires a lot of computation to do loop closure in large map.

The approach taken in this paper is that the global map is corrected when loop closure is identified during the RGB-D SLAM, optimizing the pose of all scans and submaps and reducing the accumulated error in the global map. We directly use the loop closure information provided by RGB-D SLAM, making full use of that excellent loop closure algorithm to optimize the global pose. The global map optimization problem can be formulated as a non-linear least squares problem, using Ceres [23] to calculate problem as follow:

$$\underset{\Xi^m,\Xi^s}{\operatorname{argmin}} \frac{1}{2}\sum_{ij} \rho(E^2(\xi_i^m,\xi_j^s;\Sigma_{ij},\xi_{ij})) \tag{5}$$

where the submap poses are $\Xi^m = \{\xi_i^m\}_{i=1,\ldots,m}$ and the poses are $\Xi^s = \{\xi_j^s\}_{j=1,\ldots,n}$ for each frame of laser data are optimized in the world coordinate system. These optimized submaps poses and Scan poses give constraints, which are expressed in terms of the poses ξ_{ij} and the covariance matrix Σ_{ij}. The residuals E are calculated as follows:

$$\begin{gathered} E^2\left(\xi_i^m,\xi_j^s;\Sigma_{ij},\xi_{ij}\right)=e(\xi_i^m,\xi_j^s;\xi_{ij})^T\,\Sigma_{ij}^{-1}\,e\left(\xi_i^m,\xi_j^s;\xi_{ij}\right), \\ e\left(\xi_i^m,\xi_j^s;\xi_{ij}\right)=\xi_{ij}-\begin{pmatrix} R_{\xi_i^m}^{-1}\left(t_{\xi_i^m}-t_{\xi_j^s}\right) \\ \xi_{i;\theta}^m-\xi_{j;\theta}^s \end{pmatrix}. \end{gathered} \tag{6}$$

where the loss function, ρ (e.g. Huber loss), can be used to reduce the impact of outliers, which may occur in similar environments.

4. Experiment Results

In order to evaluate the performance of our system, in this section we will validate our proposed system using different RGB-D SLAM in two datasets. The first test uses the Living Room dataset, and the second test uses the Office Room dataset. There are open public datasets developed by Imperial College London [24]. 2D grid maps were generated on the ICL-NUIM dataset using different RGB-D SLAM. This paper also tests the performance of the system through trajectory comparison experiments. As shown in Figure 2, the RGB-D SLAM generates corresponding pose information based on the depth image and color image provided by ICL-NUIM, and the system uses the corresponding depth map from the dataset, converts it into a laser data and generates a 2D planar grid map. All experiments were conducted using an Intel Core i5-5200 (with @2.20GHz) and without any use of GPU.

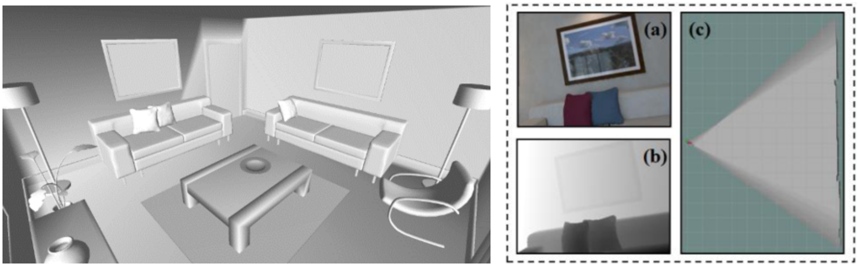

Figure 2. The image on the left shows the interior of the Living Room scene from the ICL dataset (the colors have been removed to highlight the geometry). (a) is a color frame of the Living Room. (b)is the depth frame of the Living Room at the same moment. (c) is the 2D grid map being generated.

4.1. Two-dimensional Grid Map

We use the ICL-NUIM Living Room dataset and the Office Room through different RGB-D SLAM respectively. ORB-SLAM2, Planer-SLAM and Planer-SLAM (with HM) were used for the experiments, and the ground truth data from the ICL-NUIM dataset were also used to generate 2D grid maps and trajectory maps as a reference. Figure 3 shows the experimental results of our system.

