A novel SLAM framework based on 2D LIDAR

Shaofeng Wu*, Jingyu Lin*
School of Electrical Engineering, Guangxi University, Nanning, China

*Corresponding author e-mail: jylin@gxu.edu.cn, *shaofeng693@126.com

Abstract. SLAM is a basic problem in many mobile robot applications. The most commonly used SLAM framework is too complex, and many 3D data algorithms are not suitable for 2D LIDAR. In order to solve this problem, we propose a novel SLAM framework which is more in line with embedded platform. In our framework, we use the classic ICP algorithm as the odometer to calculate the pose of the mobile robot, and use Kalman filter as optimization to remove the accumulating drift. In the aspect of map generation, we first generate the occupancy grid map, then transform occupancy grid map to binary map as the final environment map. We build a simulation platform based on MTLAB to verify the feasibility and effectiveness of our proposed framework.

1. Introduction
The simultaneous localization and mapping (SLAM) technology has become a research hotspot in the field of robotics and has important research value and economic value [1, 2, 3].

Based on the different sensors, we classify them into three categories: LIDAR SLAM [1], visual SLAM [2] and SLAM of vision and LIDAR fusion [3]. The whole SLAM framework can be roughly divided into front end, back end, loop closing and mapping. The front end and back end are also called odometer and optimization.

As for LIDAR SLAM, the front end is to estimate the pose of the mobile robot by LIDAR data registration method [5, 9, 10]. The back end mainly uses the filter theory [6, 11, 12] to optimize the results of the front end. The loop closing to determine whether the robot has ever reached the previous position. Mapping generates a map describing the unknown environment.

The visual SLAM is similar to LIDAR SLAM, but these methods used are somewhat different. The main visual odometer is based on feature points [14]. The mainstream of visual SLAM optimization is graphics technology [15, 16]. For mapping, there are topology map, semantic map and so on.

SLAM of vision and LIDAR fusion is a research hotspot, and mainly about how to integrate different data and method. These three frameworks are very complex and not cost-effective in some cases.

Indoor environment is a simple and structured environment, and most of the objects are stationary. Furthermore, many indoor mobile robots have limited computing resources and can’t run complex algorithms like servers. In order to solve these problems, we propose a novel 2D LIDAR SLAM framework. This framework is composed of odometer, optimization and mapping. We calculate the pose of mobile robot by ICP [5] and use Kalman filter [6] to remove accumulated errors. For mapping, we first spliced the data, then generate the occupancy grid map and the binary map.
The problem description is in Section 2. The SLAM framework we addressed in Section 3. We design a practical SLAM system in Section 4. These experiments show that our method feasibility and effectiveness in Section 5.

2. Problem Description

In the unknown environment, the problem of SLAM can be described as that the mobile robot can get its own location only relying on its own sensors and generate the map about the environment.

The sampling of the sensor has time interval, we change the motion of continuous time into that of discrete time for the convenience of mathematical description. The relationship between the sensor carried by the mobile robot and its motion can be described mathematically as follows:

$$M_k = f(M_{k-1}, U_k) + R_k$$  \hspace{1cm} (1)

Where $M_k$ represents the position (pose) of mobile robot at the $k$th moment, $M_1, \ldots, M_k$ represent the trajectory of the mobile robot. $U_k$ is the reading of the motion sensor, $R_k$ is the noise of the motion sensor.

The whole map of mobile robot is made up of different landmarks. At the $M_k$ position, the mobile robot can observe the landmark $LM_j$ with its sensors, and get the data $Z_{k,j}$ about the environment. We use the observation equation to express this relationship as follows:

$$Z_{k,j} = h(LM_j, M_k) + Q_{k,j}$$  \hspace{1cm} (2)

Where $Q_{k,j}$ is the noise produced during observation. Different landmarks will be measured at different times, while the data is being collected. $LM_1, \ldots, LM_N$ is the landmark at different times.

