Outdoor SLAM Using Monocular Vision-Based Localization with LIDAR-Aided Mapping for service robot in Highway

Shuwen Pan[1], Yuanyuan Li[2], Pengying Du[1]*, Yan Liu[1]

[1]*Key Discipline of Automation, School of Information and Electrical Engineering, Zhejiang University City College, Hangzhou, Zhejiang, China*

[2]*College of Control Science and Engineering, Zhejiang University, Hangzhou, Zhejiang, China*

**Corresponding Author.*

Abstract

This paper designed an intelligent service robot system in highway based on multi-sensor fusion. The mobile robot attempts to fuse the lidar information and monocular vision information to estimate the pose of itself and obtain an environmental map. It adapts a new SLAM method which combines lidar and vision information. Lidar is used to obtain the 2D occupancy grid map and the monocular vision SLAM algorithm uses the Extended Kalman Filter (EKF) to magnify the pose estimation. The 3-DOF pose provided by lidar is obtained through Cartographer algorithm and the monocular vision SLAM who offers the 6-DOF pose is realized with ORB-SLAM. The experimental results show that the system is effective in application as an intelligent service robot of highway.

Keywords: *Monocular Vision, SLAM, EKF, Lidar-based, Service Robot, Sensor Fusion*.

I. Introduction

Intelligent transportation has become a main feature of the transportation system, exerting a broad and profound impact on the industrial governance system and service model. After the accident, there is a large area of traffic congestion, and it is difficult for the accident treatment vehicle to enter the accident scene at the first time. In the scene of rescue, the mobile robot needs to navigate in unknown environments relied on its own onboard sensors.

The mobile robot adapts a new SLAM method which combines lidar and vision information. Lidar is used to obtain the 2D occupancy grid map and the monocular vision SLAM algorithm uses the Extended Kalman Filter (EKF) to magnify the pose estimation. The 3-DOF pose provided by lidar is obtained through Cartographer algorithm and the monocular vision SLAM who offers the 6-DOF pose is realized with ORB-SLAM [1]. A 2D map is generated to calculate the pose not sensitive to errors. The complete 6-DOF pose estimation is got by EKF (Extended Kalman Filter) which filters the estimated position of robot with IMU and visual SLAM information.

This paper is devoted to the development of an outdoor intelligent service robot which can be transported by police vehicles when an accident occurs, and then carry necessary drugs to the destination at the same time.

II. Related Works

2.1 Lidar SLAM

Smith *et al*. [2] proposed a Kalman filter-based real-time localization and mapping technology (Simultaneous Localization and Mapping, or SLAM) in 1988, which laid a basic blueprint for the development of the entire SLAM.

On the basis of the Kalman filter, the extended Kalman filter method and the lossless Kalman filter method have been proposed one after another, which became the most popular SLAM algorithm at that time, but the use of the extended Kalman filter method requires the definition of road signs in advance, and once the noise of the strong

assumptions of distribution and state transition are not established, the algorithm will diverge [3]. Therefore, scholars such as Murphy [4] later adopted the Rao-Blackwellised particle filter method, but this method often causes the map to occupy a lot of resources and affects the performance of the SLAM algorithm. In order to reduce resource consumption and improve the accuracy of map construction, Grisetti *et al*. [5,6] proposed the GMapping algorithm based on the Rao-Blackwellised particle filter method, which is currently a relatively mature laser SLAM method.

The graph optimization scheme proposed by Gutmann *et al*.[7] is beneficial to the improvement of computer performance and has been paid attention again. Algorithms such as Cartographer [8] and LSD-SLAM [9] are developed based on graph optimization theory. This method divides the SLAM process into a front-end and a back-end. The front-end mainly updates the pose information of the robot by filtering. Most of the back-end optimizes the global pose and landmark information of the robot through optimization methods, forming a typical SLAM system framework. When loopback is detected, global optimization can be performed to improve the accuracy of map construction in complex environments, but the real-time performance is poor.

