Research on the Positioning of AGV Based on Lidar

Yuheng Liu*, Yan Piao and Luyuan Zhang
1School of Electronic and Information Engineering, Changchun University of Science and Technology, Changchun 130022, China
*Corresponding author’s e-mail: 2019100562@mails.cust.edu.cn

Abstract. The rapid development of the e-commerce industry has led to the transformation of the logistics industry from labor-intensive to technology-intensive. The intelligent management system has gradually eliminated the past manual operation methods. AGV (automated guided vehicle, AGV) robot is an indispensable equipment in modern intelligent production and enterprise warehousing logistics systems. In order to improve the positioning performance of the AGV robot to a certain extent, this paper proposes an optimized AMCL (Adaptive Monte Carlo Localization) positioning algorithm based on EKF data processing. First, input the odometer and IMU data into the EKF model for fusion, then the fused state is used as the motion model of the positioning algorithm to predict the pose of the particle set and assist the particle update. The AMCL output after weighted average processing is used as the initial value of scan matching. By constructing a matching function model of lidar observation points and a priori map, using Gauss Newton's method to optimize the solution, the accuracy has been improved.

1. Introduction
Accompanied by people's vigorous pursuit of online shopping forms, intelligent and efficient warehouse management systems appear to have broad application scenarios, and traditional manual operation management models are being replaced step by step. AGV robot is an indispensable equipment in modern intelligent production and enterprise warehousing logistics systems[1]. It can complete efficient and intelligent autonomous movement in a specific space. As people's expectations for the intelligentization of AGV robots increase, as the most critical technology of AGV robots, the research of autonomous navigation technology has become particularly important. The robot's own positioning is the core problem in autonomous navigation. The accuracy of its own location determines the success of navigation. Accurate positioning can reduce time and production costs, which has strong practical significance.

When the AGV robot is working in a complex environment map, the prerequisite for completing the navigation task is accurate self-positioning. Aiming at the problem that the resources generated by the particles in the estimation may be wasted, this paper adopts the AMCL algorithm based on the KLD algorithm that can independently adapt to the current situation and adjust the number of particle samples. The odometer and IMU data are fused through EKF, and the fused data estimation is used as the motion predictive model of the AMCL positioning algorithm to predict the posture of the particle set and assist the update of the particles. Then we use the weighted mean output of AMCL as the initial value of scan matching. By constructing the lidar observation point and the prior map matching function model, in order to improve accuracy, gauss Newton's method is used to optimize the results.
2. AMCL positioning algorithm based on EKF data processing

2.1. Fusion of IMU and odometer data based on EKF

When the robot is moving, the wheels are slipping or bumping, which may cause distortion of the encoder, resulting in a certain degree of deviation in the data of the odometer, and ultimately the robot is not effective in predicting the pose. In order to overcome this situation, this article uses the EKF algorithm to fuse encoder data and inertial measurement unit data.

The binary function formula (1) shows the state of the EKF prediction, and the formula (2) is the measured value and a Gaussian noise will be added afterwards.

$$x_k = g(x_{k-1}, u_k) + \epsilon_k$$
$$z_k = h(x_k) + \delta_k$$

Kalman gain updated to

$$K' = \frac{P_k H^T_k}{H_k P_k H^T_k + \delta_k}$$

$H_k$ is the state transition matrix, $P_k$ is the covariance matrix describing the uncertainty of the state prediction, from which the best estimation of the pose after fusion of the odometer and inertial navigation element data is obtained $\hat{x}_k'$.

$$\hat{x}_k' = \hat{x}_k + K'(z_k - H_k \hat{x}_k)$$

The basic process of EKF fusion is shown in Figure 1.

2.2. AMCL algorithm based on KLD sampling

The purpose of Monte Carlo Localization\cite{5} is an experimental simulation method to solve complex calculation problems. The algorithm needs to use a set of weighted and limited number of particles to try to approximate the posterior probability density of any state\cite{6}. However, since the observation information of the system in real-time is not taken into account, the weight of some particles may be too low, leading to the emergence of particle set degradation problems, resulting in unsuccessful positioning of the robot. When the sensor's perception effect is not accurate enough, the conventional
Monte Carlo-based effect is very poor and cannot solve the problem of robot kidnapping. Fortunately, these problems can be solved by simple exploration algorithms such as adding random particles to a collection of particles.

