

Received August 18, 2020, accepted August 23, 2020, date of publication August 26, 2020, date of current version September 9, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3019659

Research on SLAM Algorithm of Mobile Robot Based on the Fusion of 2D LiDAR and Depth Camera

LILI MU¹, PANTAO YAO¹, YUCHEN ZHENG², KAI CHEN¹, FANGFANG WANG¹, AND NANA QI¹

¹School of Mechanical Engineering, Anhui University of Science and Technology, Huainan 232001, China

²School of Mathematics and Statistics, The University of Sydney, Sydney, NSW 2006, Australia

Corresponding author: Pantao Yao (2575249103@qq.com)

ABSTRACT This paper proposes a new Simultaneous Localization and Mapping (SLAM) method on the basis of graph-based optimization through the combination of the Light Detection and Ranging (LiDAR), RGB-D camera, encoder and Inertial Measurement Unit (IMU). It can conduct joint positioning of four sensors by taking advantage of the unscented Kalman filter (UKF) to design the related strategy of the 2D LiDAR point cloud and RGB-D camera point cloud. 3D LiDAR point cloud information generated by the RGB-D camera under the 2D LiDAR has been added into the new SLAM method in the sequential registration stage, and it can match the 2D LiDAR point cloud and the 3D RGB-D point cloud by using the method of the Correlation Scan Matching (CSM); In the loop closure detection stage, this method can further verify the accuracy of the loop closure after the 2D LiDAR matching by describing 3D point cloud. Additionally, this new SLAM method has been verified feasibility and availability through the processes of theoretical derivation, simulation experiment and physical verification. As a result, the experiment shows that the multi-sensor SLAM framework designed has a good mapping effect, high precision and accuracy.

INDEX TERMS SLAM, RGB-D, graph-based optimization, multi-sensor fusion.

I. INTRODUCTION

SLAM technology is one of essential technologies to achieve autonomous movement of mobile robots, widely applied into mobile robots, self-driving, drones, and autonomous underwater vehicles, etc. Up to now, SLAM technology has been developed for more than 30 years. And the theory on SLAM has now been a relatively mature stage. Visual simultaneous localization and mapping (VSLAM) and laser SLAM have achieved remarkable results. Laser SLAM relies on a high-precision laser sensor to scan surroundings and build a map. Although it is highly accurate for LiDAR, the range of perception on it is short (about 150m) which is highly sensitive to the environment and easy to create noise and leakage. VSLAM is currently a hot research topic in the SLAM field. Many researchers pay more attention on VSLAM because it involves a large amount of information and its price is quite cheap. However, due to the fact that its angle and field of view is small, it is easier to be affected by the changes of environmental light, dynamic objects, and other

factors. Therefore, when single-sensor SLAM technology comes across many difficulties. The addition of Multi-sensor will greatly improve the accuracy of the whole sensing system which could be the mainstream direction of SLAM technology in the future.

II. RELATED WORK

SLAM was first proposed by Cheeseman, Self, and Smith at the 1986 California Robotics and Automation Conference [1]. The author mentioned in [2]–[4] first introduced the probabilistic method into solving the SLAM problem. He can use the probability distribution to represent the noise and attitude caused by the robot motion to solve the robot state estimation problem, which has achieved good results and pointed out the direction for solving the problem. Subsequently, the research on solving the SLAM problem through probabilistic methods has achieved rapid development. The authors in [5], [6] apply the idea of particle filtering to solve the SLAM problem and propose the Rao-Blackwellised Particle Filter (RBPF). The authors in [7]–[9] build on the Rao-Blackwellised particle filter by proposing a more approximating the proposed distribution of the true posterior

The associate editor coordinating the review of this manuscript and approving it for publication was Heng Wang.

distribution and using adaptive resampling technology developed Fast-SLAM 1.0. Fast-SLAM 2.0 and achieved good results. Although the application of the filtering technique can solve the SLAM problem quickly, a large amount of data is discarded during the computation, and when constructing the global map there is no guarantee of getting a globally consistent map. Therefore, the authors in [10] propose to solve the global SLAM problem by optimization based on factorial graphs. The method divides the SLAM problem into a data correlation at the front end and an optimization problem at the back end, where the SLAM back end is converted into a least-squares problem. The problem is linearized by expanding the objective function through a first-order Taylor formula and then solved using an iterative method. However, due to the requirement for Jacobi matrices, the matrices explode as the number of data increases, leading to an exponential increase in computational complexity. By not being able to satisfy real-time, the method was less practical at first. With the sparsity of the SLAM problem being applied to the solution process [11]–[15], the graph-based optimization method has been used in terms of computational speed and filtering. The methods are already similar and can meet the real-time requirements of the algorithm. In addition to graph-based optimization, algorithms can build more accurate and globally consistent maps in large-scale environments, graph-based optimization SLAM. The method is gradually replacing the filter method as the dominant method today.

SLAM algorithms initially mainly used traditional sensors with few observations, such as laser and sonar, and the lack of information made the performance of the algorithm being restricted, the fusion of multiple sensors has been a focus of researchers. The authors in [16] applied Kalman filtering for data fusion and proposed a Kalman filter-based prediction and updated at the level of the fusion method. Establishing on the Bayesian fusion method, the authors in [17] propose a way to perform sensory datasets to generate an environment occupancy. The new method of raster mapping was tested on a mobile robot with eight ultrasonic waves. The authors in [18] obtained measurements from mileage sensors, sonar sensors and will assume that Gaussian measurement noise with extended Kalman filtering apparatus was compared with a particle filter that made no assumptions about the measured noise distribution and showed that the particle filter was better than the extended Kalman filter. The apparatus has better performance but at the cost of higher computational requirements. The authors in [19] took the autonomous underwater vehicle (AUV) equipped with Mechanical Scanning Imaging Sonar (MSIS) as the main sensor of the research object and introduced a method based on the Smooth Variable Structure Filter (SVSF). This filter merges the information from various sensors (Doppler Velocity Logs (DVL), the MTi motion reference unit (MRU)), and the observations from the feature extraction based line to estimate the vehicle's motion and to construct a map in partially structured environments. The proposed algorithm is validated and provides an accurate estimation which gives good robustness.

Due to sensor characteristics, even with EKF or particle filtering and fusion data, the SLAM algorithm is still not sufficient for complex environments. It is not possible to create an accurate map of the environment. Visual imagery is rich in information which can provide a large number of useful features, and with the development of computer technology, it is becoming SLAM Hot Topics in Domain Research. The authors of [20] developed MonoSLAM, the first monocular vision SLAM system to run online in real-time. The algorithm uses the EKF framework to create sparse feature point maps online. In the same year, the authors of [21] proposed a parallel tracking and map construction method (Parallel tracking and mapping (PTAM) algorithm and open source code. That was the first time for changing the back-end optimization from filter to nonlinear optimization and dividing the system into two separate subroutines, motion tracking, graph building and running on two threads simultaneously which further improved the efficiency of the algorithm. VSLAM then entered a period of rapid development and a large number of excellent algorithmic frameworks emerged, ORB-SLAM [22], [23], DTAM [24], LSD-SLAM [25]–[27], SVO [28] and DVO-SLAM [29].

Caused by increasing complexity of the robot's environment, vision-based or laser-based sensors often fail to locate the robot in time and inaccurate construction of maps. Currently, in the field of SLAM, the greater number of researchers are focusing on multi-sensor fusion SLAM techniques. The authors of [30] fused LiDAR and binocular camera information, using an improved RBPF algorithm, in calculating the proposed distribution of the fusion and vision information, LiDAR information and odometer information of mobile robots improve SLAM algorithms for mobile robots' robustness. The authors in [30] combined LiDAR and vision cameras to fuse laser point clouds with image feature points by adjusting sparse attitude to optimize the robot's position pose, while using the word bag model for loop closure detection and then adding constraints to further optimize the position pose, the algorithm in the results show that the positioning accuracy of the algorithm is improved and the accuracy of loop closure detection is further improved, environmental maps are more accurate than a single laser map construction. The author in [32] proposed a hybrid beam adjustment visual-inertial positioning framework based on the geometric information of the laser image. The laser image mapping was performed at the same time as the visual image mapping, and the two maps were aligned to improve the accuracy of the image creation. The algorithm is very robust when the environment changes. Kumar *et al.* [33] described a methodology to fuse data available from multiple sensors for improved localization performance under conditions of collinear ambiguity. Jin *et al.* [34] fused visual stereo image and 2D LiDAR data and uses loop closure for accurate odometry estimation.

Generally speaking, compared to the single-sensor SLAM, the mapping accuracy and real-time performance of the SLAM fusion of LiDAR and RGB-D camera have been

improved. The accuracy of LiDAR is high, usually at the millimeter level, but only partial information can be obtained. RGB-D camera can get depth information, but the direct use of its point cloud data will lead to limited use scenes, which can only be used in small scenes and the error of the point cloud increases sharply with the distance. The point cloud data at the edge of the credibility is extremely low and if it is directly matched, the confidence of the robot's pose is also relatively low. Therefore, a new fusion strategy of 2D LiDAR point cloud and RGB-D camera point cloud is proposed in this paper. Four kinds of sensor systems are used and the UKF is used for joint positioning and the theoretical and experimental studies are carried out.