2.2 Monocular Vision SLAM

According to different working methods, vision sensors are mainly divided into three categories: monocular cameras, binocular cameras and RGB-D cameras [10]. Monocular cameras are favored by researchers around the world due to their low cost, small size, and low power consumption. However, monocular cameras alone cannot restore the true scale of the visual map, so they usually need to be combined with other perceptual sensors.
The research on visual positioning is mainly the research on visual SLAM. The front end and the back end together form a complete visual SLAM system. The front-end is the realization of the visual odometer, which mainly includes the processes of feature extraction and matching, pose initialization, and tracking. The back-end eliminates the cumulative error of the system by detecting closed-loop constraints and constructing global optimization problems, completing the overall optimization of the camera trajectory and the map point cloud, and constructing and storing a 3D point cloud map for visual positioning.

When the robot is performing laser global positioning, it is easy to fail in positioning where features are missing. At this time, if there is the aid of visual positioning, the stability will be greatly improved. ORB-SLAM includes modules common to all SLAM systems: tracking, mapping, re-localization, and loop closing. ORB-SLAM adopts the Bag of Words model to perform loop detection and re-localization. The effect of eliminating the cumulative error of the system positioning is obvious. The bag of words model greatly reduces the amount of calculation when the system searches for features [11]. When the robot encounters some unexpected situations such as loss of positioning, the bag of words model can be used to reposition the robot's approximate pose on the map in a short time.

III. System Configuration

3.1Coordinates System Establishment

The chassis of the mobile robot adopts four-wheel structure, which is symmetrically distributed, and a variety of sensors are arranged on the platform. In this paper, the Cartesian coordinate system is adopted, and there are mainly three coordinate systems: the world coordinate system$X_W Y_W Z_W$, the robot coordinate system $X_R Y_R Z_R$ and the sensor coordinate system $X_S Y_S Z_S$. The coordinate systems are all right-handed. Among them, IMU and GPS sensors are integrated with the robot chassis and can be incorporated into the robot coordinate system, so the sensor coordinate system mainly refers to the coordinate system where the lidar is located. Therefore, the coordinate system relationship of the 2D mobile robot can be expressed as figure 1.

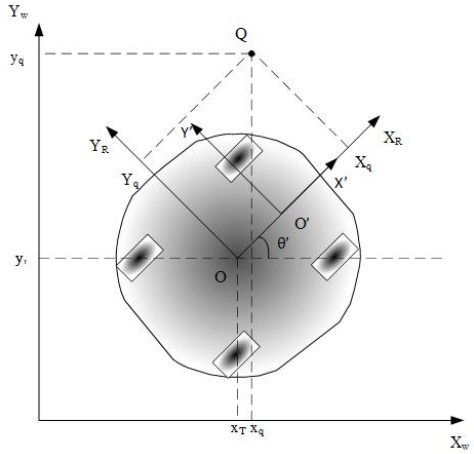


Fig 1: The Coordinate System in 2D Plane

The origin of the world coordinate system $X_W Y_W Z_W$ is the starting point of the robot's motion, and the origin of the robot coordinate system $X_R Y_R Z_R$ is the center of gravity of the moving chassis. The X and Y axes of the sensor coordinate system $X_S Y_S Z_S$ are the same as that of the mobile robot, but there is deviation in the X and Z axes. At time T, the robot coordinates in $X_W Y_W Z_W$ are (x_t, y_t), and the direction of motion is expressed by $\theta([-\pi, \pi])$ of the X-axis of the robot coordinate system $X_R Y_R Z_R$, moving forward from the X-axis of the $X_W Y_W Z_W$. Therefore, at time T, the robot coordinates in $X_W Y_W Z_W$ are expressed as (x_t, y_t, θ_t). The above coordinate systems can be transformed between each other by rotation and translation. The rotation transformation matrix of robot coordinate system $X_R Y_R Z_R$ to world coordinate system $X_W Y_W Z_W$ is as follows.

$$R_W^R = \begin{bmatrix} \cos\theta_T & -\sin\theta_T & 0 \\ \sin\theta_T & \cos\theta_T & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

The rotation transformation matrix from the world coordinate system $X_W Y_W Z_W$ to the robot coordinate system $X_R Y_R Z_R$ is written as Eq.2.

$$R_R^W = \begin{bmatrix} \cos\theta_T & \sin\theta_T & 0 \\ -\sin\theta_T & \cos\theta_T & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

There is a point Q at time T, and its position coordinate in $X_R Y_R Z_R$ is expressed as (x_t, y_t).

$$\begin{bmatrix} x_q \\ y_q \end{bmatrix} = \begin{bmatrix} X_q \cos\theta - Y_q \sin\theta + x_W \\ X_q \sin\theta - Y_q \cos\theta + x_W \end{bmatrix} \quad (3)$$

According to the above transformation matrix, the coordinate of Q in $X_W Y_W Z_W$ can be deduced as Eq.3.

