

## Research Article

# An Improved AGV Real-Time Location Model Based on Joint Compatibility Branch and Bound

Yang Yang,<sup>1</sup> Juntao Li ,<sup>2</sup> and Xiaoling Li<sup>2</sup>

<sup>1</sup>School of Management, China University of Mining & Technology (Beijing), Beijing, China

<sup>2</sup>School of Information, Beijing Wuzi University, Beijing, China

Correspondence should be addressed to Juntao Li; [ljtletter@126.com](mailto:ljtletter@126.com)

Received 31 July 2020; Revised 5 October 2020; Accepted 26 October 2020; Published 23 November 2020

Academic Editor: Jianbing Ma

Copyright © 2020 Yang Yang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Automated Guided Vehicle (AGV) indoor autonomous cargo handling and commodity transportation are inseparable from AGV autonomous navigation, and positioning and navigation in an unknown environment are the keys of AGV technology. In this paper, the extended Kalman filter algorithm is used to match the sensor observations with the existing features in the map to determine the accurate positioning of the AGV. This paper proposes an improved joint compatibility branch and bound (JCBB) method to divide the data and then randomly extract part of the data in the divided data set, thereby reducing the data association space; then, the JCBB algorithm is used to perform data association and finally merge the associated data. This method can solve the problem of the increased computational complexity of JCBB when the amount of data to be matched is large to achieve the effect of increasing the correlation speed and not reducing the accuracy rate, thereby ensuring the real-time positioning of the AGV.

## 1. Introduction

The traditional robot positioning method generally uses the internal sensors, gyroscopes, odometers, and so forth of the robot to estimate the positioning information for the robot's next moment or uses markers in the real environment to estimate the robot's pose through the AGV's internal sensors. The accuracy of the posture of the AGV obtained by the calculation is very low. When relying on markers to locate the robot, additional facilities are needed for assistance. However, in an unknown environment, the robot has no external objects to use as a reference. The robot uses the internal sensors and matches the environmental characteristic information collected by the lidar to obtain accurate positioning.

Simultaneous localisation and mapping (SLAM) was first proposed by Hugh Durrant Whyte and John J. Leonard; it was the first time the method of map construction and then localisation was proposed. In [1], it was first proposed to integrate the extended Kalman filter (EKF) algorithm into the SLAM problem, which laid the foundation for the

research into SLAM-related methods in recent years. Lowe et al. proposed SLAM based on visual information, which is the pioneering work in the area of SLAM research. This approach put forward new ideas for the research and development of SLAM in the following decades [2]. Davison proposed SLAM based on monocular vision on the basis of the visual SLAM, the earliest SLAM research with real-time performance up to 2003 [3]. Wang et al. proposed a fusion of a laser and a vision indoor SLAM method, based on the robot's own odometer to establish a robot motion model, obtain environmental information in the experimental environment through lidar, and calculate the physical coordinates and pixel coordinates [4]. Davison et al. proposed a SLAM based on a monocular vision to obtain the real-time position and motion trajectory of the camera. Finally, the real-time positioning of the camera was realised [5]. Gao proposed a robot positioning and grid map construction based on the track estimation algorithm in the absence of environmental information and the specific information of known road signs in the environment based on the robot's own driving model [6]. Luo and Chen proposed a method of

laser and monocular vision fusion, which solved the problem of robot positioning and map construction when the environment is large or the obstacles in the environment are not obvious, and designed a visual loop detection to ensure AGV positioning with the accuracy of the map [7]. Li et al. proposed an improved visual simultaneous positioning and map creation (vSLAM) algorithm for high positioning accuracy of the AGV in the industrial field [8]. At the front end of the algorithm, the image collected by the camera is matched by a binocular matching algorithm. At the back of the algorithm, the information is detected by a monocular camera shooting vertically to optimise the global position and reduce the error of the SLAM system. This configuration can meet industrial site requirements.

Data association is a very important part of SLAM research. It is the process of matching the existing environmental characteristics inside the robot with the observed environmental characteristics to determine whether they have a common source. In this way, the noise points collected by the lidar are filtered out to ensure the correct matching of the same environmental features observed at different times. The correct data association will make the robot positioning accurate while ensuring the correctness of the environment map established by the robot, but if the data matching is wrong, it will not only affect the robot's real-time positioning and map construction but also cause the divergence of the SLAM algorithm, so the data association is very important in the SLAM research. Data association is the matching between the data in the environmental characteristics and the observations. The data association is used to complete the simultaneous positioning and map construction of the robot. The data association algorithm not only compares the observations with the existing environmental characteristics but also needs to be able to remove the noise points. The noise points are removed, so the data association algorithm selects the observation values that fall within the threshold to match the existing environmental characteristics by setting the association threshold.

In this paper, the extended Kalman filter algorithm is used to match the sensor observations of AGV with the existing features in the map to realise the accurate positioning of the AGV. The process of data matching [9, 10] is the process of data association [11, 12]. This paper proposes an improved joint fusion branch and bound (JCBB) method to divide the data, randomly extract part of the data in the divided data set, reduce the data association space, perform data association based on the JCBB algorithm, and then combine the associated data and union. This method can solve the problem of the increased computational complexity of JCBB when the amount of data to be matched is large to achieve the effect of increasing the correlation speed and not reducing the accuracy rate, thereby ensuring the real-time positioning of the AGV. The positioning accuracy of AGV directly affects the accuracy of the map it builds independently. The data matching algorithm proposed in this paper can improve the positioning accuracy of the AGV, which is of great significance for the AGV to independently construct a global map.

## 2. AGV System Model Based on DashgoD1

**2.1. Dashgo Platform.** DashgoD1 was developed by Shenzhen Enjoy AI Co., Ltd., which was established in 2015, and is globally oriented. DashgoD1 includes lidar, positioning, a navigation module, and the robot mobile platform D1. The application of the robot's photomagnetic wireless technology greatly extends the life of the robot while ensuring the high reliability and high precision performance of the robot. Dashgo, a mobile platform developed for ROS, has high precision, a heavy load, long battery life, and strong scalability. Its characteristics are high precision, ease of use, and an ROS development kit, which can independently develop robot projects, as shown in Figure 1.

**2.2. AGV Coordinate Model and Motion Model.** The positioning of AGV in the warehouse environment generally uses the Cartesian coordinate system and a polar coordinate system. Polar coordinates are generally converted into two-dimensional coordinates through the conversion of polar coordinates to two-dimensional coordinates. The Cartesian coordinate system needs to know the abscissa of the AGV in the environment. The ordinate and the polar coordinate require the angle of the AGV in the environment and the distance under that angle. The conversion between the two needs to establish the relationship between the AGV coordinate system and the global coordinate system. The laser rangefinder used in this article is installed in the centre of the AGV. The coordinate system of the lidar can be the coordinates of the AGV by default. The coordinates of the AGV in the global map are the coordinates of the lidar in the global map. Therefore, the global coordinates of AGV in the global map are the global coordinates of the lidar  $X_w O_w Y_w$ . For central coordinates,  $(x_r, y_r)$  indicates that the laser rangefinder is in the global coordinate system  $X_w O_w Y_w$ . Also used in Chinese,  $(x_r, y_r)$  represents that a feature point is used in global coordinates  $(x_{wi}, y_{wi})$  in the AGV coordinate system  $(x_{ri}, y_{ri})$  expresses.  $[x_r(k), y_r(k), \theta_r(k)]$  represents the global coordinate system of AGV at time  $KX_w O_w Y_w$ . The state vector of the  $\theta_r(k)$  representation of AGV in global coordinates  $x_w$  is shown in Figure 2: it is the schematic diagram of the AGV coordinate system.