III. MULTI-SENSOR SYSTEM FOR MOBILE ROBOT

A. KINEMATIC MODEL

In this paper, a mobile robot was built with four driving wheels. The kinematic model is the same as that of the two-wheeled differential mobile robot, so the analysis is simplified and the two-wheel differential model is adopted. The differential model of the robot is shown in Fig. 1.

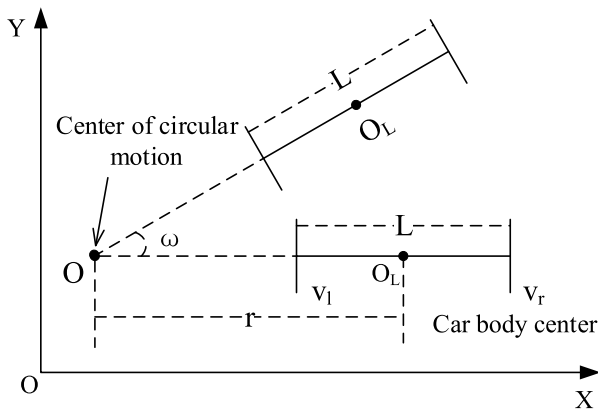


FIGURE 1. Arc motion model of the differential robot.

The robot motion model is approximated by circular motion, the point O is the center of motion, O_L is the center of the mobile robot body, the radius of the circular motion is represented by r , the diameter of the car body is represented by L , and the angular velocity and linear velocity of the left and right wheels are respectively for ω_l, ω_r and v_l, v_r . The angular velocity of the car body during the movement is ω , which can be obtained according to the kinematic relationship:

$$\omega = \frac{v_l}{r - \frac{L}{2}} = \frac{v_r}{r + \frac{L}{2}} \quad (1)$$

$$v_l \left(r + \frac{L}{2} \right) = v_r \left(r - \frac{L}{2} \right) \quad (2)$$

$$(v_r - v_l)r = (v_r + v_l) \frac{L}{2} \quad (3)$$

$$r = \frac{(v_r + v_l) \frac{L}{2}}{(v_r - v_l)} \quad (4)$$

From the formula (4), when the speed of the left and right wheels is equal, the denominator is infinitesimal, and the

motion radius r of the robot will be infinite. At this time, the robot will move in a straight line; when the speed of the left and right wheels is not equal, the robot will drive in a certain direction to realize turning.

According to r can obtain:

$$r + \frac{L}{2} = \frac{(v_r + v_l) \frac{L}{2}}{(v_r - v_l)} + \frac{(v_r - v_l) \frac{L}{2}}{(v_r - v_l)} = \frac{v_r L}{(v_r - v_l)} \quad (5)$$

Substituting into (3) gives:

$$\omega = \frac{(v_r - v_l)}{L} \quad (6)$$

According to the relationship of angular velocity and linear velocity:

$$v = \omega r = \frac{(v_r - v_l)}{L} \frac{(v_r + v_l) \frac{L}{2}}{(v_r - v_l)} = \frac{(v_r + v_l)}{2} \quad (7)$$

Therefore, during the movement of the differential car, as long as you know the speed of the left and right wheels, you can find the speed (v, ω) of the robot at the current moment. Generally, the wheel speed of the mobile robot is obtained by the encoder.

B. ODOMETER MODEL

As shown in Fig. 2, the state model of the robot at adjacent times in the global coordinate system assumes that the wheel of the robot contacts with the ground at any time point, and there is no relative sliding. The left wheel and right wheel of the mobile robot are respectively installed with wheel encoder. Based on the above conditions, the kinematics model of the robot was established.

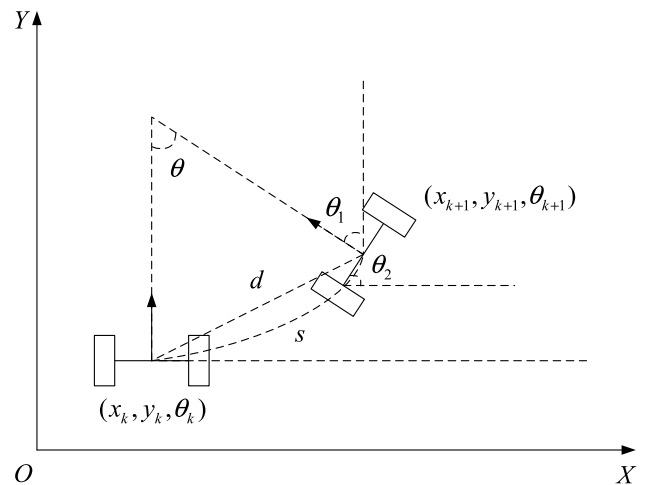


FIGURE 2. Kinematics model of the mobile robot.

In Fig. 3, the pose of the robot at k is $p_k = (x_k, y_k, \theta_k)^T$, the velocity of the robot is expressed as (v, ω) , θ_1 is the angle of the change of the robot's head directly at the time of $k + 1$, θ_2 is the angle of the wheel rotation, θ is the angle of the mobile robot turning in a circular motion, s is the arc of the robot's movement, d represents the relative distance between

the two adjacent moments of the robot. In a very short time Δt , $d \approx s$. Then:

$$\begin{cases} \theta = \omega \Delta t \\ d \approx s = v \Delta t \end{cases} \quad (8)$$

The pose of the robot at $k+1$ is expressed as p , which can be obtained by the odometry model [35]:

$$\begin{cases} x_{k+1} = x_k + d \cos(\theta_k + \frac{\theta}{2}) \\ y_{k+1} = y_k + d \sin(\theta_k + \frac{\theta}{2}) \\ \theta_{k+1} = \theta_k + \theta \end{cases} \quad (9)$$

Substituting (10) into the above formula gives:

$$\begin{cases} x_{k+1} = x_k + v \Delta t \cos(\theta_k + \frac{\omega \Delta t}{2}) \\ y_{k+1} = y_k + v \Delta t \sin(\theta_k + \frac{\omega \Delta t}{2}) \\ \theta_{k+1} = \theta_k + \omega \Delta t \end{cases} \quad (10)$$

According to the above formula, the pose of the robot at the next moment can be calculated according to the velocity and time of the robot and the pose of the robot at the last moment.

IV. MULTI-SENSOR SYSTEM EXTERNAL REFERENCE CALIBRATIONS

A. PRINCIPLE OF JOINT LiDAR AND ODOMETER CALIBRATION

The mobile robot runs on a 2D plane, the odometer coordinate system and the robot coordinate system coincide, because it is not possible to measure their motion of the tilt angle and roll angle, so it is assumed that the motion is on the horizontal surface, the robot coordinate system and LiDAR coordinate system are maintained horizontal position. Assuming that $SE(2)$ is a special Euclidean group of plane motion, $Se(2)$ represents its Lie algebra, then the robot's pose relative to the world coordinate system is expressed as $q = (q_x, q_y, q_\theta)$, and the position of the LiDAR coordinate system relative to the robot coordinate system is expressed as $t = (t_x, t_y, t_\theta) \in SE(2)$. As shown in Fig. 3, $q^k \in SE(2)$ represents the pose

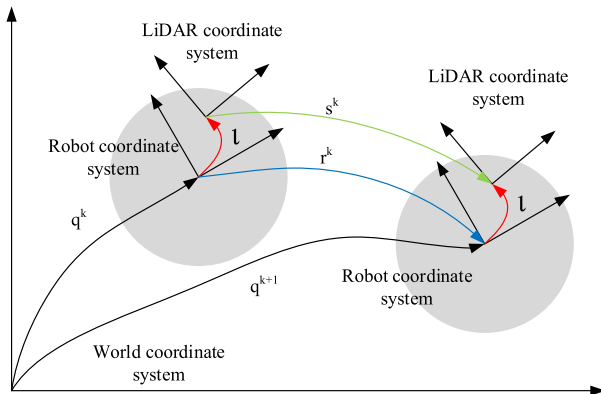


FIGURE 3. The relative pose of robot motion.

of the robot for the world coordinate system, $t \in SE(2)$ represents the pose of the LiDAR for the robot coordinate system, $r^k \in SE(2)$ represents the pose of the robot at adjacent times, and $s^k \in SE(2)$ represents the pose between sensors at the adjacent time.

As described in Chapter 3, encoders are installed on the two wheels of the differential robot to measure the rotation speed of the left and right wheels, then the robot's motion state (v, ω) can be expressed by (6) and (7) obtained. For the convenience of description, it is written as:

$$\begin{pmatrix} v \\ \omega \end{pmatrix} = J \begin{pmatrix} v_l \\ v_r \end{pmatrix} \quad (11)$$

where J is the parameter matrix of the car body itself, expressed as:

$$J = \begin{pmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{pmatrix} = \begin{pmatrix} +\frac{r_l}{2} & +\frac{r_r}{2} \\ -\frac{r_l}{L} & +\frac{r_r}{L} \end{pmatrix} \quad (12)$$

where r_l and r_r represent the left and right wheel radius, and L represents the distance between the two wheels. When the robot is running, there is a load, plus the error of its measurement, and the radius of the left and right wheels is different from the nominal one. Therefore, it is also necessary to calibrate the two-wheel radius.