These two equations describe the relationship between variables in a SLAM system. $f$ represents the mapping relationship from $M_{k-1}$ and $U_k$ to $M_k$. $U_k$ can be obtained from the reading of motion sensors, such as IMU, code disk, etc. In 2D space, the pose of the mobile robot is represented by a $2 \times 1$ translation matrix $T = [x, y]^T$ and a rotation angle $\phi$. For 2D LAIDR, we specify the motion equation as concretize:

$$\begin{bmatrix}
\hat{x} \\
\hat{y} \\
\hat{\phi}
\end{bmatrix}_k = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
\hat{x} \\
\hat{y} \\
\hat{\phi}
\end{bmatrix}_{k-1} + \begin{bmatrix}
\hat{x} \\
\hat{y} \\
\hat{\phi}
\end{bmatrix}_{k-1} + R_k$$  \hspace{1cm} (3)

$$M_k = A \cdot M_{k-1} + U_k + R_k$$  \hspace{1cm} (4)

Where $M_k = \begin{bmatrix}
\hat{x}_k \\
\hat{y}_k \\
\hat{\phi}_k
\end{bmatrix}^T$, $M_{k-1} = \begin{bmatrix}
\hat{x}_{k-1} \\
\hat{y}_{k-1} \\
\hat{\phi}_{k-1}
\end{bmatrix}^T$, $U_k = \begin{bmatrix}
\bar{x}, \bar{y}, \bar{\phi}
\end{bmatrix}^T$, $A$ is unit matrix, $R_k$ is a $3 \times 3$ covariance matrix.

For 2D LIDAR, their readings are a pile of point cloud data, which need to use data matching algorithm to find the final $Z_{k,j}$. Therefore, $h$ represents the mapping relationship to get $Z_{k,j}$ from $M_k$ and point cloud sampled at $LM_j$. We specify the observation equation as concretize:
\[
\begin{bmatrix}
  x \\
  y \\
  \phi_k
\end{bmatrix} = \begin{bmatrix}
  1 & 0 & 0 \\
  0 & 1 & 0 \\
  0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
  \hat{x} \\
  \hat{y} \\
  \hat{\phi}_k
\end{bmatrix} + Q_k
\] (5)

\[
Z_k = H \cdot M_k + Q_k
\] (6)

Where \(Z_k = [x_k, y_k, \phi_k]^T\), \(H\) is unit matrix, \(Q_k\) is a 3×3 covariance matrix.

3. Framework design of 2D LIDAR SLAM

The classic LIDAR SLAM framework [4] as follows:

![Figure 1. Classic LIDAR SLAM framework.](image)

The classic LIDAR SLAM framework consists of five parts: LIDAR Information, Odometry, Optimization, Loop Closing and Mapping. Because classic LIDAR SLAM framework is too complex, and many 3D algorithms are not suitable for 2D LIDAR. We propose a novel SLAM framework which is more in line with 2D LIDAR. We modified the whole SLAM framework as follows:

![Figure 2. Our LIDAR SLAM framework.](image)

In order to increase the reliability of the sensors and simplify the optimization algorithm, we added a motion sensor. The blocks of LIDAR information and Motion Sensor Information mainly includes data preprocessing. Odometry is to calculate the transformation relationship between two frames LIDAR data collected by adjacent position, which is a data registration problem. The section of Mapping consists of Optimization, Occupancy Grid Map and Binary Map. Compared to the classical LIDAR SLAM framework, our proposed framework omits the complex Loop Closing and add the motion sensor.

4. LIDAR SLAM System Overall Design

Based on the 2D SLAM framework we proposed, we designed a SLAM system. To make the description easier to understand, we give the data "flow" (input-output) relationship between the parts of the system:
Registration of LIDAR data

Optimization

Mapping

\[ k \text{ updates to the next moment} \]

\[ \text{Is } k \text{ equal to } N? \]

\[ \text{No} \]

\[ \text{Yes} \]

\[ \text{End} \]

**Figure 3.** Flow chart of 2D LIDAR SLAM system.

LIDAR collects a frame of point cloud data about unknown environment at time \( k \), expressed as:

\[
D^k = \left\{ D^k_i = \left[ x^k_i, y^k_i \right]^T | i = 1, 2, ..., N \right\}
\]  

(7)

Where \( N \) is the number of LIDAR data points of a frame, \( D^k_i \) is the \( i \)th data point of the frame \( k \), \((x^k_i, y^k_i)\) is the coordinate in Cartesian coordinate system.

### 4.1. Registration of LIDAR data

We input two frames of 2D LIDAR data \( D^k \) and \( D^{k-1} \), and then output the translation matrix \( T_k \) and rotation angle \( \varphi_k \). Widely used data registration method is ICP [5], TrICP [9], SICP [10], etc. We use the standard 2D ICP [5] method to align the 2D LIDAR data, and the flow of ICP algorithm is:

1. Search closest points and construct point pairs.
2. Calculate translation matrix \( T_k \) and rotation angle \( \varphi_k \) according to the point pairs.
3. Use \( T_k \) and \( \varphi_k \) to transform the point set.
4. Calculate the error function. If the error is less than the threshold or the number of iterations reaches the threshold value, terminate the iteration. Otherwise return to step (1).

