2017 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration

3D mapping of Multi-Floor Buildings based on sensor fusion

Zhun Fan, Chong Li, Yupeng Wang, Lei Zhao, Li Yao, Guijie Zhu, Zhongxing Li, Honghui Xie, Yang Xiao
Department of Electronic Engineering
Shantou University
Guangdong, ShanTou,515063
Email: zfan@stu.edu.cn

***Abstract*—In this paper, we design a highly accurate method for 3D mapping of multi-floor buildings. The basic idea is to combine the laser range sensor for metric mapping and barometric pressure sensor for detecting floor transition and kinect sensor for collection 3D environment information. Meanwhile, we adopt the Monte Carlo localization in 2D map to improve the accuracy of the localization. Finally, the barometric pressure is used to merge the 3D map of the multi-floor buildings. By using the information collected by a real robot in a typical multi-floor environment, the method is tested and compared with some other approaches, the results show that the method is efficient and has a better result in 3D mapping of multi-floor buildings.**

***Keywords*-multi-floor buildings; barometric pressure; kinect; visual odometry; Monte Carlo Localization**

I. Introduction

In the last two decade, the problem of the Simultaneous Localization and Mapping (SLAM) has been one hot research direction, and attracted the attention of high-technological companies [1] [2]. The SLAM techniques build a 2D mapping or 3D mapping of an unknown environment and localize the robot in the map with a strong real-time operation. In the complex and dynamic environments, the robot can quickly obtain the 3D map from the local environment. Currently, many methods have been proposed to solve the SLAM problem, such as ORB-SLAM [2], RGBD-SLAM [3].

The idea of the paper is that a robot uses a kinect, a lidar and odometry and then fuse them with Monte Carlo localization (MCL) algorithm. The approach can avoid the process of mismatching and image feature extraction. Firstly, we build the 2D map with the lidar and Gmapping package [4] [5], and the robot can focus on the real-time localization and mapping. Then we use the robot poses in the process of the navigation that we get at different time affording by the MCL. Thus, we can get the pose transition matrices by calculating the these poses which are used for 3D point cloud registration [6]. Finally, we build the 3D map, through a series of position transformation. We mainly adopt the value of the barometric pressure to merge the whole floors by the point cloud registration with the transition matrices onto a global coordinate frame physically consistent multi-floor representation.

In the rest of the paper, we introduce the related work in section II. In section III, we present the implementation method. In section IV, we use the some experiments to verify the idea and compare with some state-of-art algorithms. The last section summarize the our work.

II. Related work

Building map of the indoor environment is a hot research topic. Currently, many solutions exist in 2D maps and methods to extend them to 3D are emerging [5]. In this paper, we use the (metric) occupancy maps in the 2D maps [7]. Wheeled robots often rely on 2D laser range scanners, which commonly provide very accurate geometric measurements of the environment at high frequencies. To compute the relative motion between observations, most state-of-the-art SLAM use variants of the ICP algorithm [3]. Recent approaches demonstrated that even small changes of the robot pose can be estimated [8] using two laser range scanners and ICP.

Though there are many methods of solution for 2D mapping and 3D mapping, only a few approaches are able to deal with the multi-floor buildings. A visual odometry based approach is proposed to analyze the alignments of floors [9]. In this approach, however, the floors should be mapped sequentially; and salient visual features are required during floor transitions. In [10], the author proposed to use a barometer pressure value for mobile robot localization in outdoor environment. This method use differential Global Positioning System (GPS) receiver and barometric sensor. The pressure value is applied to estimate the altitude in some areas where the GPS can not get the accurate information.

In this paper, we present a method different from the above mentioned ideas. We use the Monte Carlo Localization (MCL) algorithm in the process of the navigation which can provide a higher localization accuracy. Meanwhile, we adopt the rotation matrices of the different poses to generate 3D mapping when the robot moves from the current position to the next position. After building the map of each floor, we can merge the maps and build the 3D map of the multi-floor buildings.