3.2System Model

The robot is driven by 4 wheels, as shown in figure 2. It is equipped with a lidar, an IMU, a GPS, a micro host with the ROS, a screen and the RealSense R200 camera.

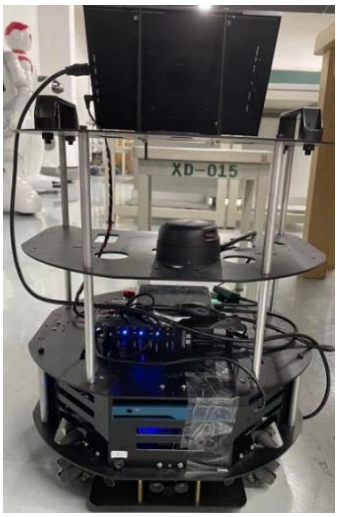


Fig 2: The Mobile Robot

3.2.1 Control Unit

The core of the robot control unit is STM32, which is directly connected with the host through USB, and the motor and voltage regulator are connected with STM32 through a control board.

The procedure converts the input instructions into the corresponding signals and transmits them to STM32, and then STM32 drives the motor to realize the movement according to the instructions. In the process of navigation, the procedure transmits the corresponding drive signal to STM32 according to the planned path and motivates the robot to move along the planned path.

3.2.2 Motion model

Four-wheel differential vehicle has a light gray outer contour, which can be similar to a dark gray single-wheel model. As shown in figure 3, four-wheel drive has strong ability and driving force in straight walking.

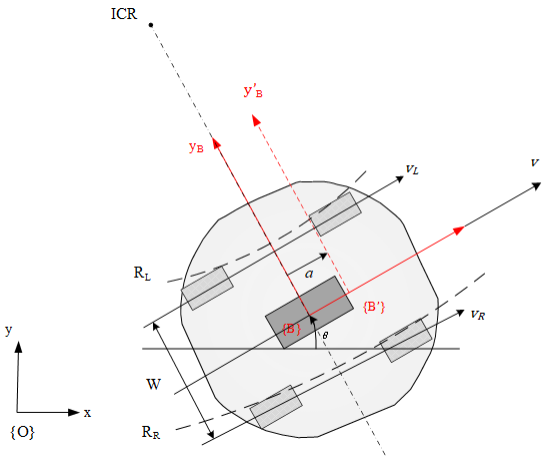


Fig 3: The Motion Model of The Robot

The wheel radius is R, the angular velocity of the left and right wheels is u, and the linear velocity is v_L and v_R respectively. B is the coordinate system of the car, whose origin is the center of the car, X-axis is the forward direction of the car, and the angular velocity of rotation around ICR (instantaneous center of rotation) is θ'. W is the distance between the left and right wheels, R_R is the rotation radius of the right wheel, andR_L is the rotation radius of the left wheel, and the deduced robot motion model is $\dot{q}$. The equations are shown as Eq. 4.

$$
\begin{aligned}
u &= [u_L \quad u_R] \\
\dot{x} &= v\cos\theta \\
\dot{y} &= v\sin\theta \\
\theta' &= \frac{v_L}{R_L} = \frac{v_R}{R_R} = \frac{v_R - v_L}{W} \\
\dot{q} &= \begin{bmatrix} \theta' \\ \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -\frac{r}{W} & \frac{r}{W} \\ \frac{r}{2}\cos\theta & \frac{r}{2}\sin\theta \\ \frac{r}{2}\sin\theta & \frac{r}{2}\sin\theta \end{bmatrix} \begin{bmatrix} u_L \\ u_R \end{bmatrix}
\end{aligned} \tag{4}
$$

3.2.3 Sensor Configuration1

The robot is equipped with a variety of sensors, including Silan A2 lidar, IMU (MPU6050), and ATGM332D GPS sensor, which are directly connected to the host through USB to TTL interface, a RealSense R200 camera which can be seen in figure 2. By running the corresponding ROS program, the data of the sensor can be read directly, and the corresponding topic can be published. IMU can provide linear acceleration, angular velocity and other data for pose estimation, while GPS can realize the absolute positioning of the robot in the outdoor environment.