Define \oplus and \ominus as operations in $SE(2)$, then the pose of the LiDAR coordinate system relative to the world coordinate system at any time is expressed as $q^k \oplus t$, and \oplus indicates the addition of $SE(2)$:

$$\begin{pmatrix} a_x \\ a_y \\ a_\theta \end{pmatrix} \oplus \begin{pmatrix} b_x \\ b_y \\ b_\theta \end{pmatrix} = \begin{pmatrix} a_x + b_x \cos(a_\theta) - b_y \sin(a_\theta) \\ a_y + b_x \sin(a_\theta) + b_y \cos(a_\theta) \\ a_\theta + b_\theta \end{pmatrix} \quad (13)$$

\ominus indicates the inverse of $SE(2)$:

$$\ominus \begin{pmatrix} a_x \\ a_y \\ a_\theta \end{pmatrix} = \begin{pmatrix} -a_x \cos(a_\theta) - a_y \sin(a_\theta) \\ +a_x \sin(a_\theta) - a_y \cos(a_\theta) \\ -a_\theta \end{pmatrix} \quad (14)$$

Then at two adjacent time's k and $k+1$, the relative change between the LiDAR coordinate system can be obtained as:

$$s^k = \ominus(q^k \oplus t) \oplus (q^{k+1} \oplus t) \quad (15)$$

The relative changes between robot coordinate systems obtained by odometer through track calculation are recorded as follows:

$$r^k = \ominus q^k \oplus q^k \quad (16)$$

Substituting (22) into (21), we can get the following results:

$$s^k = \ominus t \oplus r^k \oplus t \quad (17)$$

The relative pose of the LiDAR coordinate system in the above formula is computed by calculating the relative pose of the robot coordinate system by the odometer. The relative pose of the neighboring moments can also be obtained by the data scan matching of the LiDAR itself, which is recorded as

\hat{s}^k , The problem of joint calibration of odometer and LiDAR is expressed as the speed ($v_l(t)$, $v_r(t)$) of the two left and right wheels of the robot at a given time $t \in [t_1, t_n]$ and the relative pose $\hat{s}^k (k = 1, \dots, n - 1)$ of the LiDAR coordinate system obtained by LiDAR scan matching, to find a maximum likelihood estimation of the parameter r_l, r_r, L, t . The objective function \mathfrak{S} is recorded as follows:

$$\mathfrak{S} = -\frac{1}{2} \sum_{k=1}^n \left\| \hat{s}^k - \Theta t \oplus r^k \oplus l \right\|^2 \quad (18)$$

B. CALIBRATION PRINCIPLE OF VISION SENSOR AND LiDAR

The RGB-D camera has two lenses, and the external parameters between the lenses are known. Therefore, when performing external reference calibration, any one of these lenses can be used. Here, an RGB camera is used. The camera model can be represented by a pinhole model. A certain point p in space is observed by the camera and LiDAR at the same time, its coordinate is defined as $p^p = (u, v)^T$ in the camera pixel plane, its coordinate is defined as $p^i = (x^i, y^i)$ in the image plane, and its coordinate is defined as $p^c = (x^c, y^c, z^c)^T$ in the camera coordinate system.

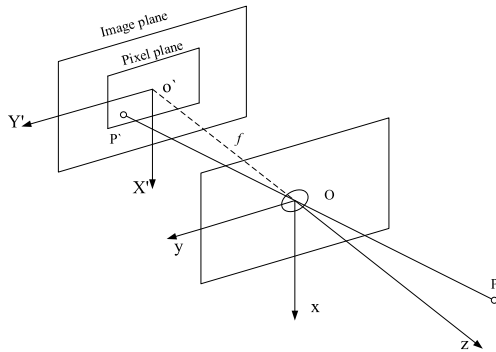


FIGURE 4. Pinhole camera model.

According to the similarity relationship of triangles, there are:

$$\begin{cases} x^i = f \times \frac{x^c}{z^c} \\ y^i = f \times \frac{y^c}{z^c} \end{cases} \quad (19)$$

Where f represents the focal length of the camera and the unit is m. The image is mapped to the pixel coordinate system to obtain:

$$\begin{cases} u = f_x \times \frac{x^c}{z^c} + c_x \\ v = f_y \times \frac{y^c}{z^c} + c_y \end{cases} \quad (20)$$

where f_x and f_y are the focal length, but the units are pixels. c_x and c_y are the offset of the pixel coordinate system relative to the image coordinate system. Move z to the left and use

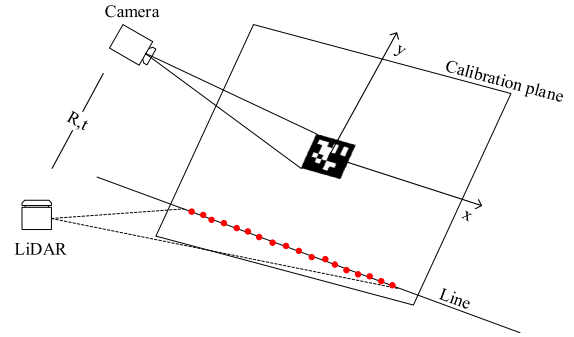


FIGURE 5. Calibration system.

homogeneous coordinates to write it in matrix form:

$$z^c \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x^c \\ y^c \\ z^c \end{pmatrix} = KP^c \quad (21)$$

where the matrix K is called the internal parameter matrix of the camera.

The external parameters are represented by a 3×3 rotation matrix R and a 3×1 translation vector t . Taking the LiDAR coordinate system as the reference, the camera coordinates of the point p are obtained by transforming the LiDAR coordinates according to the external parameters, namely:

$$p^l = Rp^c + t \quad (22)$$

where $p^l = (x^l, y^l, z^l)^T$ is the coordinate of a point p in the LiDAR coordinate system.

In the actual external reference calibration process, the calibration board is usually used as the calibration tool. At the same time, the calibration board appears in the camera and LiDAR field of view. In the LiDAR coordinate system, the LiDAR plane intersects the calibration plane, and the LiDAR spot falls on the calibration board, generating a straight line. In the camera coordinate system, the Perspective-n-Point (PnP) algorithm can be used to solve. For the depth camera, since the 3D information of the spatial point can be output, the point cloud information can be used to directly fit the plane equation through the RANdom SAMple Consensus (RANSAC) algorithm.

In the world coordinate system, the calibration plane is represented as a plane with $Z = 0$. Correspondingly, in the camera coordinate system, the calibration plane is represented by a 3×3 vector N . The vector N is perpendicular to the calibration plane and its length $\|N\|$ is equal to the distance from the camera to the plane.

According to the constraint, the LiDAR point falling on the calibration plate is on the calibration plane. For the LiDAR point p_i^l , the coordinate $p_i^c = R^{-1}(p_i^l - t)$ in the camera coordinate system can be obtained by using (22). If $N \cdot P^c = \|N\|^2$ is satisfied, then:

$$N \cdot R^{-1}(p_i^l - t) = \|N\|^2 \quad (23)$$

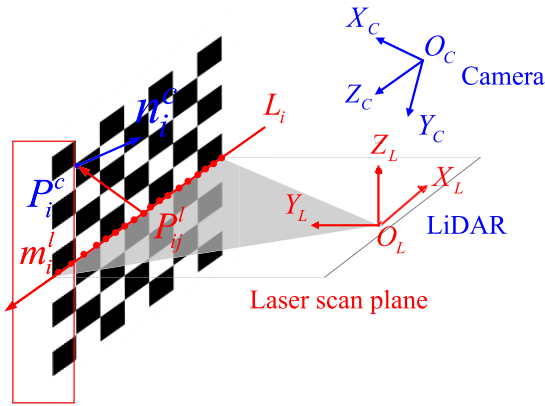


FIGURE 6. Calibration initial value solution system.

Construct an error function and solve it through an optimized method. The error function ξ represents the residual sum of the distance between each LiDAR point of the calibration plate and the calibration plane under the camera coordinate system on different poses:

$$\xi = \sum_i \sum_j \left(\frac{N_i}{\|N_i\|} \cdot (N \cdot R^-(p_{ij}^l - t) - \|N\|^2) \right) \quad (24)$$

where N_i represents the equation of the calibration plane in the pose i . The rigid body transformation between the camera and LiDAR is solved by minimizing the error function.

Using nonlinear optimization functions requires providing an initial value, and the initial value should be as close to the true value as possible to avoid falling into the local extremum. Because of the relative transformation of three-dimensional space, the initial value obtained by the manual measurement is unreliable and the error is large. By constructing linear equation solving, it is possible to provide an initial value for optimization.

In the LiDAR coordinate system, the LiDAR falls on the calibration board at the point \vec{p}^l . The calibration board is bounded, and the LiDAR points that fall on the edge of the calibration board can be obtained. In the camera coordinate system, the depth camera hits the calibration board to generate a 3D point cloud. The points on the edge of the calibration plate are fitted with a straight line L^b , whose normal vector is n^{lb} , the point distance on the line L^b meets the $d = 0$ relationship, there are:

$$d = \frac{\vec{p}^l p^c \times n^{lb}}{|n^{lb}|} = 0 \quad (25)$$

By directly solving this linear equation, (R_0, t_0) can be found as the initial solution for the optimization.