The establishment of the AGV motion model is based on the encoder installed on each wheel to count the output pulse number of each wheel. Then, according to the diameter of the two wheels, we can calculate the respective running distance of the two wheels after a period of movement. The motion model establishes the relationship between the AGV distance and the angle from the previous time to the next moment, as shown in Figure 3 if the AGV state vector is  $[x_r(k), y_r(k), \theta_r(k)]$ . Then, the state vector at  $K+1$  is  $[x_r(k+1), y_r(k+1), \theta_r(k+1)]^T$ . The encoder, which is used to measure the moving distance of AGV during driving, is located on the left and right wheels. The diameter of the driving wheel is  $D$ . The number of output pulses of the encoder per wheel is  $n$  linear rotation. From  $K$  time to  $(K+1)$  time, the output pulse of the left and right encoder is  $m_L, m_R$ ,

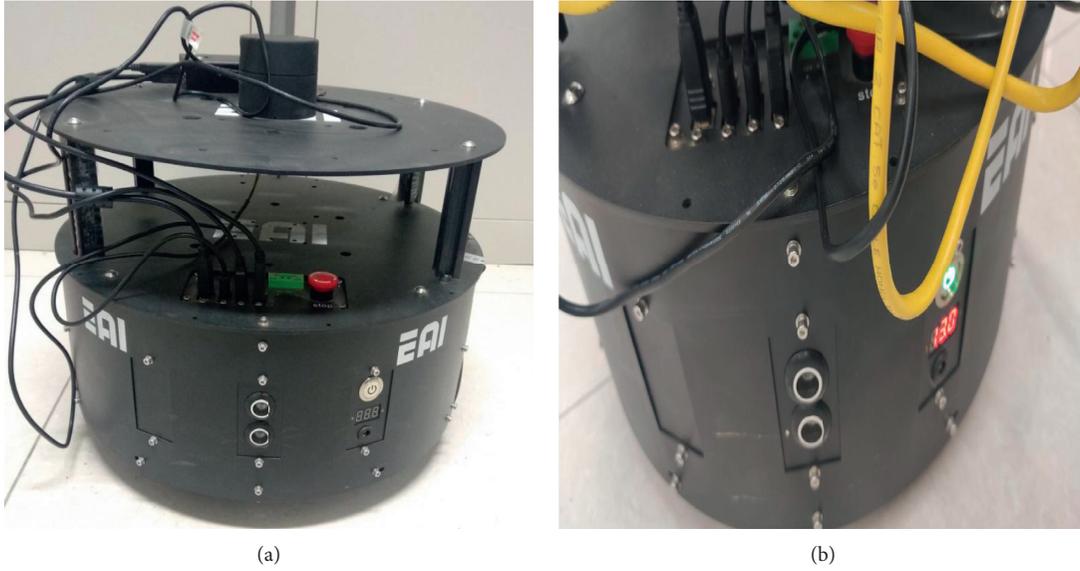


FIGURE 1: Dashgo D1 mobile handling robot.

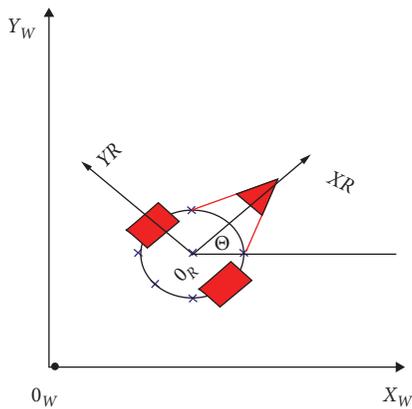


FIGURE 2: Schematic diagram of the AGV coordinate system.

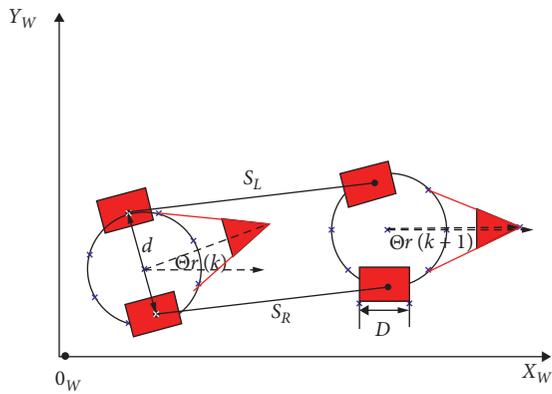


FIGURE 3: Intention of AGV motion model.

then  $m_L$  the  $m_R$ , which can be calculated from time  $k$  to time  $(K + 1)$ , with  $S_L$  and  $S_R$  distances:

$$S_L(K \rightarrow K + 1) = \pi D \cdot \frac{M_L}{n}, \tag{1}$$

$$s_R(k \rightarrow k + 1) = \pi D \cdot \frac{m_R}{n}.$$

The AGV angle variation from  $K$  time to  $(K + 1)$  time is as follows:

$$\Delta\theta(k \rightarrow k + 1) = \frac{s_R(K \rightarrow K + 1) - s_L(K \rightarrow K + 1)}{d}, \tag{2}$$

where  $D$  is the distance between two wheels, that is, the diameter of AGV:

$$\Delta S(k \rightarrow k + 1) = \frac{S_L + S_R}{2}. \tag{3}$$

From equation (2), we determine that

$$\theta(k + 1) = \theta_k + \Delta\theta(k \rightarrow K + 1). \tag{4}$$

To summarize, at  $(K + 1)$  time, the AGV is at the global coordinates  $X_w O_w Y_w$ . In the middle of  $X_w$ , the horizontal and vertical coordinates of the direction are as follows:

$$y_r(k + 1) = y_r(k) + \Delta S(k \rightarrow k + 1) \cdot \sin\left(\frac{\theta(k) + \theta(k + 1)}{2}\right),$$

$$x_r(k + 1) = x_r(k) + \Delta S(k \rightarrow k + 1) \cdot \cos\left(\frac{\theta(k) + \theta(k + 1)}{2}\right). \tag{5}$$

Hypothesis  $w(k)$  is the Gaussian white noise with zero mean value; that is, the system equation of AGV at time  $k$  is

$$X_{(k+1)} = \begin{bmatrix} x_r(k+1) \\ y_r(k+1) \\ \theta_r(k+1) \end{bmatrix} = \begin{bmatrix} x_r(k) + \Delta S(k \rightarrow k+1) \cdot \cos\left(\frac{\theta(k) + \theta(k+1)}{2}\right) \\ y_r(k) + \Delta S(k \rightarrow k+1) \cdot \sin\left(\frac{\theta(k) + \theta(k+1)}{2}\right) \\ \theta_r(k) + \Delta\theta_r(k \rightarrow k+1) \end{bmatrix} + w_k. \quad (6)$$

### 3. AGV Location Model Based on EKF-SLAM

#### 3.1. Mathematical Model

**3.1.1. State Vector.** AGV needs to use the road sign information in the existing map to locate, and the accurate positioning of the landmark information cannot do the location without the accurate positioning performance of AGV, because only when the AGV positioning is accurate can the surrounding environment be accurately scanned, and the accurate environment map can be obtained. Therefore, AGV positioning and map construction are two mutual problems, and they affect each other. The accuracy of one affects the accuracy of the other. Therefore, they cannot be studied as two separate problems, and they cannot be estimated separately as two separate problems. The extended Kalman filter algorithm is adopted, which takes the AGV position and all the landmarks in the environment into an extended state vector and carries them into the algorithm for iterative estimation. The covariance matrix is used to reflect the degree of uncertainty between each landmark and between AGV and each landmark  $x_r(k) = [x_r(k), y_r(k), \theta_r(k)]$ . With the  $i^{\text{th}}$  characteristic,  $P_i(k) = [x_i(k), y_i(k)]$ . The vector composed of all the feature points in the map is  $P(k) = [P_1(K), P_2(K), \dots, P_m(K)]^T$ . The pose of AGV and all the feature points in the map constitute an extended state vector  $X(K) = \begin{bmatrix} x_r(k) \\ P(k) \end{bmatrix}$ . The covariance matrix corresponding to the state vector is as follows:

$$P(k) = 1m \begin{bmatrix} P_{vv}(k) & P_{v1}(k) & \cdots & P_{vm}(k) \\ P_{v1}^T(k) & P_{11}(k) & \cdots & P_{1m}(k) \\ \vdots & \vdots & \ddots & \vdots \\ P_{vm}^T(k) & P_{1m}^T(k) & \cdots & P_{mm}(k) \end{bmatrix}. \quad (7)$$

**3.1.2. State Transition Model.** At  $k$  time, the state of the system is  $X(K)$  control input of the system  $U(k)$  ( $U(k)$  is the angle of rotation of the AGV  $\theta$ ). At  $K+1$ , the state of the

system is  $X(k+1)$ . The state vector transition model of the system is as follows:

$$X(K+1) = F(X(k), u(k)) = F(x_r(k), P(k), u(k)). \quad (8)$$

AGV is the equation of motion:

$$\begin{cases} x_r(k+1) = x_r(k) + \Delta S(k \rightarrow k+1) \cdot \cos\left(\frac{\theta(k) + \theta(k+1)}{2}\right), \\ y_r(k+1) = y_r(k) + \Delta S(k \rightarrow k+1) \cdot \sin\left(\frac{\theta(k) + \theta(k+1)}{2}\right), \\ \theta_r(k+1) = \theta_r(k) + \Delta\theta_r(k \rightarrow k+1). \end{cases} \quad (9)$$

In this formula, the coordinates of agv  $x$  and  $y$  at time  $K+1$  are expressed as the coordinates of time  $k$  plus the variation from time  $k$  to time  $K+1$ . This variation is the average value of the moving distance of AGV in this time period and the angle of AGV at two times. The angle of  $K+1$  is the angle change of  $K$  time and this time period.