IV. Sensor Fusion Slam

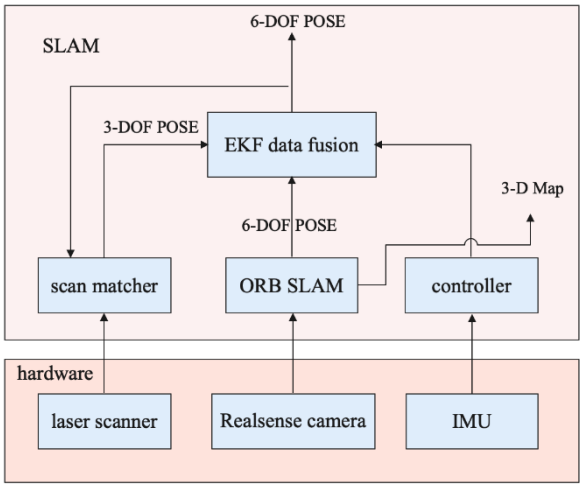


Fig 4: The Frame of the Proposed Method

4.1Lidar SLAM

4.1.1 Local maps

The main function of the local SLAM is to establish and maintain submaps, which is a loop-closed iterative process. Whenever a frame of lidar scan data is obtained, the position at this moment is regarded as a node to match with the nearest submap. To obtain an optimal pose, the generation of submaps requires continuous scan frames, and a series of submaps generate a global map [12]. The pose of the submap scanning frame ζ can be represented by T transformation, and T can convert the scanning frame into a frame in the submap (as is shown in figure 4).

$$
T = \begin{bmatrix} \cos\zeta_\theta & -\sin\zeta_\theta \\ \sin\zeta_\theta & \cos\zeta_\theta \end{bmatrix} p + \begin{bmatrix} \zeta_x \\ \zeta_y \end{bmatrix} \tag{5}
$$

The submap is shown in the form of occupied grid map. After the lost collision algorithm, each scan frame that collides with the object is inserted into a submap grid point which is combined with the segmented pixels, and the lost scan frame is inserted into the grid pixels which has already been occupied.

4.1.2 Scan Matching

Scan matching uses two or more continuous scan frames to calculate the posture of the robot to achieve the purpose of relative positioning. The obtained posture will be used as the reference coordinate of the scan frame inserted into the submap.

The process of scanning and matching is realized by the Ceres method, which uses an optimized method to obtain the pose. The initial pose provided by this method should be as accurate as possible [13]. The error of a given grid matching degree is used as a residual function, and the pose is iterated to converge using this as a reference.

$$argmin \sum_{k=1}^{K} \left(1 - M_{smoot\,h}(T_k)\right)^2 \tag{6}$$

As shown in Eq. 6, ζ is the scanning pose, T_ζ converts h_k from scanning frame to submap frame according to ζ. $M_{smoot\,h}$ function is the smooth form of the probability value in local submap. When two scan frames match, you can use IMU to measure the rotation component θ in the pose.

4.1.3 Close Loops Optimization

Sensor noise, grid map resolution and other reasons will affect the accuracy of the pose estimation in local SLAM. Although the error is small, it will accumulate with the change of time, and as the distance of advancement increases, especially the drift error will become larger [14]. Therefore, the global SLAM uses the sparse attitude adjustment (SPA) method to reduce such errors. The closed loop optimization process matches the sensor scan and the accumulated submap to identify the visited area. The optimization problem can be expressed by a nonlinear least squares problem. Every few seconds, the ceres matcher will perform calculations such as Eq.7.