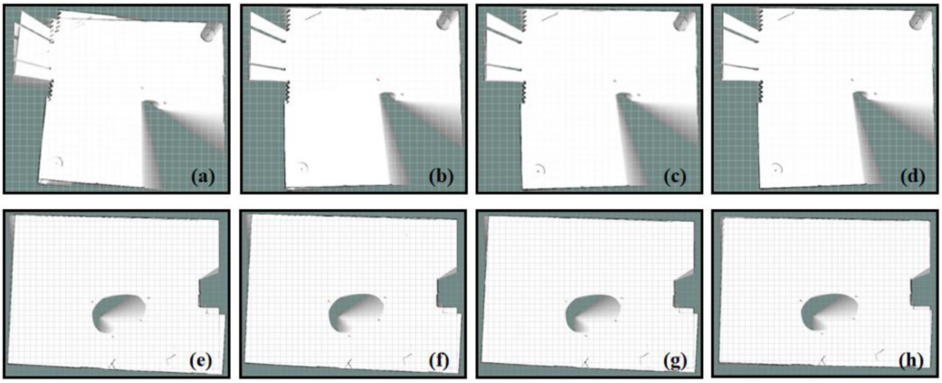

Figure 3. Results obtained using different RGB_D SLAM in the ICL dataset: the first row is Living Room：(a) ORB_SLAM2 (b) Planar-SLAM (c) Planar-SLAM with Manhattan (d) ground truth; the second row is Office Room: (e) ORB_ SLAM2 (f) Planar-SLAM (g) Planar-SLAM with Manhattan (h) ground truth.

As show in Figure 3, we see that with fewer feature points in the image data, the ORB-SLAM2 outputs less accurate poses resulting in fewer effective maps being built than Planar-SLAM or Planar-SLAM (with HM), which can improve the accuracy of the poses by detecting line and surface features. In the next section, we explain quantitatively by using the experimental data from the trajectory comparison in next section.

4.2. Trajectory curves

We obtained eight different trajectory curves from three different RGB-D SLAM and ground truth on the two datasets, as illustrated in Figure 4. We also calculated the average Euclidean distance of other trajectories compared with the ground truth to judge the similarity, and the results are shown in Table 1.

Table 1. Average Euclidean distances of trajectories obtained with different RGB-D SLAM

RGB-D SLAM	Living Room	Office Room
ORB-SLAM 2	0.407	0.454(m)
Planer SLAM	0.309	0.364(m)
Planer SLAM with MH	0.288	0.245(m)

As can be seen from the table, as the performance of the RGB-D SLAM positioning performance gets better, the closer the trajectories obtained are to the true values and the more similar the resulting maps are to the ground truth.

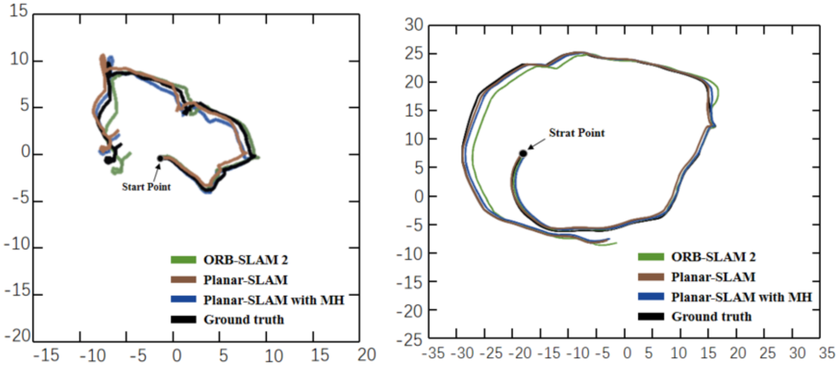

Figure 4. Experimental paths (unit meter) in Living Room (left) and Office Room (right) using different RGB_D SLAM: ORB_SLAM2 (green), Planar-SLAM (brown), Planar-SLAM with Manhattan (blue), ground truth (black).

5. Conclusion

We present a method for rapidly generating indoor 2D grid navigation maps by combining RGB-D SLAM and laser SLAM systems. The robust localization algorithm of RGB-D SLAM is utilized to determine the camera's poses and loops, and a depth map to laser data method is used to enable the depth map information to be used for laser SLAM, achieving advanced results by fusing pose. The lightness of the sensor and the fact that the algorithm does not require the use of large computational performance allows us to use the system for a robot with a depth sensor, using an IPC for indoor navigation of the robot. As a next step, we hope to make full use of the 3D information provided by the depth camera to generate indoor 3D navigation maps, suitable for complex indoor environments.