V. MULTI-SENSOR SLAM STATE ESTIMATION BASED ON UKF

A. APPLICATION OF UKF IN SLAM

In the nonlinear system Kalman filter SLAM problem, the state transfer function $P(x_i|x_{i-1}, u_i)$ is the system model,

and its functional form is usually:

$$X_t = f(X_{t-1}, u_{t-1}, w_t) \quad (26)$$

The system equation describes the changes that the system takes under the control input u_t command, where X_t is the system state quantity, and the random variable w_t is a Gaussian random variable, which represents the uncertainty introduced by the state transition, the mean value is 0, and the variance is R_t , expressed as $w_t \sim (0, R_t)$.

The update phase mainly uses observations to modify the state quantity, and the observation density $p(z_t|x_t)$ is expressed as:

$$Z_t = h(X_t, v_t) \quad (27)$$

where h is the nonlinear function of the observed measurement, v_t is the measurement noise, and the distribution satisfies the Gaussian distribution with the mean value of 0 and the variance of Q_t .

The observation model conforms to the pose measurement model, and the model equation is:

$$z_t = x_t + v_t \quad (28)$$

where the relationship between the observed value and the real state is linear, v is the observation noise, which conforms to the Gaussian distribution $v \sim N(0, V)$, and is given by the odometer. The wheel odometer and laser odometer are following the pose measurement model, which can be used to update the state variables in the update phase.

Based on the UKF, this paper designs a mobile robot positioning system fused with multi-sensor information, as shown in Fig. 7. The IMU measures the posture of the robot, while the wheel encoder measures the position of the robot, comprehensively obtains the current pose of the robot, and inputs it into the filter for measurement update. The update frequency of the IMU and the encoder is relatively high, which can reach 20Hz. The combined pose and scan matching provided to the point cloud was used as the initial pose, which accelerates the matching speed of the point cloud and improves the accuracy of the match. The optimized pose obtained by point cloud matching is sent to the filter as the measured value, and the measurement equation is updated. With the continuous entry of sensor data, the filter can be updated in a loop closure.

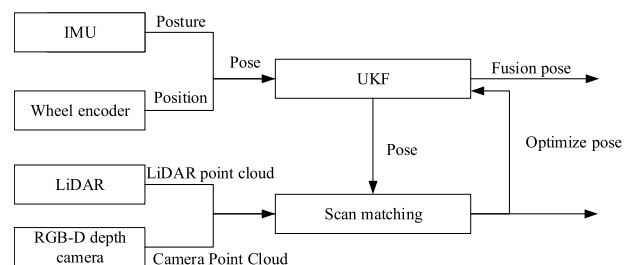


FIGURE 7. Multi-sensor positioning system framework.

B. EXPERIMENT AND ANALYSIS

A simulation experiment was performed on the UKF algorithm. At the same time, the robot positioning based on Extended Kalman Filter (EKF) was tested. The simulation environment is shown in Fig. 8, simulating the indoor environment of 8 m × 8 m. The initial position is set manually. The simulation robot starts from the initial position and moves along the specified route. The running speed is 0.25 m/s, the maximum angular velocity is 20 ° / s, the angle range of the robot is - 30 ° to 30 ° and the speed noise is 0.1m/s and the rotation noise is 2 °.

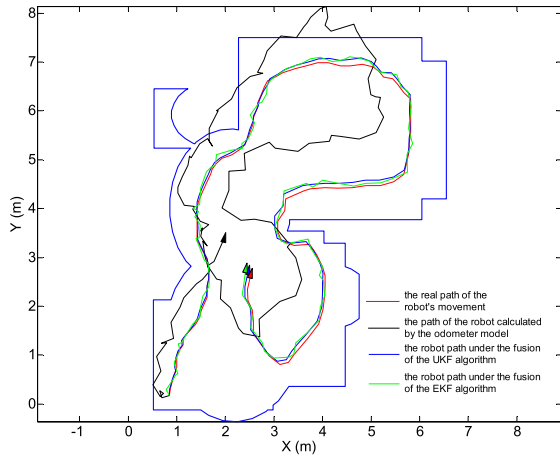


FIGURE 8. Simulation scenario.

As shown in Fig. 8, the red line represents the real path of the robot’s movement, and the black line represents the path of the robot calculated by the odometer model.

As can be seen from Fig. 8, the path and real path deviation calculated by the odometer model are large in the case of noise. The green segment in the figure represents the robot path under the fusion of the EKF algorithm, and the blue segment represents the robot path under the fusion of the UKF algorithm. The robot motion path and the real path obtained by the two fusion algorithms are more consistent, however, the path based on the fusion algorithm of UKF is more accurate than the path derived from the fusion algorithm based on EKF. The specific positioning error of the two algorithms is shown in Fig. 9.

It can be seen from Fig. 9 that in the X and Y directions, the positioning error of the EKF algorithm is higher than that of the UKF algorithm, the former positioning deviation on the X-axis and Y-axis are 0.262 m and 0.3085 m, the latter positioning deviation on the X-axis and Y-axis are 0.147m and 0.206 m. The performance of the fusion algorithm based on UKF is better than the fusion algorithm based on EKF. Table 1 lists the root mean square errors of the two algorithms in X-axis and Y-axis, further illustrating that the fusion algorithm based on UKF is more accurate than the fusion algorithm based on EKF.

The algorithm was tested on a real robot, and the experimental platform is as described in the previous two chapters.

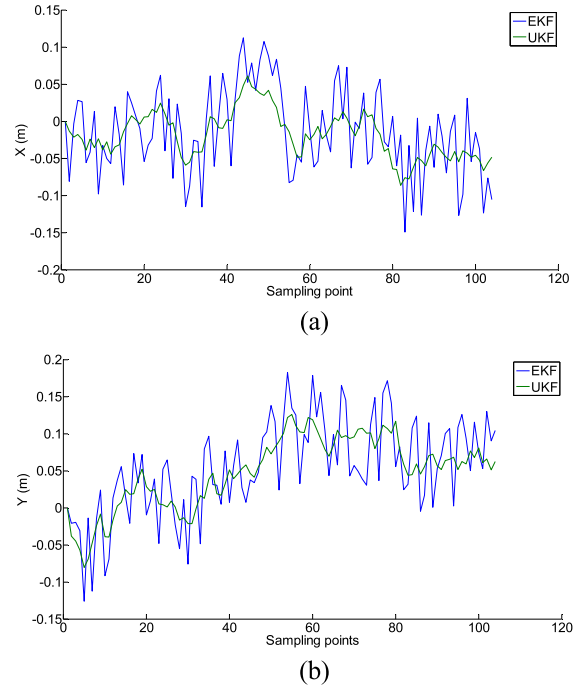


FIGURE 9. Positioning error. (a) Error in the X-direction. (b) Error in the Y-direction.

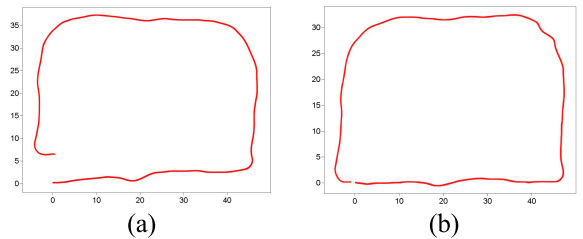


FIGURE 10. Loop closure detection test of an actual robot. (a) Pure laser SLAM loop closure test. (b) UKF fusion SLAM loop closure test.

Since the real position of the robot cannot be measured, the robot has carried out a loop closure experiment. Starting from the starting point, it moves in a large circle and then returns to the starting point. The performance of the algorithm is judged by judging the position of the starting point and the ending point of the robot. Record the relative coordinates of the robot, as shown in Fig. 10. Fig. 10(a) is the robot position matched by the LiDAR scan. It can be seen from the picture that the difference between the end position of the robot and the starting point is large, and there is no loop closure. The maximum deviation is calculated by the

TABLE 1. Comparison of simulation results.

Algorithm	RMS in the X direction	RMS in the Y direction
EKF	0.0837	0.1435
UKF	0.0360	0.0677
Error ratio	124%	120%

head-to-tail position difference and the displacement ratio of the entire path. The maximum deviation has reached 2.9%. Fig. 10(b) is the position of the UKF fusion robot. The robot has returned to the starting point with an error of less than 1%. Experiments show that the multi-sensor fusion positioning scheme based on the UKF algorithm has high positioning accuracy.

VI. SIMULTANEOUS LOCATION AND MAP CONSTRUCTION OF MULTI SENSOR SYSTEM

A. SLAM FRAMEWORK BASED ON GRAPH-BASED OPTIMIZATION

Slam is essentially a state estimation problem, which uses sensor information to estimate the robot's pose and environment. As shown in Fig. 11, the slam method based on graph-based optimization divides the SLAM problem into two parts: front-end and back-end. The front-end is responsible for the construction of the pose map, and the back-end adjusts the configuration of the graph through optimization.

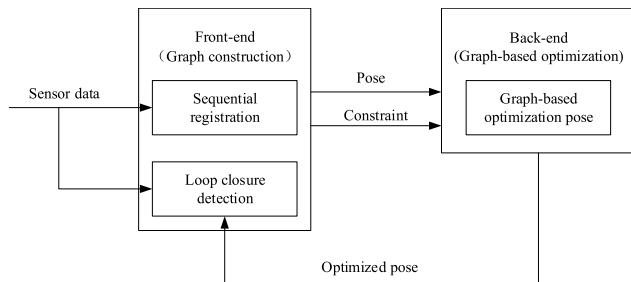


FIGURE 11. SLAM framework based on graph-based optimization.