$$arg\min_{p_m, p_s} \frac{1}{2}\rho\left(E^2\left(\zeta_i^m, \zeta_j^s; \sum_{ij}, \zeta_{ij}\right)\right) \tag{7}$$

$$\begin{aligned} E^2\left(\zeta_i^m, \zeta_j^s; \sum_{ij}, \zeta_{ij}\right) &= e\left(\zeta_i^m, \zeta_j^s; \zeta_{ij}\right)^T \sum_{ij}^{-1} e\left(\zeta_i^m, \zeta_j^s; \zeta_{ij}\right) \\ e\left(\zeta_i^m, \zeta_j^s; \zeta_{ij}\right)^T &= \zeta_{ij} - \begin{bmatrix} R_{\zeta_i^m}^{-1}\left(t_{\zeta_i^m} - t_{\zeta_j^s}\right) \\ \zeta_{i;m}^m - \zeta_{j;\theta}^s \end{bmatrix} \end{aligned} \tag{8}$$

Among them, $p_m = \{\zeta_{im}\}_{i=1,\ldots m}$ represents the submap pose, and $p_s = \{\zeta_{js}\}_{j=1,\ldots n}$ represents the scan pose. They can restrict E to optimize [15], whose calculation method of E is shown in Eq.8.

4.2Monocular Vision SLAM

The visual SLAM is based on ORB-SLAM, which is used to obtain 6-DOF pose estimation. Extended Kalman filter in this approach combines the final estimation with mobile robot pose provided with the sensors to obtain a 6-DOF estimate of the robot position in the generated map [16]. The structure of the VSLAM is shown in figure 5.

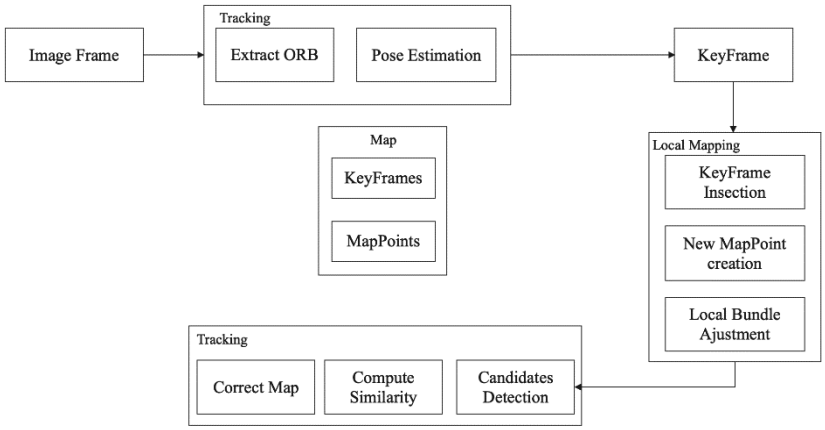


Fig 5: Description of ORB-SLAM

4.2.1Description of ORB-SLAM

ORB-SLAM is a slam system based on feature points which can build the map in real-time. ORB-SLAM can run in various environment and it has an excellent performance in large-scale environment, small-scale environment, indoor and outdoor environment. The system is robust to drastic motion, and it supports loop closing detection and relocation based on broad baselines. The system contains 3 parts which is common to all SLAM algorithms, they are Tracking, Mapping, Re-localization, and Loop closing.

4.2.2Map Initialization

Relative pose of the two frames is calculated in map initialization to triangulate a set of initial map points[1], and then use it for key frame tracking. To find a camera pose, we use the PnP algorithm for every keyframe by performing alternately RANSAC iterations. Once a camera pose with enough interior points is found, we will optimize the pose and use the map points of candidate keyframes to execute a guided search for more matches. Finally, optimize camera pose again, if there is enough interior point support, the tracking process will continue.

4.2.3Tracking Local Map

The tracking thread provides positioning and map construction. After detecting the ORB angle, the tracking thread gradually develops the map on the restored 3D map points and calculates the camera pose at the same time. Tracking is performed in a smaller subset of the entire map, called the local map, which covers the currently visible keyframes and some connected keyframes to speed up this process.

4.2.4Loop Closing

Loop-closure detection enhances the accuracy of SLAM algorithms. Bag of words method is used in ORB-SLAM loop-closure to search for loop candidate keyframes in the local map. And then loop correction merges the repeated map points and insert new edges in the co-visibility map that closes the additional loops. In order to effectively close the loop, pose graph optimization is performed on the basic graph to distribute loop closure errors along the graph.

4.2.5Data Fusion with EKF

To make full use of the available data, we adapt EKF to align the time in the robot system.