III. Method

We use a laser range sensor to generate 2D maps of floors and estimate pose transformation using a MCL algorithm for

978-1-5386-2434-0/17 $31.00 © 2017 IEEE
DOI 10.1109/ICIICII.2017.10

kinect, which is applied to 3D point cloud registration. At the same time, we use barometric pressure sensor to detect the floor transitions. Finally, we get the 3D mapping of multi-floor buildings based on sensors fusion.

A. Mapping of Individual Floors

In this paper, based on the state-of-art-algorithms that address 2D mapping problem, we use the Gmapping package to build our 2D maps. Gmapping is a package of the Robot Operation System (ROS) [4] [11] [12].

We use the Rao-Blackwellized particle filters (RBPF) algorithm to generate the 2D map from the lidar range data [13]. The key contribution of the RBPF is to double check the joint posterior for the path of the mobile robot $x_{1:t} = x_1 \ldots, x_t$ and the map m,based on the lidar observations $z_{1:t} = z_1 \ldots, z_t$ and the wheeled odometry measurements $u_{1:t} = u_1 \ldots, u_t$. We use the following equation to present the joint posterior.

$$p(x_{1:t}, m \mid z_{1:t}, u_{1:t-1}) = p(m \mid x_{1:t}, z_{1:t}) \cdot p(x_{1:t} \mid z_{1:t}, u_{1:t-1}) \tag{1}$$

According to this equation, we use the probability $p(x_{1:t} \mid z_{1:t}, u_{1:t-1})$ to estimate the path of the robot, which can build the map $p(m \mid x_{1:t}, z_{1:t})$.

B. Monte Carlo Localization (MCL)

MCL is applicable to both local and global localization problem in robotics. Through this algorithm, we can get the robot position in the 2D map. Both the robot motion model $p(x_t \mid x_t - 1, u_t - 1)$ and measurement model $p(z_t \mid x_t)$ are applied to the prediction stage and the update stage of MCL [14].

In this paper, we use the KLD-sampling which is a variant of the MCL. It is derived from the Kullback-Leibler divergence, which is a measure of the difference between two probability distributions. At each iteration of the particle filter, KLD-sampling determines the number of samples such that, with probability $1 - \delta$, the error between the true posterior, and the sample-based approximation is less than ε . The algorithm takes as input the previous sample set χ_{t-1} along with the map m and the most recent control u_t and measurement z_t. For the implementation of MCL, we refer to [6] [14].

C. 3D Map Building

In this section, we use the kinect sensor to collect the color and depth image where the robot pose x_t at the time t in the 2D map can be obtained. At the same time, through wheeled odometry and lidar range, KLD-sampling MCL obtains control information and measurement information to locate the robot in the 2D map. According to this method, we obtain the pose transformation between the current and last moment, which is used for the kinect, by the following equation:

$$x_t = R \cdot x_{t-1} + t \tag{2}$$

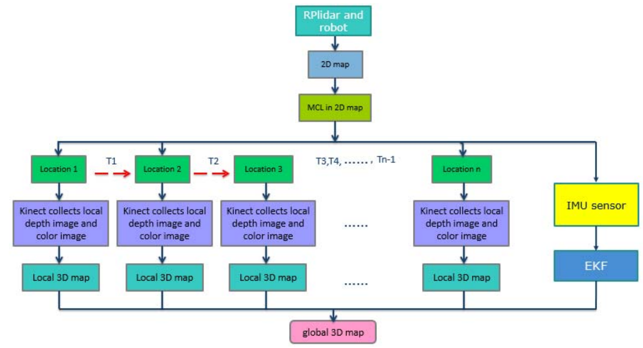

Figure 1. The 3D map building

Where R is a rotation matrix and t is a translation vector.