The premise of the back-end optimization is the correct construction of the front-end pose map, and the accuracy of the map construction directly affects the results of the back-end optimization. Wrong nodes and edges will cause the optimization result to deviate from the true value, resulting in an error in the final pose, which will affect the construction of the map. The research object of this paper is mainly 2D LiDAR and RGB-D camera. The following data fusion of 2D LiDAR and RGB-D camera for front-end data registration and loop closure detection is used to improve the accuracy of pose image.

B. CSM REGISTRATION FOR FUSION DEPTH POINT CLOUDS AND LiDAR POINTS

Generally speaking, the detection distance of the depth camera is limited, and the direct use of its point cloud data will lead to the limited use of the scene and can only be used in small scenes. Moreover, the error of the point cloud increases rapidly with the distance, and the credibility of the point cloud data at the edge is very low. If it is directly matched, the robot's pose confidence is also relatively low. The detection range of LiDAR is much longer than that of an RGB-D camera, and the accuracy is very high, usually in millimeter level. Therefore, this paper proposes a point

cloud fusion strategy, which fuses 2D LiDAR point cloud and RGB-D camera point cloud.

Assuming that $S = \{s_i\}_{i=1,\dots,I}$, $s_i \in R^2$ represents a 2D LiDAR point cloud, $P = \{p_j\}_{j=1,\dots,J}$, $p_j \in R^3$ represents a LiDAR point cloud converted by an RGB-D camera, and the rigid body transformation between the robot's relative pose is represented by T_ζ , then in the pose ξ , point cloud S and point cloud P are converted to The submap coordinate system is expressed as:

$$T_\zeta s = \begin{pmatrix} \cos \zeta_\theta & -\sin \zeta_\theta \\ \sin \zeta_\theta & \cos \zeta_\theta \end{pmatrix} p + \begin{pmatrix} \zeta_x \\ \zeta_y \end{pmatrix} \quad (29)$$

$$T_\zeta p = \begin{pmatrix} \cos \zeta_\theta & -\sin \zeta_\theta & 0 \\ \sin \zeta_\theta & \cos \zeta_\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} + \begin{pmatrix} \zeta_x \\ \zeta_y \\ 0 \end{pmatrix} \quad (30)$$

Correlation scan matching (CSM) was used to obtain the optimal pose ζ^* . Due to the fusion of LiDAR data and RGB-D camera data, the score is expressed as follows:

$$Score = \omega_s S_s + \omega_p S_p \quad (31)$$

where S_s is the score of 2D LiDAR point, ω_s is its score weight, S_p is the point cloud score of RGB-D camera, and ω_p is its weight. The point clouds of 2D LiDAR and RGB-D camera are unified in the same coordinate system to generate a submap in 3D space. The submap is also rasterized to generate a probability map, and then S and P are projected into the raster map respectively to score separately. Due to the difference in sensors, the scores are multiplied by their respective weight factors. The weight factor can be the covariance matrix of LiDAR and RGB-D camera data.

To further improve the matching accuracy, the Gauss-Newton method is used to optimize the results. Due to the high accuracy of 2D LiDAR, the only 2D point cloud is considered in the optimization. Therefore, the objective function is as follows:

$$E(T) = \arg \min \left(\sum_{i=1}^I (1 - M(S_i(T)))^2 \right) \quad (32)$$

where $S_i(T)$ is the coordinate of LiDAR point converted to the submap coordinate system, $M(S_i(T))$ is the occupation probability of LiDAR point coordinate on the map after conversion. For nonlinear function, first-order Taylor expansion and linearization, we get:

$$E(T + \Delta T) = \arg \min \left(\sum_{i=1}^I (1 - M(S_i(T)))^2 - \nabla M(S_i(T)) \frac{\partial S_i(T)}{\partial T} \Delta T \right)^2 \quad (33)$$

Find the minimum value, that is (39) differentiates ΔT and makes it equal to 0:

$$\sum \left[\nabla M(S_i(T)) \frac{\partial S_i(T)}{\partial T} \right] \times \left[1 - M(S_i(T))^2 - \nabla M(S_i(T)) \frac{\partial S_i(T)}{\partial T} \Delta T \right] = 0 \quad (34)$$

Expansion (40) gives:

$$\begin{aligned} & \sum \left[\nabla M(S_i(T)) \frac{\partial S_i(T)}{\partial T} \right]^T \left[1 - M(S_i(T))^2 \right] \\ &= \sum \left[\nabla M(S_i(T)) \frac{\partial S_i(T)}{\partial T} \right]^T \left[\nabla M(S_i(T)) \frac{\partial S_i(T)}{\partial T} \Delta T \right] \end{aligned} \quad (35)$$

Let H be expressed as:

$$H = \sum \left[\nabla M(S_i(T)) \frac{\partial S_i(T)}{\partial T} \right]^T \left[\nabla M(S_i(T)) \frac{\partial S_i(T)}{\partial T} \Delta T \right] \quad (36)$$

Then the solution is.

$$\Delta T = H^{-1} \sum \left[\nabla M(S_i(T)) \frac{\partial S_i(T)}{\partial T} \right]^T \left[1 - M(S_i(T))^2 \right] \quad (37)$$

where $\nabla M(S_i(T))$ represents the derivative of the map and is solved by bicubic interpolation.

In summary, the sequential registration algorithm that combines LiDAR and RGB-D depth cameras is mainly divided into three steps:

Firstly, the LiDAR data and depth camera data are obtained and converted to the global coordinate system. Among them, the data acquired by the RGB-D camera is 3D point cloud data. To reduce the amount of calculation, a 3D point cloud is filtered. The filtering method adopts bilateral voxel filtering, and the edge points are filtered to reduce noise.

Then, the inter-frame matching is performed by the CSM method. Probabilistic raster maps are constructed from the LiDAR point cloud and camera point cloud in the submap, and then the converted LiDAR point cloud and camera point cloud are projected into their respective raster maps for scoring, and the final results are both scored. It is the sum of the weighted scores of the two, and the scoring formula is shown in (31).

Finally, the point cloud data is inserted into the submap, reproject the LiDAR data, the least square problem is constructed, and the matching accuracy is further improved through optimization. After the matching is completed, the current LiDAR point and RGB-D camera point are inserted into the submap. When the number of frames reaches a certain size, it is saved as a keyframe for subsequent loop closure detection and the establishment of a new keyframe.

C. LiDAR LOOP CLOSURE DETECTION COMBINED WITH 3D POINT CLOUD DESCRIPTORS

Loop closure detection is the most important part of slam, which is directly related to the accuracy of the final output pose. In the long-term motion process of the robot, the cumulative error is increasing. When the robot returns to the previous position, new constraints can be added. The correct loop closure information can be used to eliminate odometer error, to get a globally consistent map. The wrong loop closure will not only interfere with the graph-based optimization but also result in totally wrong mapping results.

Loop closure detection is a matching process, that is, when a new scan is obtained, the optimal matching frame is searched within a certain range around it. If the optimal matching frame meets the requirements, it is regarded as a loop closure.

$$\xi^* = \frac{\arg \max}{\xi \in W} \sum_{k=1}^K M_{nearest}(T_\xi h_k) \quad (38)$$

where W is the search space, and M is to extend $M_{nearest}$ from the nearest raster point (that is, convert the raster point value to the corresponding pixel) by rounding its parameters to all two-dimensional planes R^2 . The simplest search method is to search one by one from the three dimensions of x , y and θ in the search window to find the best matching pose.

With the increase of the running trajectory, the number of sub-maps gradually increases, and the search space also becomes larger and larger. Searching directly in the search space will cause the performance of the algorithm to deteriorate seriously and cannot meet real-time performance. Therefore, the principle of the branch and bound [36] is used to improve the search space data structure, to efficiently find T_ξ .

Due to the addition of depth camera point cloud data, the amount of submap information becomes considerable, and the amount of data of a single submap also becomes huge. During the loop closure detection, the newly observed data is directly matched with the keyframe submap and solving it is very time-consuming and laborious. The above loop closure detection based on the branch and bound algorithm is solved on the 2D submap. If the data of the 3D point cloud is directly discarded and the submap formed by LiDAR points is used alone, the accuracy of loop closure cannot be guaranteed, and it is also a waste of data information. Therefore, 3D point clouds are stored in a submap in the form of descriptors. The descriptors not only greatly reduce the amount of original data, but also retain enough environmental information.

For the RGB-D camera point cloud extraction descriptor in submap, the first step is to filter out the ground. The ground point is the main proportion in the 3D point cloud. At the same time, in the SLAM problem based on ground leveling, the ground point needs to be segmented. The methods used to segment ground points include clustering segmentation algorithm and random consistency based sampling. Once the ground point is deleted, then the point cloud is divided into a set of point clusters C_i by the Euclidean clustering method. For each C_i , the centroid is calculated by the formula (39):

$$c_i = \frac{1}{k} \sum_{k=1}^n c_k \quad (39)$$

For a point cluster C_i , the descriptor is defined as a set of multiple feature vectors $f_i = [f_i^1, f_i^2, \dots, f_i^n]$. The more vectors, the more fully reflect the characteristics of the original point cloud. This paper selects two types of feature vectors with better performance: based on eigenvalues and shape function sets.