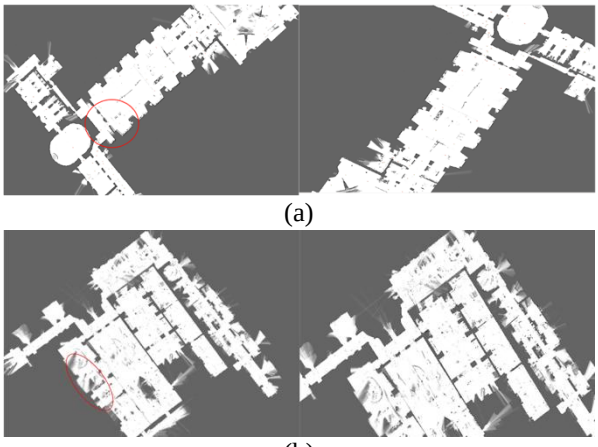
The parameters are as follows. (x_t, y_t, z_t) is the position of the mobile robot, $(v_{x_t}, v_{y_t}, v_{z_t})$ is the velocity of the robot, $(\Phi_t, \Theta_t, \Psi_t)$ is the roll, pitch and yaw angles. These parameters are all in the world coordinate.

The visual SLAM is running, the 6-DOF pose is transformed from the mobile robot to the robot coordinate, so we can obtain the pose of the robot. $E_{c,t}$ represents the estimated camera pose and $E_{D,t}$ is the transformation from camera to the mobile robot. Function f represents the transformation function from SE(3) to the vector form.

$$z_{vslam} \coloneqq f\left(E_{DC}, E_{C,t}\right) \in R^6$$
$$h_{vslam}\left(x_t\right) \coloneqq \left(x_t, y_t, z_t, \Phi_t, \Theta_t, \Psi_t\right)^T \in R^6 \quad (9)$$

V. Experiment & Result

First, we evaluate the improved SLAM method qualitatively. We use publicly available data Google to evaluate this method. The results are shown in figure 6. The left figure shows the effect of Cartographer, and the right figure shows the improved method. It can be seen that the effect of the map constructed by the improved method has been im- proved at the edge, with a lower degree of blur.



(a)

(b)

Fig 6: Map built with (a)b1-2014-10-07-12-34-51 and (b)b2-2016-03-15-14-23-01 using Cartographer and the proposed method.

figure 7 and figure 8 is the pose estimation gotten form ORB-SLAM. Correct pose estimation needs the data from other sensors. The tracking is correct, and it can detect the corner turning, but the estimation of the distance is inaccurate.

Fig 7: ORB-SLAM Starting point.

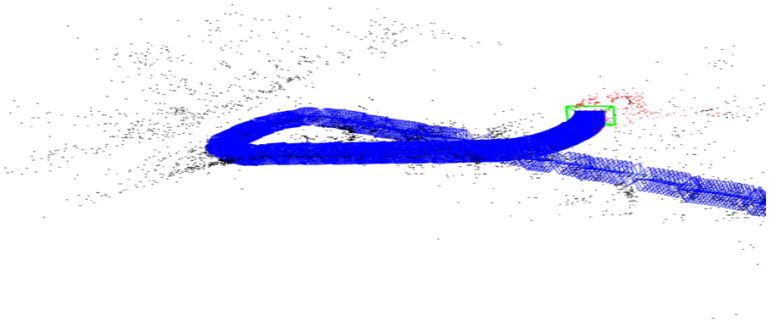


Fig 8: Results of ORB-SLAM.

VI. Conclusion

A highway intelligent service robot system based on multi sensor fusion especially camera and lidar is designed and implemented in this paper. The experimental results show that the robot can meet the needs of the scene to a certain extent in aspects of map construction. The accuracy of the sensors will be improved in the future, and visual SLAM which includes more information allow the robot to adapt different environments and deal with more complex situations, but the accuracy of the visual SLAM needs to be enhanced in the future.

Acknowledgments

This research is partially funded by the National Natural Science Foundation of China (No.61673348), the Hangzhou Agricultural Scientific Research Project (No.20180432B15), Public Welfare Projects in Zhejiang Province (No.LGF20F030002), Zhejiang Natural Science Foundation Youth Fund (NO. LQ18F030009), and Project of Zhejiang Science and Technology Bureau (NO. 20201203B96).