**3.1.3. Observation Model.** The observation model can connect the global coordinate system with the AGV coordinate system. It is assumed that the  $i^{\text{th}}$  environmental feature point is observed by AGV at  $k$ -time  $P_i$  for the establishment of the observation model and the position and attitude of AGV  $x_r(k)$  as well as  $P_i$ . The observation model is as follows:

$$Z(k) = h(x(k)) + w(k) = h(x_r(k), P_i(k)) + w(k). \quad (10)$$

According to  $P_i$ , the state variables of the characteristic point at time  $k$  and AGV at time  $k$ , as well as the angle value of AGV at this time, are expanded from equation (10) to obtain the following equation:

$$\begin{cases} Z_x(k) = (x_{pi}(k) - x_r(k))\cos\theta_r(k) + (y_{pi}(k) - y_r(k))\sin\theta_r(k) + w_x(k), \\ Z_y(k) = (x_{pi}(k) - x_r(k))\sin\theta_r(k) + (y_{pi}(k) - y_r(k))\cos\theta_r(k) + w_y(k). \end{cases} \quad (11)$$

Among the  $w(k)$  values is the white Gaussian noise when observing the feature points, and the variance is  $R(k)$ .

### 3.2. Algorithm Flow

**3.2.1. Initialisation.** In the initial state, the AGV has no action, so the state vector has only three parameters to describe the position and attitude of the AGV, that is, the two-dimensional coordinates and the angle value of the AGV at the initial time. The initial dimension of the extended state vector is 3D. To make the extended state vector have parameters and carry out subsequent motion and scanning, the AGV coordinate system is taken as the global coordinate system of the system. If there is no initialisation operation at the initial time of AGV, the subsequent dead reckoning will become quite complex, and the accuracy of AGV positioning will be affected, which will lead to the low accuracy of the environment map constructed at  $x_0 = [0 \ 0 \ 0]^T$ . The covariance matrix corresponding to the position and pose of AGV and the state vector composed of all feature points in the map is as follows:

$$P_0 = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix}. \quad (12)$$

**3.2.2. Forecast.** When the AGV is walking, whenever its angle or walking distance exceeds a certain range, the AGV will automatically adjust. It will be adjusted according to the AGV internal sensor data and the data collected by the lidar, based on the extended Kalman Filter estimation, which are to use AGV internal sensor data and lidar data to estimate the pose at the next moment.

According to the state transition equation, the estimated value of the system at the last moment and the control input of the system are taken as the input parameters  $u(k)$  (the control input is the robot rotation angle  $\theta$ ). According to the state transition model equation (8), the system state estimation value and its corresponding covariance matrix can be calculated:

$$x(k+1|k) = F(x(k|k), \bar{u}(k)). \quad (13)$$

Formula (8) is linearly expanded by the Taylor formula (Taylor formula is a formula that uses the information of a function to describe the value near it):

$$x(k+1|k) \approx F(x(k|k), u(k)) + \left[ \frac{\partial F}{\partial x} \Big|_{x=x(k|k)} \frac{\partial F}{\partial u} \Big|_{u=u(k)} \right] \begin{bmatrix} x(k) - x(k|k) \\ u(k) - \bar{u}(k) \end{bmatrix}. \quad (14)$$

According to the definition of covariance

$$\begin{aligned} P(k+1|k) &= \text{cov}(x(k+1|k), x(k+1|k)), \\ &= E\{(x(k+1) - x(k+1|k)) \cdot (x(k+1) - x(k+1|k))^T\}. \end{aligned} \quad (15)$$

Because of the error,  $\Delta x = x(k+1) - x(k+1|k)$ ,  $\Delta u = u(k) - \bar{u}(k)$ . The state estimation is independent of control input noise  $\Delta u(\Delta x)^T = 0$ . By combining formulas (13) and (16), we obtain the following results:

$$\begin{aligned} p(k+1|k) &= \left( \frac{\partial F}{\partial x} \Big|_{x=x(k|k)} \right) P(k|k) \left( \frac{\partial F}{\partial x} \Big|_{x=x(k|k)} \right)^T \\ &\quad + \left( \frac{\partial F}{\partial u} \Big|_{u=\bar{u}(k|k)} \right) Q(k) \left( \frac{\partial F}{\partial u} \Big|_{u=\bar{u}(k|k)} \right)^T. \end{aligned} \quad (16)$$

**3.2.3. State Estimation.** Suppose that, at the time of  $K+1$ , the observed value of AGV is  $Z(k+1)$ . If the influence of noise is ignored, then the observed value at  $K+1$  time is

$$\hat{Z}(k+1) = h(x(k+1|k)). \quad (17)$$

Innovation is the difference between observation and prediction  $V(k+1)$ . The specific steps of information calculation are as follows: (1) first, the observation model  $h$  is used to calculate the predicted observation through the current state estimation; (2) the laser rangefinder scans the real environment to obtain the characteristic information of the environment, and the characteristic information is used for the observation information  $z(k)$ ; and (3) according to the innovation calculation formula, the new interest is obtained as follows:

$$v(k+1) = Z(k+1) - \hat{Z}(k+1). \quad (18)$$

The error covariance of innovation is used with  $s(k+1)$  to express  $R(k)$ :

$$s(k+1) = HP(k+1|k)H^T + R(k). \quad (19)$$

To observe the noise covariance,  $h$  is the Jacobian matrix of the partial derivative of  $H$  to  $X$   $H_{i,j} = (\partial h_i / \partial w_i)$ .

**3.2.4. Data Association.** Data association matches the landmark features existing in the environment with the measured values of the environment observed by AGV, so that the AGV can obtain accurate positioning, thus building an accurate environment map. Suppose there are  $n$  environmental features in the robot environment map at a certain time  $F_n = \{F_1, F_2, F_3, \dots, F_n\}$ . The data obtained by the sensor are  $E_m = \{E_1, E_2, E_3, \dots, E_m\}$ . Data association is to establish the hypothesis of the two  $H_m = \{j_1, j_2, j_3, \dots, j_m\}$ .

### 3.2.5. Pose Estimation

(1). *Calculation of Kalman Gain.* Kalman gain can be easily understood as a controller in feedback control, where the parameters of this controller can change from time to time. When we can control a system, the Kalman filter is stable, and the corresponding value is in the Kalman filter equation. The variance matrix  $P$  tends to a constant, so the Kalman gain tends to a constant. The calculation formula is as follows:

$$K_k = P(k|k-1)H^T S_k^{-1}. \quad (20)$$

(2). *Pose Estimation.* According to the above steps, the innovation, the innovation covariance, and the state prediction value are calculated to update the pose state, assuming that there are  $n$  line segment features in the global map. The specific pose state formula is as follows:

$$X(k+1) = x(k+1|k) + \frac{1}{n} \sum_{i=1}^n K_{(k+1)}(i) v_{(k+1)}(i). \quad (21)$$

3.2.6. *Feature Update.* If there are  $n$  environmental features in the AGV environment map at  $K+1$  time, when the  $N$  features in the environment are matched with lidar data, the existing environment feature data contains the data that cannot be matched with  $x_i$ . At this time, the data can be added to the environment feature data as a new environment feature:

$$\begin{aligned} F_{(n,k+1)} &= \{F_1, F_2, F_3, \dots, F_n\} \longrightarrow, \\ F_{(n,k+1)^+} &= \{F_1, F_2, F_3, \dots, F_n, E_i\}. \end{aligned} \quad (22)$$

According to the above steps, the accurate positioning of AGV is basically completed. Through the extended Kalman filter algorithm, AGV will always calibrate its own posture in the process of walking to ensure the accuracy of AGV's position and posture at each time. During the walking process, AGV will scan the feature points in the environment through the lidar and match these environmental feature points with the data association algorithm. If the new features have not been added to the environment, the new features are added to the environment.