The kinect uses the above information to get the images at different moments and to build a 3D map which is generated by the point cloud registration. A general overview of the 3D map building is shown in Figure 1. Then we introduce the transformation from 2D image to 3D point cloud by using the following equations.

$$z = d/s \tag{3}$$

$$x = (u - c_x) \cdot z/f_x \tag{4}$$

$$y = (v - c_y) \cdot z/f_y \tag{5}$$

where c_x, c_y, f_x, f_y are the kinect internal parameter. d and s, are depth information and scaling factor, respectively. u and v are the coordinate of the image pixels p_t. x, y, z are the position of the 3D point cloud. Once the 3D point clouds at the different poses are estimated, we start to implement point cloud registration for the p_{t-1} and p_t by the transform matrix:

$$T = \begin{pmatrix} R_{3\times3} & t_{3\times1} \\ 0_{1\times3} & 1 \end{pmatrix} \in R^{4\times4} \tag{6}$$

D. Extended Kalman Filter and Barometric Pressure

In this section, we first use the barometric pressure sensor to detect the pressure value of each floor. Due to the instability of the barometric pressure values, we then use the kalman filter to calibrate the value. Through the corrected pressure values, we calculate the altitude of the each floor. Finally, we get the height difference of any two floors.

1) Extended Kalman Filter: The Kalman filter was invented in the 1950s by Rudolph Emil Kalman as a technique for filtering and prediction in linear Gaussian system [14]. Unfortunately, state transitions and measurements are rarely linear in practice. Then the Extended Kalman filter (EKF) , is proposed to relax the linearity assumption. The assumption is that the state transition probability and the measurement probabilities are governed by nonlinear functions g and h , respectively:

$$x_t = g(u_t, x_{t-1}) + \varepsilon_t \tag{7}$$

$$z_t = h(x_t) + \delta_t \quad (8)$$

Here x_t and x_{t-1} are state vector, and u_t is the control vector at time t. The random variable ε_t is a Gaussian random vector that models the uncertainty introduced by the state transition, with zero mean and covariance denoted as R_t. z_t is the measurement vector, and vector δ_t describes the measurement noise. The distribution of δ_t is a multivariate Gaussian with zero mean and covariance Q_t.

EKF utilizes a method called (first order) Taylor expansion. The Taylor expansion constructs a linear approximate to a function g and h from $g^{'}s(G_t)$ and $h^{'}s(H_t)$ value and slope. The EKF algorithm are defined by the following equations [14]:

$$\bar{u_t} = g(u_t, u_{t-1}) \quad (9)$$

$$\bar{\Sigma_t} = G_t \Sigma_{t-1} G_t^T + R_t \quad (10)$$

$$K_t = \bar{\Sigma_t} H_t^T (H_t \bar{\Sigma_t} H_t^T + Q_t)^{-1} \quad (11)$$

$$u_t = \bar{u_t} + K_t(z_t - h(\bar{u_t})) \quad (12)$$

$$\Sigma_t = (I - K_t H_t)\bar{\Sigma_t} \quad (13)$$

EKF represents the belief $bel(x_t)$ (A belief reflects the robot's internal knowledge about the state of environment.) at time t by the mean u_t and covariance Σ_t , K_t is the EKF gain.

2) Barometric Pressure: Barometric pressure is defined as the force per unit area exerted against by the weight of air above that surface in the earth's atmosphere[5]. With the increasing altitude, the barometric pressure will decrease. We can express this relation by using the following equation:

$$\frac{dp}{dh} = \rho g \quad (14)$$

Where p is pressure, ρ is the density of air, h is the altitude and g is the gravity. As we know, there is also a relation of the pressure and temperature.

$$p = \rho K M \quad (15)$$

Where K is the Boltzmann constant and M is the temperature. We know the pressure of zero altitude is obtained from mean sea level as:

$$P_0 = 101325pa \quad (16)$$

Using the above the equations, we have the altitude h:

$$h = -\frac{KM}{g}\ln(\frac{p}{P_0}) \quad (17)$$

Figure 2. The Guangdong provincial Key Laboratory of Digital Signal and Image Processing, Science and Technology Building, Shantou University. The building has 6 floors.

E. 3D Mapping of Multi-Floor Building

Based on all the above methods, we start to merge maps. First, we use the corrected pressure values to calculate the altitude each floor. Then, we get the differences of height between of any two floors. The difference in height plays an important role in the point cloud registration in a process of point cloud transformation by the transform matrix T, which is used as the translation vector t of the transform matrix between two floors. The rotation matrix R is unit matrix in this case. Finally we get the transformation matrix T. This method is used to merge the maps of the different floors by using the global coordinate.