References

[1] R. Mur-Artal, J.M.M. Montiel, and J.D. Tardos, "ORB- SLAM: a Versatile and Accurate Monocular SLAM System," IEEE Transactions on Robotics,https://doi.org/ 10.1109/TRO.2015.2463671, vol. 31, no. 5, pp. 1147-1163.

[2] R. Smith, M. Self, P. Cheeseman, "Estimating Uncertain Spatial Relationships in Robotics," Machine Intelligence & Pattern Recognition, https://doi.org/ 10.1007/978-1-4613-8997-2_14, vol. 5, no. 5, pp. 435-461

[3] T. Bailey, J. Nieto, J. Guivant, "Consistency of the EKF-SLAM algorithm," IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, https://doi.org /10.1109/IROS.2006.281644, 2006.

[4] K. Murphy, S. Russell, Rao-Blackwellised, "Particle Filtering for Dynamic Bayesian Networks," Sequential Monte Carlo Methods in Practice, https://doi.org /10.1007/978-1-4757-3437-9_24.

[5] G. Grisettiyz, C. Stachniss, W. Burgard, "Improving grid-based SLAM with Rao-Blackwellized particle filters by adaptive proposals and selective resampling," Proceedings - IEEE International Conference on Robotics and Automation, https://doi.org /10.1109/ROBOT, 2005.

[6] G. Grisetti, C. Stachniss, W. Burgard, "Improved techniques for grid mapping with Rao-Blackwellized particle filters," IEEE Transactions on Robotics, https://doi.org /10.1109/TRO.2006.889486, vol. 23, no. 1, pp. 34-46.

[7] J.S. Gutmann, K. Konollge, "Incremental Mapping of Large Cyclic Environments," International Symposium on Computational Intelligence in Robotics and Automation, Monterey, CA, USA, https://doi.org /10.1109/CIRA.1999.810068.

[8] W. Hess, D. Kohler, H. Rapp, "Real-time loop closure in 2D LIDAR SLAM," IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, https://doi.org /10.1109/ICRA.2016.7487258, 2016.
[9] J. Engel, T. Sch, D. Cremers, "LSD-SLAM: Large-Scale Direct Monocular SLAM," European conference on computer vision, https://doi.org /10.1007/978-3-319-10605-2_54.
[10] K. Yousif, A. Bab-Hadiashar, R. Hoseinnezhad, "An overview to visual odometry and visual SLAM: Applications to mobile robotics," Intelligent Industrial Systems, https://doi.org /10.1007/s40903-015-0032-7, vol. 1, no. 4, pp. 289-311.
[11] Y. Zhang, R. Jin, Z.H. Zhou, "Understanding bag-of-words model: a statistical framework," International Journal of Machine Learning and Cybernetics, https://doi.org /10.1007/s13042-010-0001-0, vol. 1, no. 1-4, pp. 43-52.
[12] A. Dwijotomo, M. Rahman, M. Ariff, et al., "Cartographer slam method for optimization with an adaptive multi- distance scan scheduler," Applied Sciences, https://doi.org/10.3390/app10010347, vol. 10, no. 1, pp. 347, 2020.
[13] E. Olson, 2009. "Real-time correlative scan matching," IEEE International Conference on Robotics and Automation, Kobe, Japan, https://doi.org/10.1109/ROBOT.2009.5152375, pp.1050—4729, 2009.
[14] J. Wen, C. Qian, J. Tang, et al., "2d lidar slam back-end optimization with control network constraint for mobile mapping," Sensors, https://doi.org/10.3390/s18113668, vol. 18, no. 11, pp.1-14, 2018.
[15] K. Konolige, G. Grisetti, R. Kümmerle, et al., "Efficient sparse pose adjustment for 2d mapping," 2010.
[16] G. Sergio, M.E. L. Guillen, R. Barea, "Indoor SLAM for Micro Aerial Vehicles Control using Monocular Camera and Sensor Fusion," International Conference on Autonomous Robot Systems and Competitions (ICARSC), Bragança, Portugal, https://doi.org/10.1109/ICARSC.2016.46, 2016.