## 4. Data Association in EKF-SLAM

Data association, a very important part of SLAM research, is a process of matching the existing environmental characteristics and the observed environmental characteristics of robots to determine whether they have a common source, to filter out the noise points collected by lidar, and to ensure the correct matching of the same environmental features observed at different times. The correct data association will make the robot positioning accurate and ensure the correctness of the environment map established by the robot. However, if the data matching is wrong, it will not only affect the robot's real-time positioning and map construction but also lead to the divergence of the SLAM algorithm. Therefore, data association is very important in SLAM

research. Data association is the matching between the data in the environment features and the observation values. Using the data association to complete the robot positioning and map construction at the same time, the data association algorithm not only needs to match the observed values with the existing environmental features but also needs to be able to remove the noise points. Therefore, the data association algorithm selects the observation values falling into the threshold and matches the existing environmental features by setting the correlation threshold.

4.1. *JCBB Data Association Method.* To limit the occurrence of uncertain matching, it is necessary to reconsider the matching constraints. The joint compatibility branch and bound (JCBB) data association algorithm improves the shortcomings of ICNN. JCBB uses a joint compatibility test to test the compatibility between all observations and map features. Moreover, considering the correlation between all features and robots and between features, the probability of joint compatibility between a wrong pair and other pairs decreases with the increase in the number of pairs. JCBB uses the branch and bound algorithm to search the solution space and locate the association with the largest number of pairs. The single nondecreasing rule of the pairing number will discard the data points that will not match correctly. There are three possibilities between the survey data and the existing features in the map: ① the same source as a landmark feature in the previously constructed map; ② the data point being a new landmark feature; or ③ being an abnormal value, which is not a real physical feature. In addition, it is necessary to update the observed values according to the observed value  $z_{k,i}$ , which represents the characteristic observation at  $k$  time.

The principle of joint compatibility is to set an association gate in advance and take the observation point that falls within the association gate and is closest to the predicted position of the target as the target association object. The "association gate" here refers to the preset association range, and the so-called "nearest neighbour" means the smallest Mahalanobis distance. The dotted line represents the correlation threshold, the ellipse represents the variance, the triangle represents the observation prediction, the dot represents the observation feature, and the five-pointed star represents the real environment feature. As Figure 4 shows, there are two circles of observations of real environmental features, which indicate that another feature point with a large variance may be noise. The triangle represents the observed prediction. However, within the same variance range, there is a certain error. The robot's SLAM process is a process of constant prediction and correction. As the robot continues to observe, the number of features in the environment continues to increase. If each new feature is associated with the feature in the environment, the computational complexity will be very large. At this time, the association threshold plays a role. Filtering out a large amount of data that will not fall within this threshold greatly reduces the amount of calculation and ensures the real-time performance of robot positioning and map construction.

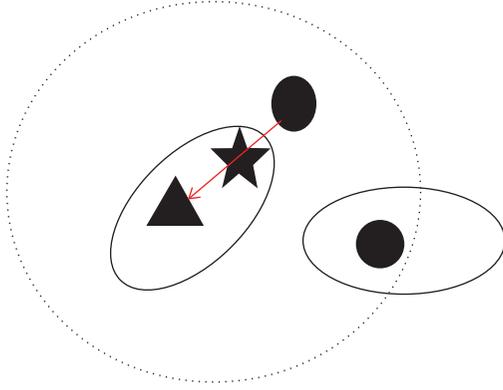


FIGURE 4: Schematic diagram of joint compatibility.

Suppose there are  $n$  environmental features in the robot environment map at a certain time  $F_n = \{F_1, F_2, F_3, \dots, F_n\}$ . The data obtained by the sensor are  $E_m = \{E_1, E_2, E_3, \dots, E_m\}$ . Data association establishes the hypothesis of the two  $H_m = \{j_1, j_2, j_3, \dots, j_m\}$  features of the environment  $F_j$ . The observation value of the robot can be obtained by a nonlinear measurement function including the position of the robot and the feature point  $h_{k,j}$ . Both are included in the state vector  $x_{(k|k-1)}$ . It observed  $z_{(k,i)}$  from  $F_j$ ; that is, there is  $z_{k,i} = h_{k,i}(x_{k|k-1}) + w_{k,i}$ .

$$h_{k,j}(x_{k|k-1}) = [h_{k,1}(x_{k|k-1}), h_{k,2}(x_{k|k-1}), \dots, h_{k,m}(x_{k|k-1})]. \quad (23)$$

The difference between the predicted values of the observed value  $I$  and the characteristic value  $J$  is expressed by deviation innovation information:

$$v_{k,ij} = Z_{k,i} - h_{k,j}(\hat{x}_{k|k-1}). \quad (24)$$

The corresponding covariance is as follows:

$$s_{k,ij} = \nabla H_{k,j} P_k \nabla H_{k,j}^T + R_{k,i}. \quad (25)$$

$\nabla H_{k,j} = (\partial h_{k,j} / \partial x_{k|k-1})|_{\hat{x}_{k|k-1}}$ ,  $R_{k,i}$  for observation noise  $v_{H_m}$ .

If the Mahalanobis distance satisfies

$$D_{k,ij}^2 = v_{k,ij}^T S_{k,ij}^{-1} v_{k,ij} < \chi_{d,1-\alpha}^2, \quad (26)$$

$D$  is observation  $Z_{k,i}$  dimension,  $1 - \alpha$ . In general, if the confidence value of the observed value is equal to that of the observed value (95.0), then the solution is considered to be consistent with the observed value  $H_m$ . The compatibility condition is satisfied.

**4.2. Improved JCBB Algorithm.** As the core of the JCBB algorithm, joint compatibility considers the compatibility between all measurements and map features. However, the time complexity of the JCBB algorithm increases exponentially with the number of observations, which seriously affects its real-time performance in the SLAM process. First, the range of data association is divided into a certain range.

Second, the strategy based on the angle threshold is used to collect and classify the data set of the divided area. It is assumed that there are  $n$  environmental features at  $k$  time  $F_k^n = \{F_1, F_2, F_3, \dots, F_n\}$  and  $M$  observational features:  $E_k^m = \{E_1, E_2, E_3, \dots, E_m\}$ .

**4.2.1. Local Data Region Selection.** The main idea of local area data selection is to be within a certain range, which is represented by  $R$ .  $R$  is the effective scanning range of the lidar, and  $D$  is the compensation distance. Since the lidar continuously collects data in the process of robot navigation and positioning, there are more and more environmental characteristic data in the system. If the radius of the observation area is too long to match the observation area, the new algorithm is used to match all the features in the environment. To reduce the amount of calculation of system data matching and ensure the real-time performance of SLAM, the specific division method is shown in equation (27)  $(x_r, y_r)$ , the location of the environment feature points in the global coordinate system  $(x_i, y_i)$  (Figures 5 and 6). The area divided meets the following conditions:

$$\begin{cases} |x_r - x_i| \leq r + d, \\ |y_r - y_i| \leq r + d, \\ \sqrt{(x_r - x_i)^2 + (y_r - y_i)^2} \leq r + d. \end{cases} \quad (27)$$