IV. Evaluation

A. The Test Environment and Robot Platform

We tested the approach in a multi-floor building using a mobile robot. The experiment took place in the Guangdong provincial Key Laboratory of Digital Signal and Image Processing, located at the Science and Technology Building, which is shown in Figure 2. The robot is a TurtleBot 2 mobile robot equipped with the Rplidar sensor, kinect sensor, Inertial measurement unit (IMU, including barometric pressure sensor) and a laptop with Linux operating system. The IMU sensor is interfaced to the laptop through the Arduino MEGA 2650 board by the USB. We start to implement the experiment at the 3^{th} , the 4^{th} , the 5^{th} and the 6^{th} floors of the building.

B. Experimental Results

1) 2D map building and localization: In our paper, we mainly use the Gmapping package to build 2D map on each floor. The constructed 2D map is shown in Figures 3-6. Then we can locate the mobile robot in 2D map with the KLD-sampling MCL.

2) 3D Map Building: In this section, the 3D environment is obtained when the mobile robot moves from one end of the corridor to the elevator. The length and width of the corridor is about 30 meters and 2 meters, respectively. We use the gamepad to control the robot moving. We subscribe the ROS topic which publishes the pose information when the robot moves 0.2 meters or rotates $45°$. At the same time,

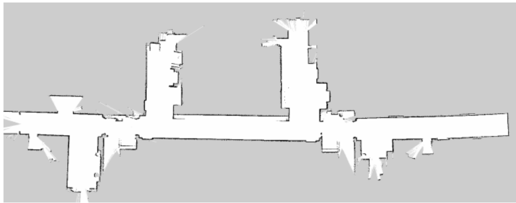

Figure 3. The 2D map of the 3^{th} floor

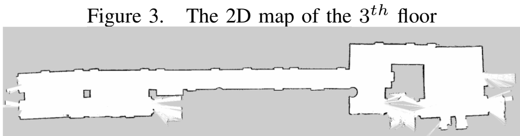

Figure 4. The 2D map of the 4^{th} floor

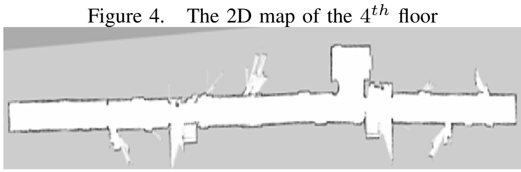

Figure 5. The 2D map of the 5^{th} floor

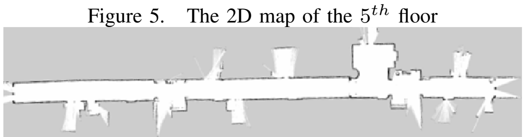

Figure 6. The 2D map of the 6^{th} floor

Figure 7. The 3D map of the 3^{th} floor

Figure 8. The 3D map of the 4^{th} floor

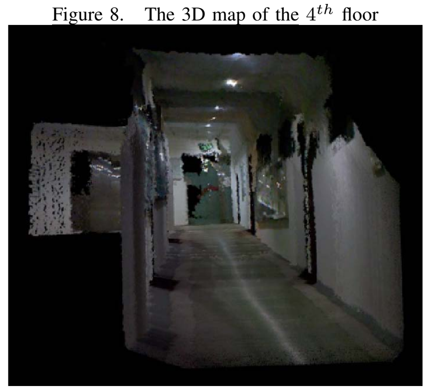

Figure 9. The 3D map of the 5^{th} floor

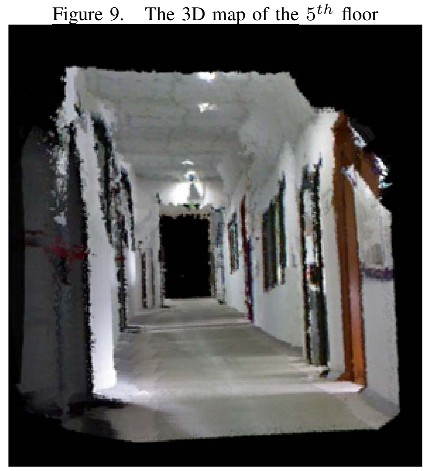

Figure 10. The 3D map of the 6^{th} floor

the kinect collects image information at the different time to merge a pair of the point clouds of the adjacent images by the system threads and modules in Figure 1. The 3D maps of the whole corridor of 3^{th}-6^{th} floors are shown in Figures 7-10.