For this paper, we reference literature [7] to calculate these parameters about 2D LIDAR data. Assuming that both \( D^k \) and \( D^{k-1} \) have \( N \) data points, the center points of the two sets are \( C^k \) and \( C^{k-1} \):

\[
C^k = \frac{1}{N} \sum_{i=1}^{N} D^k_i, \quad C^{k-1} = \frac{1}{N} \sum_{i=1}^{N} D_i^{k-1}
\]

(8)

Make \( \overline{D}_i^k = D_i^k - C^k \), \( \overline{D}_i^{k-1} = D_i^{k-1} - C^{k-1} \), then the rotation angle \( \varphi_{icp} \) between the two set of data is:

\[
\varphi_{icp} = \arctan \frac{\sum_{i=1}^{N} (D_i^{k,x} D_i^{k-1,y} - D_i^{k,y} b_i^{k-1,x})}{\sum_{i=1}^{N} (D_i^{k,x} D_i^{k-1,x} + D_i^{k,y} D_i^{k-1,y})}
\]

(9)

Where, \( D_i^{k,x}, D_i^{k,y}, D_i^{k-1,x} \) and \( D_i^{k-1,y} \) are the components of \( \overline{D}_i^k \) and \( \overline{D}_i^{k-1} \) in \( x \) and \( y \) directions respectively. \( \overline{D}_i^k \) is the difference between data point \( D_i^k \) and the center point \( C^k \).

The translation matrix \( T_{icp} \) of point set \( D^k \) and \( D^{k-1} \) can be obtained by the parameter \( \varphi_{icp} \):
\[
T = \begin{pmatrix}
C^{k,x} & \cos \varphi_{kp} & \sin \varphi_{kp} \\
C^{k,y} & -\sin \varphi_{kp} & \cos \varphi_{kp}
\end{pmatrix}
\]
(10)

Where, \(C^{k,x}\), \(C^{k,y}\), \(C^{k-1,x}\) and \(C^{k-1,y}\) are the components of \(C^k\) and \(C^{k-1}\) in \(x\) and \(y\) directions respectively.

4.2. Optimization

We add the motion sensor which can measure the rotation angle \(\varphi_k\) and translation \(T_k\) of the mobile robot. Besides we obtain \(\varphi_k\) and \(T_k = [x_k, y_k]^T\) by ICP method. Common optimization methods are Kalman filter [6], Extended Kalman Filter [11], Particle Filter [12] and so on. To reduce complexity, we use Kalman filter [6] for data fusion to get the final rotation angle \(\hat{\varphi}_k\) and translation \(\hat{T}_k = [\hat{x}_k, \hat{y}_k]^T\). The steps of Kalman filter are:

1) Predict: according to the mean \(\bar{M}_{k-1}\) and covariance \(\bar{P}_{k-1}\) of \(k-1\) moment, the mean and covariance of \(k\) moment are calculated:

\[
\bar{M}_k = A_k \bar{M}_{k-1} + U_k, \quad \bar{P}_k = A_k \bar{P}_{k-1} A_k^T + R_k
\]
(11)

2) Calculate Kalman Gain \(K\):

\[
K = \bar{P}_k H_k^T \left( H_k \bar{P}_k H_k^T + Q_k \right)^{-1}
\]
(12)

3) Update:

\[
\hat{M}_k = \bar{M}_k + K \left( Z_k - H_k \bar{M}_k \right)
\]
\[
\hat{P} = (I - KH_k) P_k
\]
(13)

4.3. Mapping

The part of Mapping needs further refinement, and its flow chart is as follows:

\[
\begin{array}{cccc}
\hat{T}_k, \hat{\varphi}_k, \hat{D}^k & \text{Coordinate Transformation} & \text{Map Generation Process 1} & \text{Map Fusion} \\
\hat{D} & \text{Map Generation Process 2} & \hat{G}^k & \hat{V}^k
\end{array}
\]

Figure 4. Flow chart of LIDAR mapping.