Characteristic vector-based on feature values describes the characteristics of point clouds as a vector of 1×7 , describing point clouds from multiple angles such as linearity, plane, scattering, total variance, anisotropy, feature entropy, and curvature changes. Make e_1, e_2, e_3 represent the spatial tensor structure eigenvalues of a 3D point cloud, and a series of characteristic values of the point cloud is extracted from the formula in Table 2.

TABLE 2. Feature vectors based on feature values.

Features	Expression
linearity	$L_\lambda = \frac{e_1 - e_2}{e_1}$
plane	$P_\lambda = \frac{e_2 - e_3}{e_1}$
scattering	$S_\lambda = \frac{e_3}{e_1}$
total variance	$Q_\lambda = \sqrt[3]{e_1 e_2 e_3}$
anisotropy	$A_\lambda = \frac{e_1 - e_3}{e_1}$
feature entropy	$E_\lambda = -\sum_{i=1}^3 e_i \ln(e_i)$
curvature changes	$C_\lambda = \frac{e_3}{e_1 + e_2 + e_3}$

The Ensemble of Shape Functions (ESF) uses three different shape functions to describe the distance, angle, and area distribution of some point clouds. The descriptor can effectively calculate the original point cloud data, can be extended to hundreds of classes, and can handle the differences in sensor characteristics, thereby achieving robust real-time classification.

As for the matching between 3D point cloud descriptors, this paper adopts the learning method. It is difficult for traditional methods to select the appropriate scale and threshold, especially in the case of multiple features. Therefore, the classifier is used to determine whether two point clouds represent the same object. To maintain efficiency, a K-D-Tree search is performed in the feature space to retrieve candidate matches, and then the candidate matches are sent to the classifier.

For the random forest classifier to determine whether C_i and C_j point cloud clusters represent the same object, calculate the absolute difference between the eigenvectors of the eigenvalues: $\Delta f^1 = |f_i^1 - f_j^1|$. The feature vectors f_i^1 and f_j^1 are input into a classifier based on a total feature dimension of 1×21 . For the ten histograms of shape histogram features,

the intersection points of the histograms are calculated to obtain the feature of dimension 1×10 . Given this set of features, the random forest classifier assigns a matching classification score ω . Apply a threshold on ω to build the final list of candidate matches passed to the next module.

In the loop closure detection process, real-time calculation of 3D point cloud matching has a great test on the performance of the algorithm, and the real-time performance of the algorithm cannot be guaranteed. Generally speaking, the recall rate of loop closure detection can be sacrificed slightly, and the accuracy rate must be guaranteed. Therefore, in the loop closure detection of the fusion of LiDAR and RGB-D camera point cloud, the strategy adopted is to first use the 2D LiDAR submap to find the appropriate loop closure, and after the loop closure is detected, the 3D point cloud descriptor is used for matching and verification the authenticity of the loop closure, to improve the accuracy.

The algorithm flow is shown in Fig. 12. After receiving a new frame of data, the LiDAR Scan data of the 2D LiDAR is searched in the submap through the CSM algorithm to detect the loop closure. When the loop closure is detected, the RGB-D camera in the original data is detected. The point cloud is matched with the descriptor in the corresponding submap. If the result matches, the relative pose between the current pose and submap is output, and new constraints are added to the pose map.

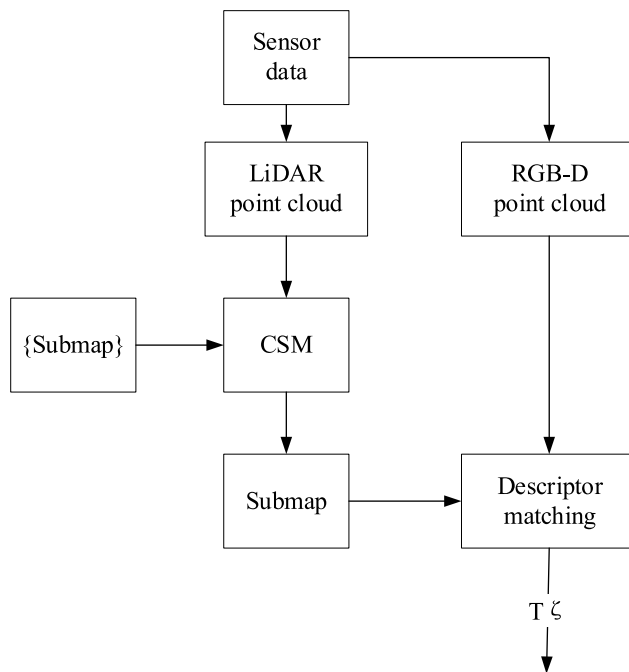


FIGURE 12. Loop closure detection of point cloud combining LiDAR and RGB-D camera.

Once loop closure detection is detected, optimization is needed to adjust the pose. The loop closure optimization

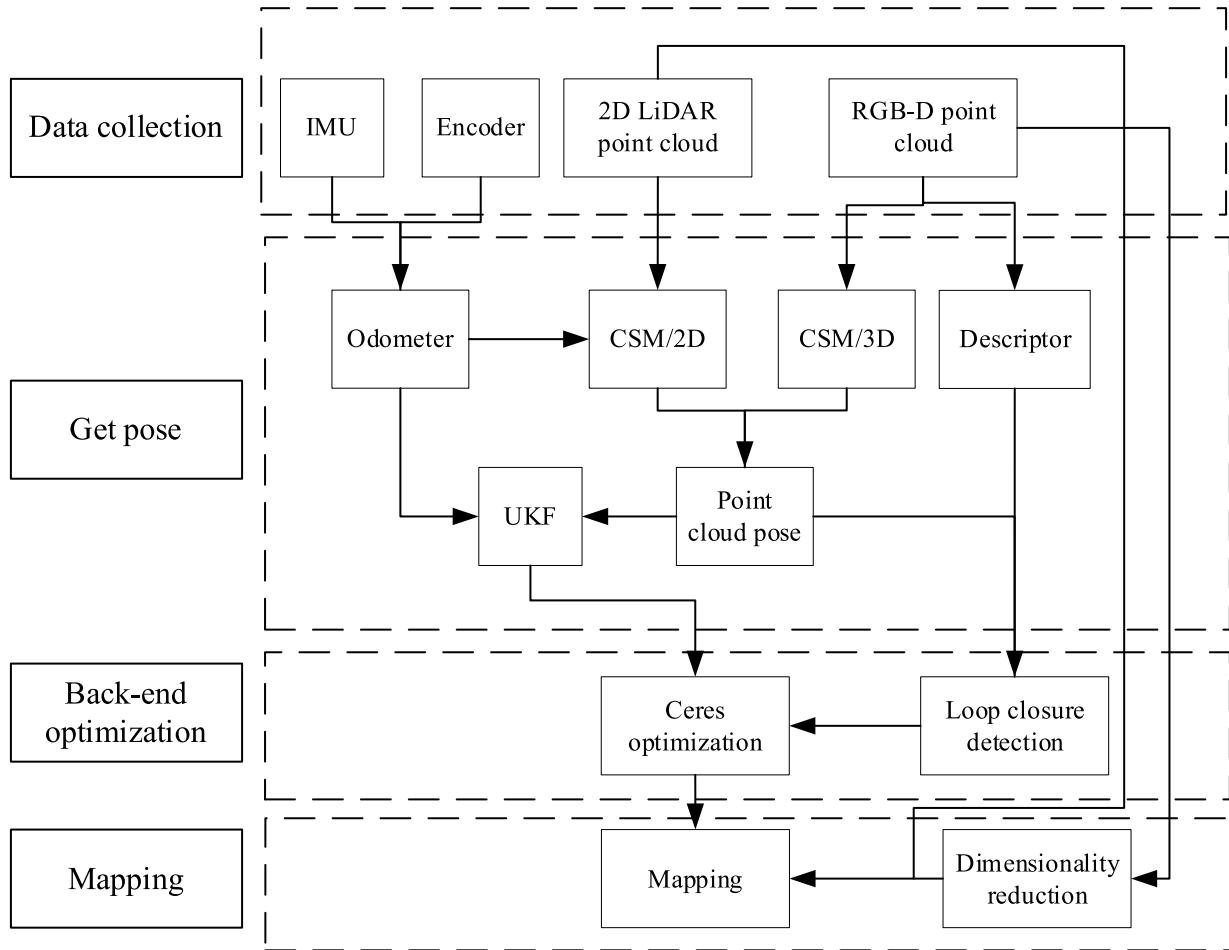


FIGURE 13. System module.

problem is described as a nonlinear least square problem

$$\arg \min \frac{1}{2} \sum_{ij} \rho(E^2(\xi_i^m, \xi_j^s, \sum_{ij}, \xi_{ij})) \quad (40)$$

where ξ_i^m is the pose of the submap and ξ_j^s is the pose of the LiDAR frame. The loop closure constraint between the LiDAR frame and submap detected during map construction is represented by their relative pose ξ_{ij} and their corresponding covariance matrix \sum_{ij} . ρ is a loss function. Its main function is to reduce the impact of error constraints during optimization. Huber loss function can be used. The residual error E is calculated by (41).

$$E^2(\xi_i^m, \xi_j^s, \sum_{ij}, \xi_{ij}) = e(\xi_i^m, \xi_j^s, \xi_{ij})^T \sum_{ij}^{-1} e(\xi_i^m, \xi_j^s, \xi_{ij}) \quad (41)$$

where:

$$e(\xi_i^m, \xi_j^s, \xi_{ij}) = \xi_{ij} - \begin{pmatrix} R_{\xi_i^m}^{-1}(t_{\xi_i^m} - t_{\xi_j^s}) \\ \xi_{i;\theta}^m - \xi_{i;\theta}^s \end{pmatrix} \quad (42)$$

D. EXPERIMENT AND ANALYSIS

This paper designs a multi-sensor graph-based optimization SLAM system, including four modules: data collection, pose acquisition, back-end optimization, and map creation. Combined with the previous theory, a multi-sensor graph-based optimization SLAM system is designed, including four modules, as shown in Fig. 13. The data acquisition module collects sensor data, processes it, and sends it to the pose acquisition module. The IMU and the encoder are fused to generate odometer information. The 2D LiDAR and RGB-D point cloud are scanned and matched to synthesize the pose, and then the UKF is fused to improve the positioning accuracy. The RGB-D point cloud is also used to solve the descriptor and simplify the point cloud information is optimized in the back-end optimization module through loop closure detection to restore the actual movement pose of the robot. The map creation module creates a raster map based on the 2D point cloud generated by the 2D LiDAR point cloud and the RGB-D point cloud dimensionality reduction.