**4.2.2. Batch Processing of Data Sets.** In this paper, the lidar scanning range is  $360^\circ$ , the angular resolution is  $0.36^\circ$ , and the environment feature points obtained by the robot scanning one circle are  $1000360^\circ$ . The set angle is  $-180^\circ \pm 180^\circ$ ; that is, the environmental feature points obtained by lidar are arranged in the order of angle, and the difference between each two environmental feature points is  $0.36^\circ$ . Therefore, the proposed data set is divided by angles in batches. First, a threshold is set as  $\Delta\theta$ . First, the angle resolution of the lidar can be 3 times that of the lidar, which can be adjusted according to the experimental results in the experimental verification process. Second, in the local area set in the local data area selection, any existing environmental feature and any observation feature can be selected to calculate the distance between them. The angle between the two feature points and the robot's positive direction is calculated, respectively, when the angle between the two points is less than or equal to  $\Delta\theta$ . When the distance between them is less than or equal to the sum of the robot scanning radius and the compensation distance, the environmental characteristics and observation features in equation (28) are matched  $(x_k, y_k)$ , including the angle with the  $x$ -axis of the robot coordinate system in the positive direction  $\theta_k$ . Other observation feature points  $(x_i, y_i)$  included the angle with the  $x$ -axis of the robot coordinate system in a positive direction  $\theta_k$  ( $\theta_k, \theta_i \in (0^\circ, 180^\circ)$ ). If the following conditions are met, the feature points can be observed  $(x_k, y_k)$  and other observed feature points  $(x_i, y_i)$  will be grouped into a data set:

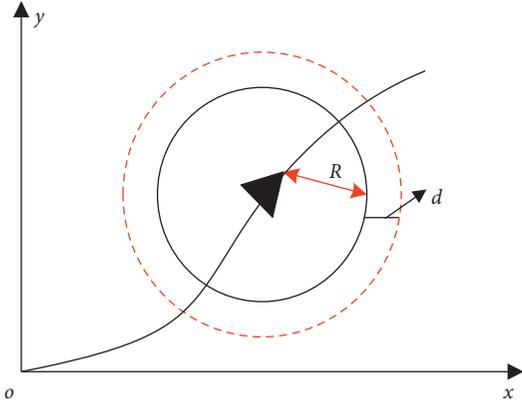


FIGURE 5: Schematic diagram of local area selection.

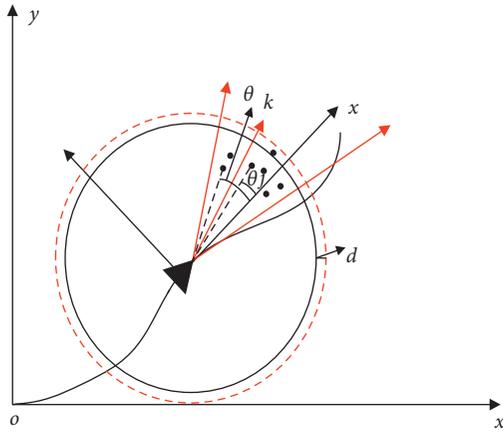


FIGURE 6: Schematic diagram of data set division.

$$\begin{cases} |\theta_k - \theta_i| \leq \Delta\theta, \\ \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2} \leq r + d. \end{cases} \quad (28)$$

In the experimental verification stage, because the F4 angle resolution of the flash lidar series is  $0.36^\circ$ , it is temporarily set at  $\Delta\theta = 3 * 0.36$ . Using the above method, the environment features are divided into  $m$  small data sets:  $E_k^1 = \{E_i | i = 1, 2, \dots, m_1\}$ ,  $E_k^j = \{E_i | i = 1, 2, \dots, m_j\}$ ,  $\dots$ ,  $E_k^M = \{E_i | i = 1, 2, \dots, m_M\}$ ,  $m = m_1 + m_j + \dots + m_M$ . Using JCBB to get each divided data set  $E_k^j$  and the environmental map features  $F_k^n$  for the correlation solution  $H_{best}^j$ , each of the solutions obtained will be composed of  $H_{best}$ .

In the process of laser data processing, the robot uses the internal sensor to obtain the attitude of the robot. At the same time, the robot scans the environment features to obtain the real-time environment characteristic data. The obtained characteristic data are the data under the robot coordinate, which is transformed into the global coordinate. For the feature points after a coordinate transformation, the local data area selection and data set division batch processing are carried out. At this time,  $M$  data sets are obtained. Each data set is used to match the existing environment

feature data in the map to ensure the accuracy of data matching and ensure real-time data matching. The flow chart is shown in Figure 7.

## 5. Simulation Experiment

**5.1. AGV Positioning Error Analysis.** In this paper, the SLAM simulator developed by Jai Juneja of Oxford University is used to build a simulation environment as shown in Figure 8, so that the robot can move at a constant speed in a square area of  $120 \times 120 \text{ m}^2$ , with obstacles and environmental road signs in the simulation environment. As shown in Figure 8, the black “-” indicates obstacles, the green “+” indicates the features of the environmental road signs, and the red “-” indicates the theoretical path of the robot. In Figure 8, the preset robot has an error of 0.1 m on the  $x$  and  $y$  coordinates, the scanning range of the laser rangefinder is 5.6 m, the observation noise is  $[0.02 \text{ m}; 1^\circ]$ , and the angle error is  $1^\circ$ . In Figure 9, the red “\*” is the obstacle scanned by the robot, and the red “-” is the theoretical path of the robot. Ten tests were performed to determine the error comparison chart, as shown in Figure 10.

As shown in Figure 10, the robot positioning pose based on the improved JCBB algorithm varies within 0.01 m–0.04 m, and the angle floats within  $0.04^\circ$ . The posture of the robot before the improvement is within the range of 0.01 m–0.15 m, and the angle is floating within  $0.12^\circ$ . Compared with the change value of the robot’s pose and angle before the improvement, the floating maximum value has been improved by 275% in the change range and the angle by 200%. The change trend of the positioning error of the improved and extended Kalman filter algorithm is shown in Figure 11. Figure 11 shows that the error is gradually larger at the beginning of the robot walking process, but as the robot continues to walk, the robot positioning error gradually decreases and gradually is stabilised. The experimental results prove that the improved algorithm in this paper is theoretically effective and has certain improvements in the robot pose error and angle error. Compared with the robot’s pose and angle error values before the improvement, the maximum value of the float is improved by 275%, and the angle is improved by 200%.

**5.2. Data Processing of AGV Estimated Position.** In the simulation experiment, first let the robot walk in a preset experimental environment without adding any improved algorithm. At this time, the robot’s own pose is calculated by dead reckoning through the internal sensor of the encoder. Figure 12 shows the real walking path and environment feature data of the robot dead reckoning through internal sensors. The robot has a large deviation during the walking process. As the robot walks, the cumulative error becomes larger and larger. Figure 13 is the extraction of the robot’s real walking trajectory and environmental feature data and the resulting dead reckoning positioning data processing diagram. The robot’s walking trajectory is constantly shifting with the robot’s walking, and the endpoint coordinates after walking a circle are (0.0813, 0.2375), which does not coincide with the initial

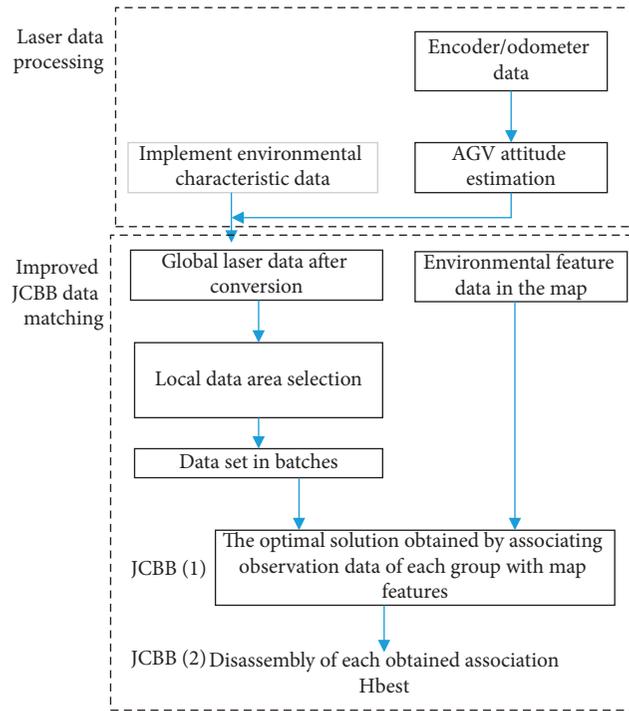


FIGURE 7: Flow chart of the improved JCBB algorithm.