According to \(\hat{T}_k\) and \(\hat{D}_k = [\hat{x}_k, \hat{y}_k]^T\), we transform the coordinates of each point in the data set \(\hat{D}^k\) to get a new data set \(\hat{D}^k\). The data rigidity transformation formula in \(\hat{D}^k\) is:
\[
\begin{bmatrix}
\cos \hat{\phi}_k & \sin \hat{\phi}_k \\
-\sin \hat{\phi}_k & \cos \hat{\phi}_k
\end{bmatrix}
\cdot \hat{D}_i^k + \hat{T}_k
\]  

(14)

Where \( \hat{D}_i^k \) and \( D_i^k \) is data point in data set \( \hat{D}^k \) and \( D^k \) respectively.

**Map Generation Process** generates the occupancy grid map \( L^k \) from data set \( \hat{D}^k \) as follows:

**Map Generation Process**

\[
G^k = L^k + \hat{G}^{k-1}
\]

(15)

Where \( L^k \), \( G^{k-1} \) and \( G^k \) are 2D matrices, and the value of each element represents the occupied state. Take Figure 5 as an example, Figure 5(c) is the result of fusion of Figure 5(a) and Figure 5(b). When transforming \( G^k \) into a binary map \( V^k \), it is only necessary to judge the state value of each element in \( G^k \). If the value is greater than the set threshold, the corresponding value in \( V^k \) is set to 1, otherwise to 0.

5. **Experiments**

In order to effectively verify our proposed framework, we built a simulation experiment platform based on MATLAB 2016a. We generate LIDAR data through this platform as follows:
Figure 6. Simulation of LIDAR scanning.

In Figure 6, left is a 640cm × 480cm 2D map scanned by a LIDAR. The black block is obstacles, the white regions is the free space, the gray star is the LIDAR center, and the red line is the LIDAR beam. The red circles in the right are the sampled LIDAR points shown in 2D coordinates. To make the scene more realistic, we add a 5dB Gaussian noise to the sample data.

In order to use on the embedded platform such as UAV, many SLAM frameworks only have positioning and mapping modules, such as [8]. Our framework is compared with this one labeled NO-OPT. We use RMSE (Root Mean Square Error) and ATE (Absolute Trajectory Error) to measure the performance of SLAM system [13]. The translation matrix \( T = [x, y]^T \) and rotation angle \( \phi \) make up the pose of the robot. To facilitate evaluation, we express the translation as \( \text{Trans} = |x^2 + y^2|^{1/2} \).

The result of map generation in the experiment is shown in Figure 7.

In Figure 7(a), 7(d) and 7(e), the black block is obstacles, the white regions is the free space, and the red dot is the trajectory of the mobile robot. The mobile robot moves along the set direction in original 640cm × 480cm 2D map and sampling data by 2D LIDAR in Figure 7(a). Figure 7(b) and Figure 7(c) is the occupancy grid map generated by our framework and NO-OPT respectively. The darker the color in the occupancy grid map, the greater the probability of free space. Figure 7(d) and Figure 7(e) is the binary map generated by our framework and NO-OPT respectively. By comparing with the original map Figure 7(a), we can see that the map generated by our framework is more perfect, and the map generated by NO-OPT is obviously distorted at Rec1 and Rec2.
We analyze the pose of the mobile robot calculated by two methods is shown in Figure 8.

In Figure 8, $GT$ is the value of ground truth, $SEN$ is the value of motion sensor reading, $NO$-$OPT$ is the value calculated by $NO$-$OPT$ framework, $Our$ is the value calculated by our framework. These four kinds of robot trajectories with different values are shown in Figure 8(a). We enlarge the regions Arc1’ and Arc2’ in Figure 8(a) to get Figure 8(c) and Figure 8(d). Figure 8(b) represents the change process of orientation angle of mobile robot.

As shown in Figure 8(c) and Figure 8(d), due to the accumulation of errors in the same direction, the trajectory calculated by method $NO$-$OPT$ deviates significantly from the ground truth, which is the reason for Arc1 and Arc2 in Figure 7(d). Compared with the ground truth, the orientation error calculated by $NO$-$OPT$ framework accumulates in the same direction in Rec1’ and Rec2’ of Figure 8(b), resulting in deformation at Rec1 and Rec2 of Figure 7(d). Most importantly, our framework
effectively eliminates the accumulated errors, the generated map truly restores the original map, and the trajectory of the mobile robot does not deviate from the ground truth.