Fig. 14 shows maps collected by the experimental robot in the laboratory. The laboratory is equipped with tables, chairs, file cabinets, and experimental benches, which are

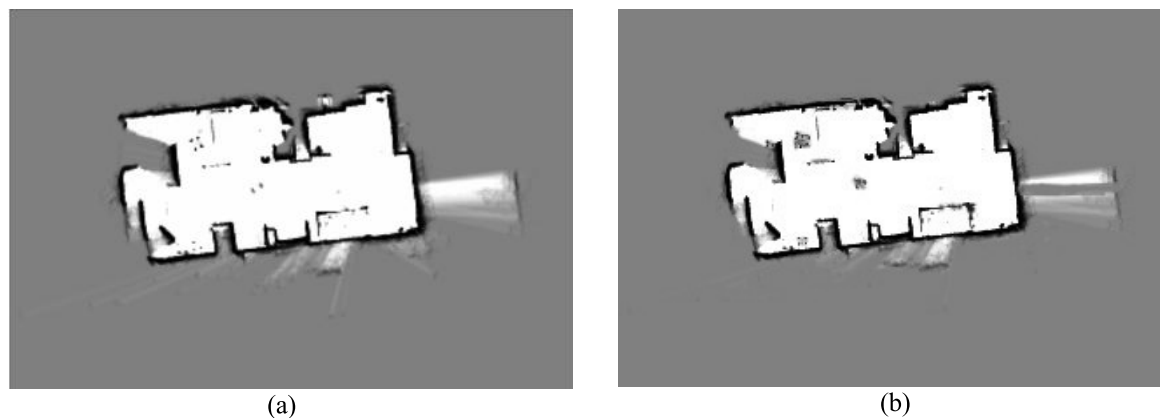


FIGURE 14. Lab map. (a) 2D LiDAR mapping. (b) 2D LiDAR + RGB-D point cloud mapping.

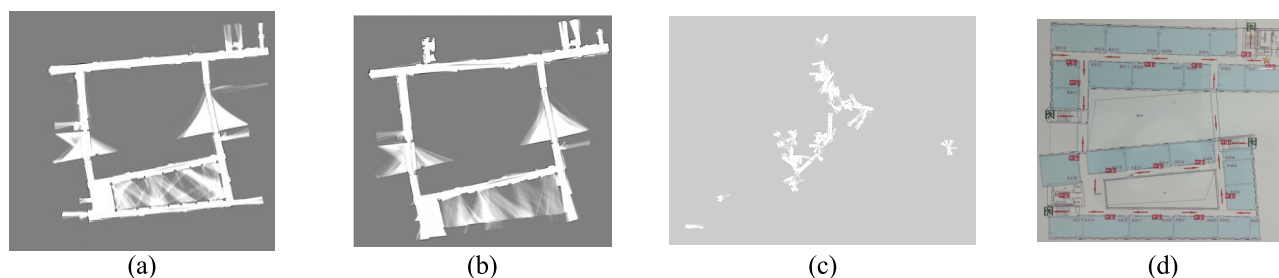


FIGURE 15. Experiment Mapping Comparison. (a) The algorithm in this paper. (b) Cartographer. (c) GMapping. (d) Lab map.

more common in real environments and can be used as representatives in small environments. In Fig. 14, white represents a feasible area, gray represents an unknown area, and black represents an obstacle. Fig. 14(a) is scanned with 2D LiDAR and Fig. 14(b) is a map that combines LiDAR and RGB-D point clouds. From the figure, it can be seen that objects such as walls and tables are in the left and right pictures all are reflected, but in (a), because the shape of the table and other furniture is irregular, for example, the four legs of the table support the table above, the representation in the figure is not clear, only some black obstacles can be seen, and in (b), because the RGB-D camera can scan 3D objects, the scanned map is closer to the actual environment, which has an important role in the subsequent application of maps.

A large-scale experiment was conducted in the experimental building. The site is a typical cloister structure with few environmental features, long linear distances, and similar structures at the corners. The experimental scene is 22.4m long from east to west and 19.7m from north to south. The layout diagram is shown in Fig. 15(d).

For the sake of comparison, first operate the mobile robot to move, first in a large loop closure, and then in a small loop closure, and repeat it many times to scan all the details of the map. The experiment also tested the GMapping algorithm.

As shown in Fig. 15, both the algorithm in this paper and the Cartographer algorithm can construct a complete environment map. The corners in the environment and the

ring structure can be well restored. As shown in Fig. 16, long-distance linear corridors can also be constructed completely. Due to the high pose accuracy, the straightness of the corridor restored by the algorithm in this paper is better. The results of the map construction show that the algorithm used in this paper is improved by 3.6% and 2.2% in two directions than the Cartographer algorithm.

The GMapping algorithm is a particle filtering algorithm in small scenes and cannot detect loop closure. At the same time, the algorithm performance requirements are high, because the robot runs faster when recording data. Therefore, the result of the map construction is poor, and the map is completely disorganized.

Since the robot cannot know the true motion posture during operation, it is impossible to find accurate positioning errors. [37] proposes an evaluation method based on posture relationship measurement. Instead of directly comparing the poses of the trajectory nodes and the corresponding ground real poses, this method compares the relative poses between the two trajectory nodes during the movement and the correspondence between the two trajectory nodes in the real trajectory. The real trajectory is generated from a trajectory with a loop closure, which satisfies certain conditions. The tool is provided in cartographer, which can be easily used for evaluation. Table 3 shows the evaluation results of the algorithm and the cartographer algorithm using this tool. From the table, it can be seen that the accuracy of this algorithm and

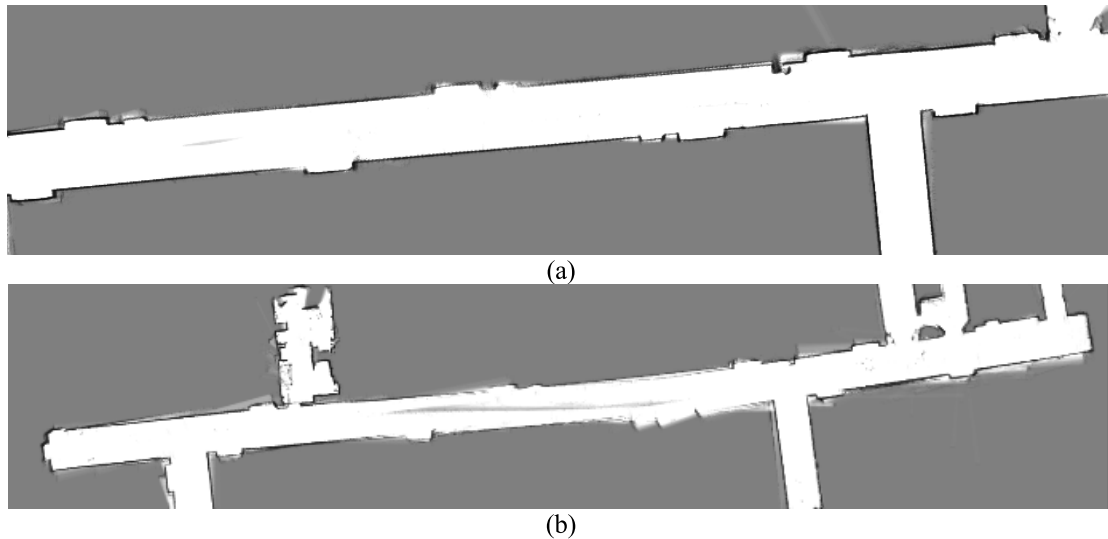


FIGURE 16. The long corridor. (a) The algorithm in this paper. (b) Cartographer.

TABLE 3. Comparison of positioning accuracy.

	Our proposed algorithm	Cartographer
Abs translational error	0.03546 m	0.04253 m
Sqr translational error	0.01032 m ²	0.00865 m ²
Abs rotational error	0.1867 deg	0.20393 deg
Sqr rotational error	0.07988 deg ²	0.08755 deg ²

cartographer are in the same order of magnitude, the accuracy is close, and the algorithm accuracy is high.