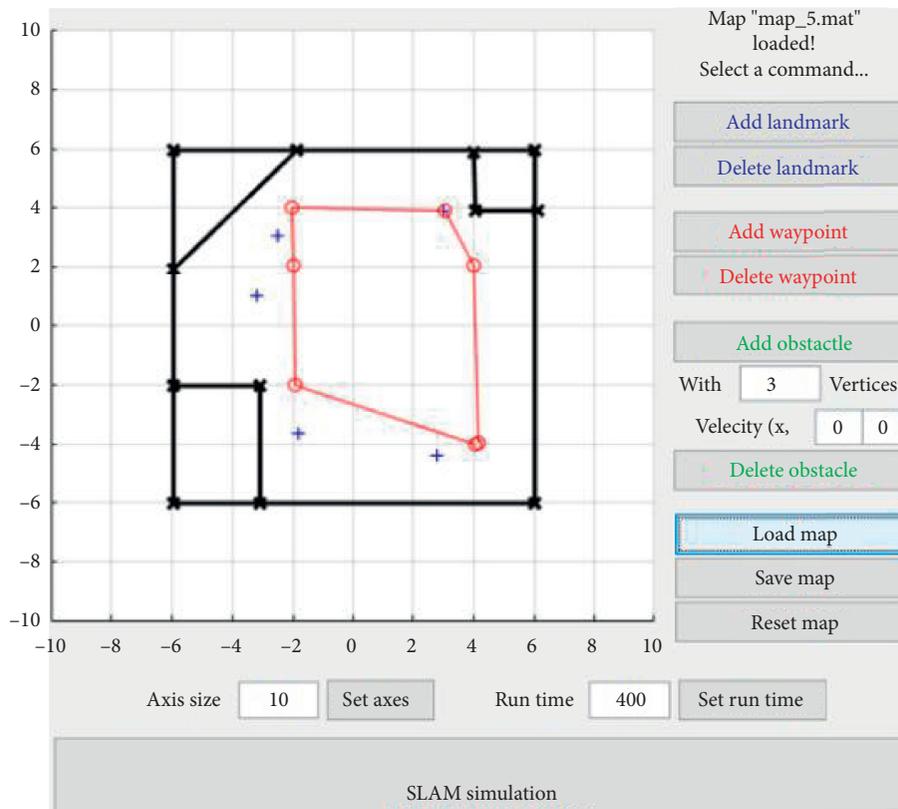


FIGURE 8: Simulation environment layout.

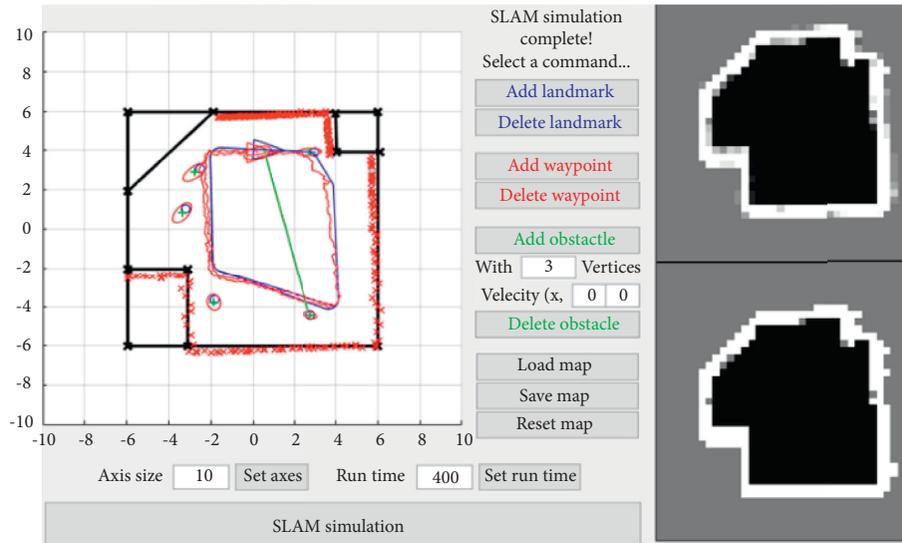


FIGURE 9: Simulation environment diagram.

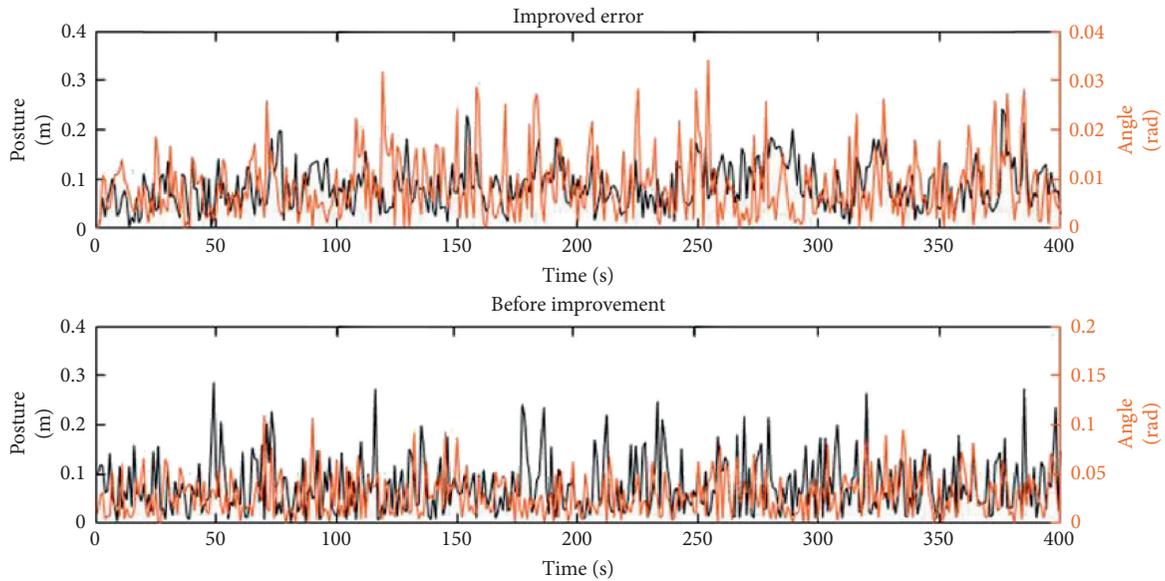


FIGURE 10: Comparison of the algorithm before and after improvement.

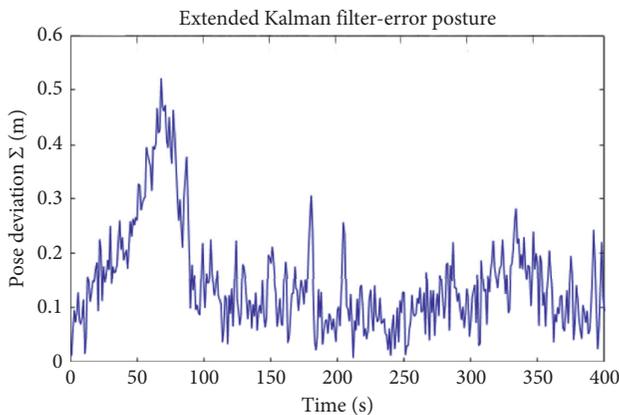


FIGURE 11: EKF-SLAM positioning error diagram.

point (0, 0) coordinates, and the robot has a major deviation during walking.