The precision results of these SLAM system are as follows:

In Table 1, The RMSE of our framework are lower than that of NO-OPT in both translation and rotation. In Figure 9(a), at the beginning of the motion, the ATE of the two frameworks are similar. The ATE of NO-OPT increased significantly with increasing motion, while the ATE about translation of our framework remained almost unchanged in Figure 9(a). In Figure 9(b), ATE of rotation of our framework has been very low, as ATE of NO-OPT continues to grow. As mentioned above, our framework has better performance, especially for rotation.

| Framework | RMSE(Trans)/cm | RMSE(φ)/degree |
|-----------|----------------|----------------|
| Our       | 9.14           | 0.50           |
| NO-OPT    | 19.95          | 9.26           |

These experiments show that the proposed framework can effectively remove accumulated errors, generate realistic unknown environment map and accurately calculate the trajectory of mobile robot.

6. Conclusion

According to the data characteristics of 2D LIDAR, we propose a novel SLAM framework. This framework uses ICP algorithm to calculate the pose of mobile robot, and then uses Kalman filter to correct the accumulated error. In order to restore the unknown environment explored by the mobile robot more realistically, we first splice the data, then generate the occupancy grid map, and finally generate the binary map. We will extend our framework to 3D LIDAR data.

Acknowledgments

This work was jointly supported by China Scholarship Council (Grant No. 201906660002) and National Nature Science Foundation of China (Grant No. 61561005).

References

[1] Han, J., Kim, J., & Shim, D. H. (2019). Precise Localization and Mapping in Indoor Parking Structures via Parameterized SLAM. IEEE Transactions on Intelligent Transportation Systems, 20(12), 4415-4426.

[2] Dong, Y., Wang, S., Yue, J., Chen, C., He, S., Wang, H., & He, B. (2019). A Novel Texture-Less Object Oriented Visual SLAM System. IEEE Transactions on Intelligent Transportation Systems, 1-14.

[3] W. Shao, S. Vijayarangan, C. Li and G. Kantor, "Stereo Visual Inertial LiDAR Simultaneous
Localization and Mapping," 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 2019, pp. 370-377.

[4] Z. Yao, X. Chen, D. Xie and Z. Yang, "Preliminary study of salient region SLAM," 2017 2nd International Conference on Robotics and Automation Engineering (ICRAE), Shanghai, 2017, pp. 451-454.

[5] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 2, pp. 239-256, Feb. 1992.

[6] Cloutier, J. R. (2001). Review of Fundamentals of Kalman Filtering: A Practical Approach. Progress in Astronautics and Aeronautics, Vol. 190. Journal of Guidance Control and Dynamics, 24(4), 863-864.

[7] Quyn. Study on 2D map creation of mobile robot based on lidar [D]. University of Defense Science and Technology, 2009. (in Chinese)

[8] Song Kainan, Wang Biao, & Tang Chaoying. (2016). "Research on Indoor Localization Method based on Laser Ranging Scanning," Laser and Infrared, 46(8), 939-943

[9] D. Chetverikov, D. Svirko, D. Stepanov and P. Krsek, "The Trimmed Iterative Closest Point algorithm," Object recognition supported by user interaction for service robots, Quebec City, Quebec, Canada, 2002, pp. 545-548 vol.3.

[10] Bouaziz, Sofien, Andrea Tagliasacchi, and Mark Pauly. "Sparse iterative closest point." symposium on geometry processing (2013), 32(5): 113-123.

[11] Bonnor, N. (2014). Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems – Second Edition Paul D. Groves Artech House, 2013, 776 pp ISBN-13: 978-1-60807-005-3. Journal of Navigation, 67(01), 191-192.

[12] Gil, A., Reinoso, I., Ballesta, M., & Julia, M. (2010). Multi-robot visual SLAM using a Rao-Blackwellized particle filter. Robotics and Autonomous Systems, 58(1), 68-80.

[13] Y. Fang, G. Shan, T. Wang, X. Li, W. Liu and H. Snoussi, "HE-SLAM: A Stereo SLAM System Based on Histogram Equalization and ORB Features," 2018 Chinese Automation Congress (CAC), Xi'an, China, 2018, pp. 4272-4276.

[14] E. Rublee, V. Rabaud, K. Konolige and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," 2011 International Conference on Computer Vision, Barcelona, 2011, pp. 2564-2571.

[15] H. Gao, X. Zhang, J. Wen, J. Yuan and Y. Fang, "Autonomous Indoor Exploration Via Polygon Map Construction and Graph-Based SLAM Using Directional Endpoint Features," in IEEE Transactions on Automation Science and Engineering, vol. 16, no. 4, pp. 1531-1542, Oct. 2019.

[16] 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 sub-maps," 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 2019, pp. 169-175.