3) 3D Map of Multi-Floor Buildings: Using the above the method, we get the full 3D map for each floor. Then according to the barometric pressure value in Figure 11, we obtain the EKF pressure values and calculate the difference in height, which is applied to the transformation matrix and to implement the point cloud registration in Figure 12. The difference in height is used as the translation vector t of the transform matrix between two floors. The rotation matrix R is unit matrix. Finally we get the transformation matrix T.

C. Comparisons

After the 3D indoor map is constructed, we make a comparison between the RGB-D SLAM algorithm and ORB-SLAM algorithm provided by Felix Endres [3] and Murartal Raul [1], respectively.

The 3D map constructed by the RGB-D SLAM algorithm is shown in Figure 13. The green line represent robot pose and the yellow lines is the transformational relation between these poses. These poses are optimized by the General Graph Optimization (g2o) [15]. Though this algorithm presents a part of environment, the image is distorted. The main reason is that the experiment environment has some very similar

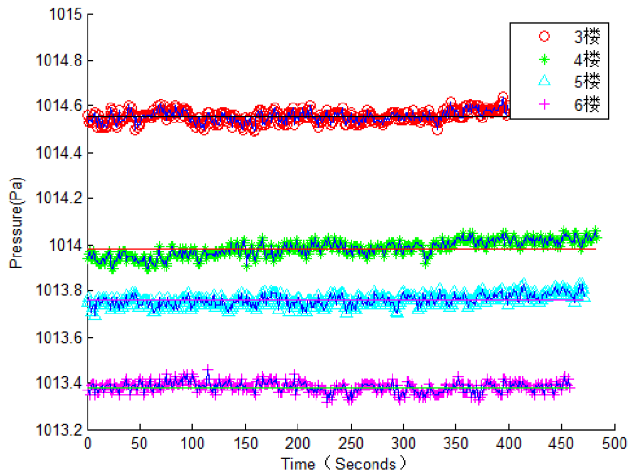

Figure 11. pressure reading from 4 different floors of the building

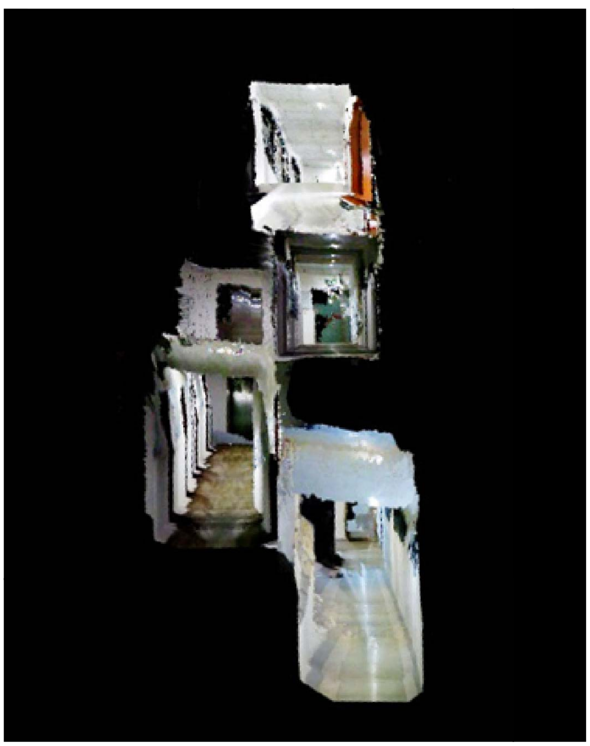

Figure 12. 3D mapping of multi-floor buildings

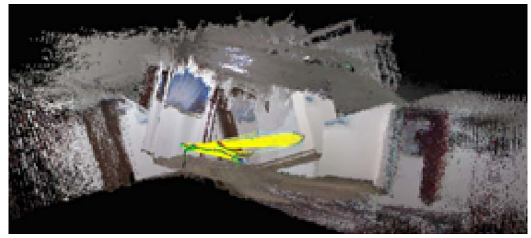

Figure 13. RGBD-SLAM: 3D map

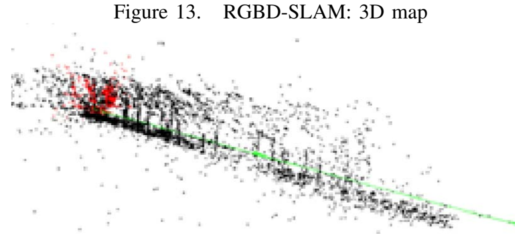

Figure 14. ORB-SLAM: 3D map

features, including brown doors, symmetrical corridor and white walls, and lack of distinct features, which could cause mismatch and cumulative errors. These errors generated by the visual odometer is not corrected by g2o [15].

The result of the ORB-SLAM algorithm is shown in Figure 14. The green line is the trajectory of the robot. The red area represent that the trajectory is the closed loop. The black point is the feature extracted by the ORB-SLAM algorithm. We only get the sparse point cloud which is the feature of the environment, such as the door or windows, not the entire environment.

V. Conclusion

In this paper, we implement the 3D mapping of multi-floor building based on several sensors fusion and Monte Carlo Localization in 2D map. The MCL makes up for the short localization accuracy of the visual localization. Some experiments have been conducted to test the proposed approach with a mobile robot in a multi-floor building, and the results show that the approach is simple, efficient and has a better result in 3D mapping of multi-floor buildings than other state-of-art methods.