In this paper, the robot position and pose obtained by the encoder and odometer at each time and environment characteristic data point obtained by F4 lidar are used to convert local coordinate data to global coordinate data, as shown in Figure 14,  $t_k$ . The two-dimensional image of the environmental characteristics collected by the lidar at this time is the two-dimensional map under the robot coordinate system. Through the coordinate transformation, the two-dimensional environment feature map under the robot coordinate system is converted to the global coordinate system. The blue area in Figure 14 is the two-dimensional map of the red area under the global coordinate  $t_k$ . The figure shows that the two-dimensional image formed by the lidar

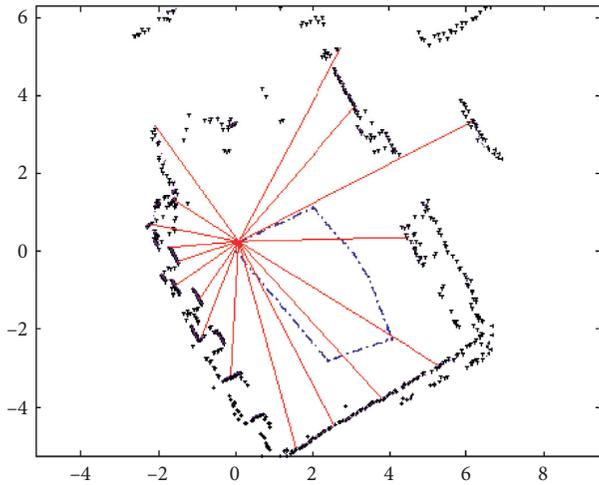


FIGURE 12: Result of dead reckoning and positioning of the robot.

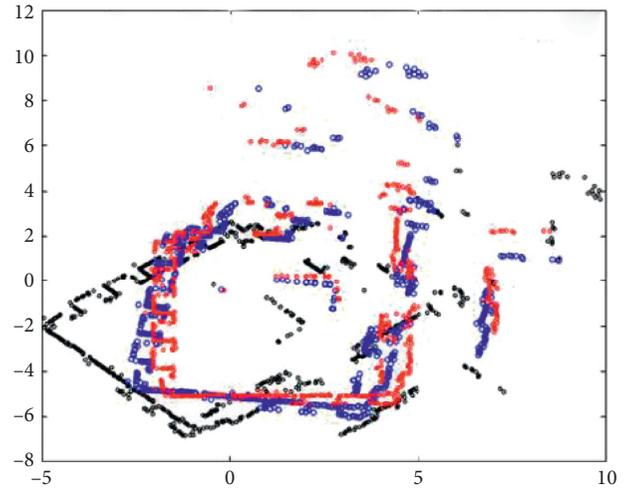


FIGURE 14: Schematic diagram of coordinate conversion of laser ranging data.

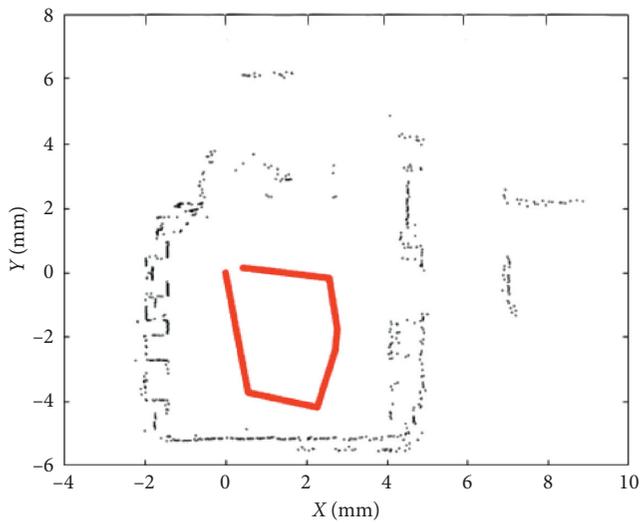


FIGURE 13: Data processing of dead reckoning and positioning of the robot.

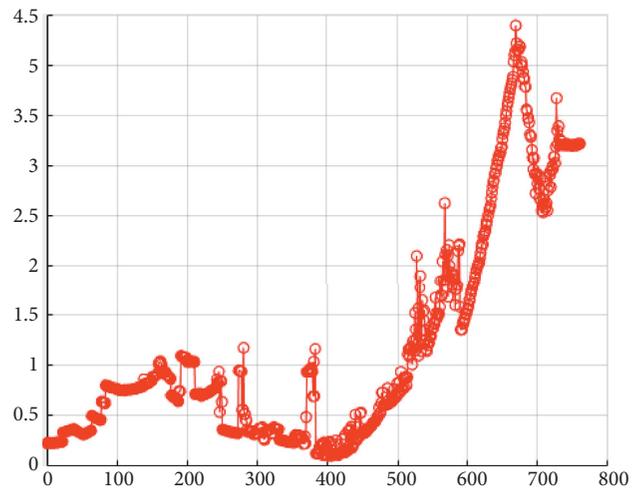


FIGURE 15: Schematic diagram of encoder error.

data converted by encoder positioning information does not coincide with the global system. In the same coordinate system, the shortest distance is used as the standard to measure the encoder error. The error is measured by calculating the distance between each point in the converted 2D-map and the corresponding point in the original global map. In principle, the smaller the distance between the corresponding points, the smaller the positioning error. Figure 15 shows that the encoder positioning error increases with the robot's moving error, on average, from the original 0.228798 cm to the maximum error of 4.3932 cm.

*5.3. AGV Autonomous Positioning Simulation Experiment.* Through EKF localisation of the improved joint compatible branch and bound algorithm (JCBB), the optimal pose estimation of the robot is finally obtained by fusing the sensor data of the robot. As shown in Figure 16, the path map of the robot walking in the real environment is shown, and the red

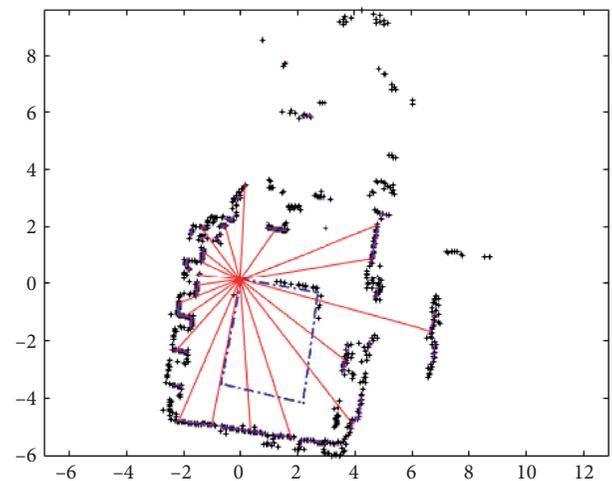


FIGURE 16: Improved EKF positioning result.

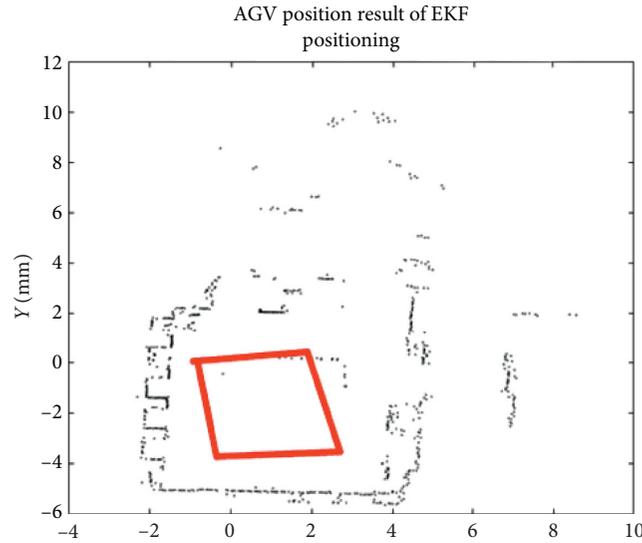


FIGURE 17: Improved EKF positioning data processing diagram.

line emitted is the laser beam scanning the environment by the lidar at that time. From the figure, we can see that, compared with Figure 12, the path of the robot is more regular, and the initial point almost coincides with the endpoint. When the robot positioning is accurate, the lidar carried by the robot can accurately scan the indoor environment, instead of obtaining numerous noise points. Figure 17 is the two-dimensional coordinate map of the data extracted by the MATLAB code in Figure 18. The figure clearly shows that the robot walking path is more accurate than that in Figure 12.