References

[1] Grisetti G , Stachniss C , Burgard W . Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters[J]. IEEE Transactions on Robotics, 2007, 23:p.34-46.

[2] Kohlbrecher S, Von Stryk O, Meyer J, et al. A flexible and scalable SLAM system with full 3D motion estimation[C]//2011 IEEE international symposium on safety, security, and rescue robotics. IEEE, 2011: 155-160.

[3] Hess W, Kohler D, Rapp H, et al. Real-time loop closure in 2D LIDAR SLAM[C]//2016 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2016: 1271-1278.

[4] Zhang J, Singh S. LOAM: Lidar Odometry and Mapping in Real-time[C]//Robotics: Science and Systems. 2014, 2(9).

[5] Zhang J, Singh S. Visual-lidar odometry and mapping: Low-drift, robust, and fast[C]//2015 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2015: 2174-2181.

[6] Shan T, Englot B. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain[C]//2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018: 4758-4765.

[7] Lin J, Zhang F. Loam livox: A fast, robust, high-precision LiDAR odometry and mapping package for LiDARs of small FoV[C]//2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2020: 3126-3131.

[8] ElGhor H E C, Roussel D, Ababsa F, et al. 3D Planar RGB-D SLAM System[C]//International Conference on Advanced Concepts for Intelligent Vision Systems. Springer, Cham, 2016: 486-497.

[9] Zhang, Shishun, Longyu Zheng, and Wenbing Tao. "Survey and Evaluation of RGB-D SLAM." IEEE Access 9 (2021): 21367-21387.

[10] Davison A J, Reid I D, Molton N D, et al. MonoSLAM: Real-time single camera SLAM[J]. IEEE transactions on pattern analysis and machine intelligence, 2007, 29(6): 1052-1067.

[11] Mur-Artal R, Tardós J D. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras[J]. IEEE transactions on robotics, 2017, 33(5): 1255-1262.

[12] Kerl C, Sturm J, Cremers D. Robust odometry estimation for RGB-D cameras[C]//2013 IEEE international conference on robotics and automation. IEEE, 2013: 3748-3754.

[13] Li Y, Brasch N, Wang Y, et al. Structure-slam: Low-drift monocular slam in indoor environments[J]. IEEE Robotics and Automation Letters, 2020, 5(4): 6583-6590.

[14] Li Y, Yunus R, Brasch N, et al. RGB-D SLAM with Structural Regularities[J]. arXiv preprint arXiv:2010.07997, 2020.

[15] Stentz A , Fox D , Montemerlo M , et al. FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association[J]. Proc.aaai Natl.conf.on Artificial Intelligence Edmonton Alberta, 2002, 50(2):240-248.

[16] Konolige K, Grisetti G, Kümmerle R, et al. Efficient sparse pose adjustment for 2D mapping[C]//2010 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2010: 22-29.

[17] Wongsuwan K, Sukvichai K. Generalizing corrective gradient refinement in RBPF for occupancy grid LIDAR SLAM[C]//2017 IEEE International Conference on Robotics and Biomimetics (ROBIO). IEEE, 2017: 495-500.

[18] Yu N, Zhang B. An improved hector SLAM algorithm based on information fusion for mobile robot[C]//2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS). IEEE, 2018: 279-284.

[19] Newcombe R A, Izadi S, Hilliges O, et al. Kinectfusion: Real-time dense surface mapping and tracking[C]//2011 10th IEEE international symposium on mixed and augmented reality. IEEE, 2011: 127-136.

[20] Whelan T, Johannsson H, Kaess M, et al. Robust real-time visual odometry for dense RGB-D mapping[C]//2013 IEEE International Conference on Robotics and Automation. IEEE, 2013: 5724-5731.

[21] Whelan T, Salas-Moreno R F, Glocker B, et al. ElasticFusion: Real-time dense SLAM and light source estimation[J]. The International Journal of Robotics Research, 2016, 35(14): 1697-1716.

[22] Mur-Artal R, Montiel J M M, Tardos J D. ORB-SLAM: a versatile and accurate monocular SLAM system[J]. IEEE transactions on robotics, 2015, 31(5): 1147-1163.

[23] S. Agarwal, K. Mierle, and Others, “Ceres solver,” http://ceres-solver.org.

[24] Handa A, Whelan T, McDonald J, et al. A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM[C]//2014 IEEE international conference on Robotics and automation (ICRA). IEEE, 2014: 1524-1531.