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] R. Mur-Artal and J. D. Tardos, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. PP, no. 99, pp. 1–8, 2016.

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

[3] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard, "3-d mapping with an rgb-d camera," *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 177–187, 2014.

[4] "http://www.ros.org/gmapping."

[5] A. G. Ozkil, Z. Fan, J. Xiao, and S. Dawids, "Mapping of multi-floor buildings: A barometric approach," *IEEE/rsj International Conference on Intelligent Robots and Systems*, pp. 847–852, 2011.

[6] L. Zhao, Z. Fan, W. Li, H. Xie, and Y. Xiao, "3d indoor map building with monte carlo localization in 2d map," *International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration*, pp. 236–240, 2016.

[7] R. Chatila and J. Laumond, "Position referencing and consistent world modeling for mobile robots," *IEEE International Conference on Robotics and Automation. Proceedings*, vol. 2, pp. 138–145, 1985.

[8] C. R, S, G. D. Tipaldi, and C. Stachniss, "On the position accuracy of mobile robot localization based on particle filters combined with scan matching," *IEEE/rsj International Conference on Intelligent Robots and Systems*, pp. 3158–3164, 2012.

[9] L. Iocchi and S. Pellegrini, "Building 3d maps with semantic elements integrating 2d laser, stereo vision and imu on a mobile robot," *Proceedings of the 2nd ISPRS International Workshop on 3D-ARCH*, 2009.

[10] Y. Morales, T. Tsubouchi, and S. Yuta, "Vehicle 3d localization in mountainous woodland environments." *IEEE/rsj International Conference on Intelligent Robots and Systems*, pp. 3588–3594, 2009.

[11] G. Grisettiyz, C. Stachniss, and W. Burgard, "Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling," 2015, pp. 2432–2437.

[12] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007.

[13] A. Doucet, N. D. Freitas, K. P. Murphy, and S. J. Russell, "Rao-blackwellised particle filtering for dynamic bayesian networks," *Conference on Uncertainty in Artificial Intelligence*, pp. 176–183, 2000.

[14] S. Thrun, "Probabilistic robotics," *MIT Press*, pp. 52–57, 2005.

[15] R. Kmmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G2o: A general framework for graph optimization," *IEEE International Conference on Robotics and Automation*, pp. 3607–3613, 2011.