To improve the accuracy of localisation, an improved joint fusion branch and bound algorithm is proposed in the process of data matching. This algorithm can improve the accuracy of robot data matching and ensure the real-time positioning of the robot. Figures 16 and 17 show that the path of the robot is much more regular than that in Figure 12. However, if we want to measure the accuracy of robot positioning, we can see that the path of the robot is much more regular than the path of the robot in Figure 12. The robot localisation accuracy is measured by using the minimum position sensor of the laser radar each time. As shown in Figure 18, the black 2D coordinate is  $t_k$ . The two-dimensional image of environmental characteristics is the two-dimensional map under the robot coordinate system, and the red area map is the global coordinate  $t_k$ . The position of the robot is estimated by the global coordinate transformation of the environment  $t_k$ . Time F4 lidar scan data is placed into the global coordinates, as shown in the green two-dimensional figure.

Similar to the encoder error analysis, the shortest distance method is also used, in principle. The error is measured by calculating the distance between the laser

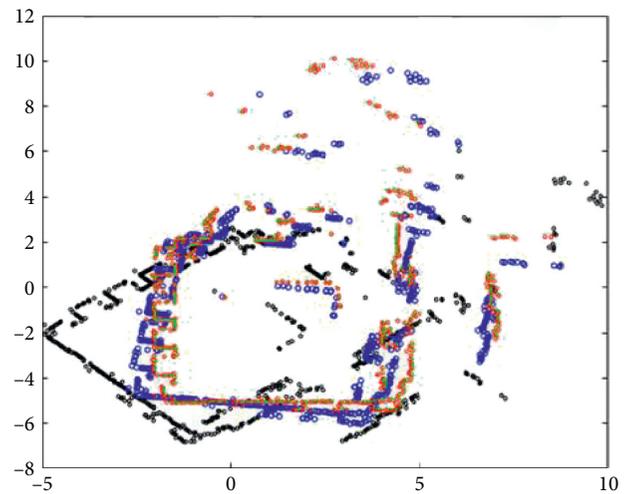


FIGURE 18: Diagram of coordinate conversion of laser ranging data.

scanning data converted by the improved EKF optimal estimation and the corresponding points in the current global map. This error is used to represent the error of the improved EKF positioning algorithm.

Through calculation, we can see that the improved EKF positioning error is between 0.2 cm and 0.45 cm in Figure 19, and the encoder error is between 0.2 cm and 4.5 cm in Figure 20. Moreover, the error change in the improved EKF positioning method is stable and small. However, when positioning only by the internal sensor of the robot, the error changes greatly, and the position changes greatly. The robot positioning error tends to

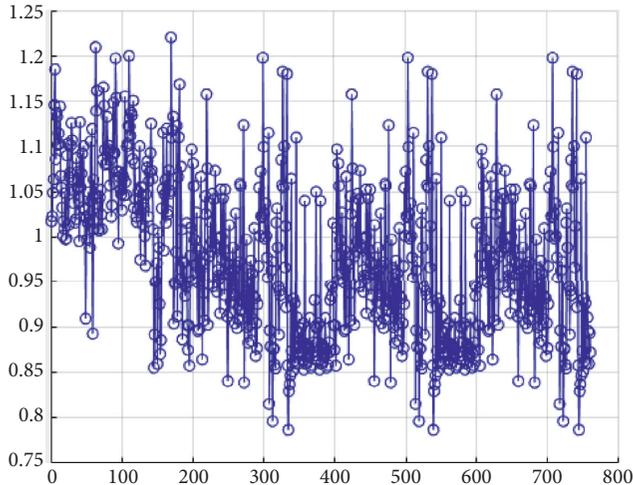


FIGURE 19: Improved EKF positioning error.

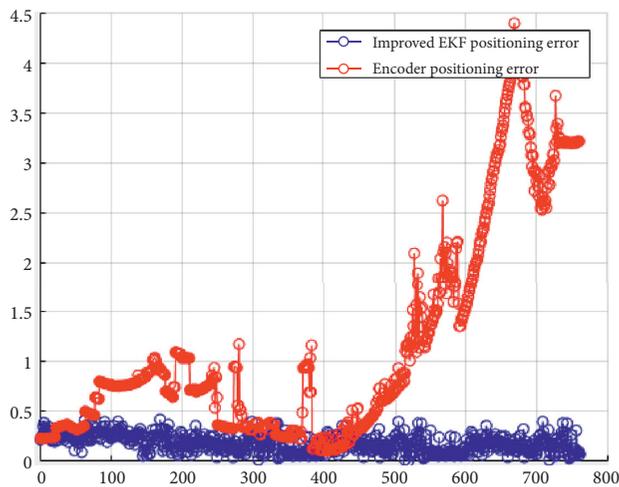


FIGURE 20: Error comparison before and after improvement.

decrease slightly, and there is no cumulative error such as that generated by the internal sensors of the robot. Therefore, it is proved that the EKF localisation algorithm based on the improved JCBB has a good effect on the precise positioning of the robot.

## 6. Summary

In this paper, the extended Kalman filter algorithm was used to match the sensor observations with the existing features in the map, to realise the accurate positioning of the AGV, including the initialisation of the AGV pose and the prediction of the AGV state, and the data association algorithm is used to match the existing features and the observed values in the environment. In this paper, the model of the AGV state vector and the state transition vector is established. The state vector model is an extended vector composed of the pose vector of the robot at a certain time and the state vector of the environmental feature points at the same time point.

The state transition vector is based on the position and posture of the robot at a certain time, including the value of  $X$  and  $Y$  coordinate points, and the angle between the robot and the positive half-axis of the global coordinate system. The observation model connects the global coordinate system with the AGV coordinate system. According to the position and attitude of the AGV at a certain time and the position and attitude of an environmental feature observed at this moment, an improved joint fusion branch and bound algorithm is proposed to match the feature points in the AGV environment with the observation features to achieve the accurate positioning of the AGV.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This research was supported by the Yueqi Youth Scholar Funding of China University of Mining and Technology (Beijing) (no. 800015Z1163). This research was also supported by the Major Program of National Natural Science Foundation of China (no. 71831001).

## References

- [1] R. Smith, M. Self, and P. Cheeseman, "A stochastic map for uncertain spatial relationships," in *Proceedings of the International Symposium on Robotics Research*, pp. 467–474, Philadelphia, PA, USA, 1987.
- [2] D. Lowe, S. Se, and J. Little, "Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks," *The International Journal of Robotics Research*, vol. 21, no. 8, pp. 735–758, 2002.
- [3] A. J. Davison, "Real-time simultaneous localization and mapping with a single camera," in *Proceedings Ninth IEEE International Conference on Computer Vision*, pp. 1403–1410, Nice, France, October 2003.
- [4] W. Wang, S. Sheng, K. Wang et al., "Indoor simultaneous localization and map construction for mobile robots based on laser rangefinder and monocular vision," *Proceedings of the Acta automatica Sinica*, vol. 31, no. 6, pp. 925–933, 2005.
- [5] A. J. Davison, I. D. Reid, N. Molton, and O. Stasse, "Mono SLAM: real-time single camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.
- [6] G. Hao, *Positioning and Mapping of Mobile Robots*, Inner Mongolia University of Science and Technology, Baotou, China, 2014.
- [7] Y. Luo and L. Chen, "Laser positioning and mapping based on visual information," *Industrial Control Computer*, vol. 30, no. 12, pp. 21–23, 2017.
- [8] Y. Li, S. Zhu, and Y. Yu, "Improved visual SLAM Algorithm in factory environment," *Robot*, vol. 41, no. 1, pp. 97–105, 2019.
- [9] J. Xiao, X. Duan, and X. Qi, "A geomagnetic matching ICCP algorithm based on probabilistic data association," *Chinese*

- Journal of Inertial Technology*, vol. 26, no. 2, pp. 202–208, 2018.
- [10] K. Zhao, *Research on Registration Algorithm Based on RGB-D Point Cloud Data in 3D Reconstruction*, Tianjin University, Tianjin, China, 2016.
- [11] Y. Ma, *Research on the Branch and Bound Algorithm of Mixed Integer Nonlinear Programming Problems*, Ningxia University, Yinchuan, China, 2014.
- [12] J. P. Vielma, S. Ahmed, and G. L. Nemhauser, “A lifted linear programming branch-and-bound algorithm for mixed-integer conic quadratic programs,” *Informs Journal on Computing*, vol. 20, no. 3, pp. 438–450, 2007.