The experiment also considered the correctness of the 3D point cloud verification loop closure detection. During the operation of the algorithm, a total of 1365 loop closure were detected. After 3D point cloud verification, 8 loop closure were excluded. After manual judgment, there are 3 actual wrong loop closure. All happen at the big corner. Point cloud verification reduces the recall rate of loop closure detection to a certain extent but improves the accuracy rate, which ensures that the map will not show large deviations.

VII. CONCLUSION

In this paper, a new SLAM method based on graph-based optimization is proposed by fusing 2D LiDAR, RGB-D camera, encoder, and IMU. In-depth theoretical and experimental research was conducted and the following results were obtained.

- 1) In this paper, the UKF is used to carry out the joint positioning of four kinds of sensors. The experimental results show that the positioning error of the UKF algorithm is 124% and 120% lower than the EKF

algorithm in X and Y direction, and the actual measurement error of the UKF algorithm is less than 1%, which has higher accuracy and more accurate path.

- 2) A fusion strategy of 2D LiDAR point cloud and RGB-D camera point cloud is proposed. Experiments show that the constructed map can reflect the actual structure and details of the environment, which is 3.6% and 2.2% higher than the Cartographer algorithm in two directions.
- 3) The evaluation method based on the posture relationship measurement was evaluated by the tool of the Cartographer. The results show that the accuracy of the algorithm in this paper is the same order of magnitude as that of the Cartographer, and the accuracy is close, but the accuracy of the algorithm is higher.
- 4) The correctness of loop closure detection in 3D point cloud verification is investigated. The results show that the algorithm in this paper reduces the recall rate of loop closure detection to a certain extent, but improves the accuracy rate and ensures that the map does not appear large deviations.

In general, the proposed algorithm has a certain application value, the construction of the map can reflect the actual structure and details of the environment, in terms of operational efficiency, the algorithm because of the addition of 3D point cloud, compared to the single 2D LiDAR, the calculation force requirements are higher, but due to the use of description sub-description 3D point cloud, the algorithm in real-time can also meet the actual operating requirements.

REFERENCES

- [1] M. Yang, "Overview on issues and solutions of SLAM for mobile robot," *Comput. Syst. Appl.*, vol. 27, no. 7, pp. 1–10, Jul. 2018.
- [2] H. Durrant-Whyte, "Consistent integration and propagation of disparate sensor observations," in *Proc. IEEE Int. Conf. Robot. Autom.*, Jul. 1986, pp. 3–24.

- [3] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," in *Proc. IEEE Int. Conf. Robot. Autom.*, Dec. 1987, pp. 435–461.
- [4] S. Holmes, G. Klein, and D. W. Murray, "A square root unscented Kalman filter for visual monoSLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008, pp. 3710–3716.
- [5] A. Doucet, "Rao-blackwellised particle filtering for dynamic Bayesian networks," in *Proc. Conf. Uncer. Artif. Intel. (UAI)*, 2000, pp. 176–183.
- [6] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte Carlo localization for mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, Dec. 1999, pp. 1322–1328.
- [7] M. Montemerlo and S. Thrun, "Simultaneous localization and mapping with unknown data association using FastSLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2003, pp. 1985–1991.
- [8] M. Montemerlo, "Fast SLAM: A factored solution to the simultaneous localization and mapping problem," in *Proc. AAAI Nat. Conf. Artif. Intell. (AAAI)*, 2002, pp. 593–598.
- [9] S. Kazuya, "FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that provably converges," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2003, pp. 1151–1156.
- [10] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Auto. Robots*, vol. 4, no. 4, pp. 333–349, 1997.
- [11] F. Dellaert and M. Kaess, "Square root SAM: Simultaneous localization and mapping via square root information smoothing," *Int. J. Robot. Res.*, vol. 25, no. 12, pp. 1181–1203, Dec. 2006.
- [12] K. Konolige, G. Grisetti, R. Kuámmerle, W. Burgard, B. Limketkai, and R. Vincent, "Efficient sparse pose adjustment for 2D mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 22–29.
- [13] M. Kaess, A. Ranganathan, and F. Dellaert, "ISAM: Incremental smoothing and mapping," *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1365–1378, Dec. 2008.
- [14] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "ISAM2: Incremental smoothing and mapping using the Bayes tree," *Int. J. Robot. Res.*, vol. 31, no. 2, pp. 216–235, Feb. 2012.
- [15] T. Duckett, S. Marsland, and J. Shapiro, "Learning globally consistent maps by relaxation," in *Proc. IEEE Int. Conf. Robot. Automat. Symp.*, Dec. 2000, pp. 3841–3846.
- [16] J. B. Gao and C. J. Harris, "Some remarks on Kalman filters for the multisensor fusion," *Inf. Fusion*, vol. 3, no. 3, pp. 191–201, Sep. 2002.
- [17] M. R. Asharif, B. Moshiri, and R. HoseinNezhad, "Sensor fusion by pseudo information measure: A mobile robot application," *ISA Trans.*, vol. 41, no. 3, pp. 283–301, Jul. 2002.
- [18] G. G. Rigatos, "Extended Kalman and particle filtering for sensor fusion in motion control of mobile robots," *Math. Comput. Simul.*, vol. 81, no. 3, pp. 590–607, Nov. 2010.
- [19] F. Demim, A. Nemra, H. Abdelkadri, A. Bazoula, K. Louadj, and M. Hamerlain, "SLAM problem for autonomous underwater vehicle using SVSF filter," in *Proc. 25th Int. Conf. Syst., Signals Image Process. (IWS-SIP)*, Jun. 2018, pp. 1–5.
- [20] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, Jun. 2007.
- [21] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. 6th IEEE ACM Int. Symp. Mixed Augmented Reality*, Nov. 2007, pp. 225–234.
- [22] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.
- [23] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.
- [24] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2320–2327.
- [25] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer*, 2014, pp. 834–849.
- [26] J. Engel, J. Stuckler, and D. Cremers, "Large-scale direct SLAM with stereo cameras," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 1935–1942.
- [27] D. Caruso, J. Engel, and D. Cremers, "Large-scale direct SLAM for omnidirectional cameras," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 141–148.
- [28] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 15–22.
- [29] C. Kerl, J. Sturm, and D. Cremers, "Dense visual SLAM for RGB-D cameras," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 2100–2106.
- [30] X. Wang, L. He, and T. Zhao, "Mobile robot for SLAM research based on Lidar and binocular vision fusion," *Chin. J. Sens. Actuators*, vol. 31, no. 03, pp. 394–399, Mar. 2018.
- [31] L. Yin, "Research on simultaneous localization and mapping fusion with low-cost laser sensors and vision," *J. Integr. Technol.*, vol. 8, no. 2, pp. 11–22, Mar. 2019.
- [32] X. Ding, Y. Wang, D. Li, L. Tang, H. Yin, and R. Xiong, "Laser map aided visual inertial localization in changing environment," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 4794–4801.
- [33] Z. Jin, Y. Shao, M. So, C. Sable, N. Shlayan, and D. M. Luchtenburg, "A multisensor data fusion approach for simultaneous localization and Mapping*," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 1317–1322.
- [34] S. Kumar and R. M. Hegde, "Multi-sensor data fusion methods for indoor localization under collinear ambiguity," *Pervas. Mobile Comput.*, vol. 30, pp. 18–31, Aug. 2016.
- [35] R. Siegwart, *Introduction to Autonomous Mobile Robots*, 2nd ed., vol. 3, Cambridge, MA, USA: MIT Press, 2004, pp. 270–275.
- [36] Z. Liu, "Research and implementation of SLAM algorithm base on sweeping robot," M.S. thesis, Dept. Soft. Eng., Eng., Xidian Univ., Shanxi, China, 2019.
- [37] R. Kämmerle, B. Steder, C. Dornhege, M. Ruhnke, G. Grisetti, C. Stachniss, and A. Kleiner, "On measuring the accuracy of SLAM algorithms," *Auto. Robots*, vol. 27, no. 4, pp. 387–407, Nov. 2009.



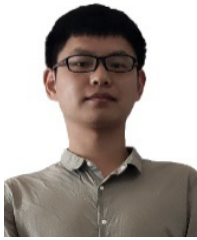
LILI MU is currently a Master Tutor with the Department of Measurement and Control, School of Mechanical Engineering, Anhui University of Science and Technology. Her main research interests include the mobile robot and intelligent measure and control.



PANTAO YAO is currently pursuing the M.S. degree with the School of Mechanical Engineering, Anhui University of Science and Technology. His main research interests include visual simultaneous localization and mapping (VSLAM).



YUCHEN ZHENG is currently pursuing the bachelor's degree with the Faculty of Mathematics and Statistics, The University of Sydney. Her main research interests include artificial intelligence algorithm and mathematical modeling.

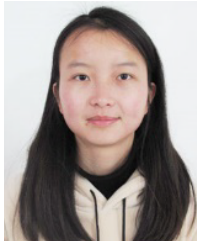


KAI CHEN is currently pursuing the M.S. degree with the School of Mechanical Engineering, Anhui University of Science and Technology. His main research interests include laser simultaneous localization and mapping.



NANA QI is currently pursuing the M.S. degree with the School of Mechanical Engineering, Anhui University of Science and Technology. Her main research interests include machine vision and microfluidics.

...



FANGFANG WANG is currently pursuing the M.S. degree with the School of Mechanical Engineering, Anhui University of Science and Technology. Her main research interests include machine vision and vibration.