KDL sampling can change the number of particles over time. It is basically designed to control the number of particles redundant. For example, in a grid map, you need to pay attention to how many grids the particles occupy. It accounts for a lot, indicating that the particles are very scattered. In each iteration of re-sampling, the maximum number of particles can be increased. A small amount means that the particles are already very concentrated, so set the upper limit of the number of samples low to save resources[7].

AMCL[8] is based on Monte Carlo and particle filtering, and uses an adaptive KLD method to update particles. Figure 2 shows the positioning of the initial state and figure 3 shows the positioning of the end state. When the robot is initialized, we cannot get the position of the robot. The particles are scattered in a large area. The denser the particles, the closer to the real position. After the robot moves, the ambiguity of the robot's own location is resolved. The particles will circle around the real position and gradually converge towards a point. Positioning completed

![Figure 2. Positioning of the initial state](image1)

![Figure 3. Positioning of the end state](image2)

3. AMCL with fusion feature matching
The problem of relying on laser sensors for positioning can be considered as a nonlinear least squares problem.

\[
\hat{\xi}^* = \arg \min \frac{1}{2} \| 1 - M \left[ S_i(\xi) \right] \|_2^2
\]

\[
\hat{\xi}^* = \xi + \Delta \xi
\]

In the formula, \( \hat{\xi}^* \) represents the real posture of the robot, \( \hat{\xi} \) represents the estimated posture of the robot, \( \Delta \xi \) represents the deviation between the real posture and the estimated posture, \( S_i(\hat{\xi}) \) represents the coordinates of the end point of the laser scanning ray of the robot at position \( \hat{\xi} \), and \( M \left[ S_i(\hat{\xi}) \right] \) represents the scan. The probability that the end point coordinates are occupied on the map. among them:

\[
S_i(\hat{\xi}) = \begin{pmatrix}
\xi_x \\
\xi_y
\end{pmatrix} + \begin{pmatrix}
cos \theta & -\sin \theta \\
\sin \theta & \cos \theta
\end{pmatrix} \begin{pmatrix}
x_{sens} + z_i' \cos \theta_{sens} \\
y_{sens} + z_i' \sin \theta_{sens}
\end{pmatrix}
\]

\( \xi_x, \xi_y \) Represents the position of the robot under the constructed map coordinates, \( x_{sens}, y_{sens} \) are the position of the laser relative to the robot, \( z_i' \) is the distance scanned by the laser, \( \theta \) is the yaw angle of the robot, and \( \theta_{sens} \) is the angle of the laser beam relative to the heading angle of the robot.

The target of target matching is given an initial value of \( \hat{\xi} \). Using Gauss-Newton iteration to try to get the deviation \( \Delta \xi \), Align the grid maps of the probability distribution of the end points of the laser
scanning to each other. After many iterations, the optimal solution $\xi$ of the pose can be obtained, so that the residual sum of squares of the original model is the smallest. Equation (5) is transformed into:

$$S = \sum_{i=1}^{N} \left(1 - M \left(S_i(\xi^*)\right)\right)^2 \rightarrow 0$$

Because we need to get the smallest residual, This means that the partial derivative of S to $\xi^*$ must be guaranteed to be 0.

$$\frac{\partial S}{\partial \xi^*} = \sum_{i=1}^{N} \left(1 - M \left(S_i(\xi^*)\right)\right) \cdot \frac{\partial M}{\partial S_i(\xi^*)} \cdot \frac{\partial S_i(\xi^*)}{\partial \xi^*} = 0$$

$$\Delta \xi = \sum_{i=1}^{N} \left(\left(\frac{\partial M}{\partial \xi^*}\right)^T \frac{\partial M}{\partial \xi^*}\right)^{-1} \left(\frac{\partial M}{\partial \xi^*}\right)^T \left(1 - M \left(S_i(\xi)\right)\right)$$

Among them,

$$\frac{\partial S}{\partial \xi^*} = \sum_{i=1}^{N} \left(1 - M \left(S_i(\xi^*)\right)\right) \cdot \frac{\partial M}{\partial S_i(\xi^*)} \cdot \frac{\partial S_i(\xi^*)}{\partial \xi^*} = 0$$

Because the probabilistic grid map is discrete and discontinuous, in order to obtain the partial derivative at the end point of the ray on the map $M$, this paper uses the bilinear interpolation method to solve it, but the result of feature matching will fall into a local minimum. This article uses the double three-line interpolation method, first interpolate the four columns, and then interpolate the output in the horizontal direction, as shown in Figure 4.

Figure 4. Bicubic interpolation

$$M(P) = f(f(P_0, P_1, P_2, P_3, y), f(P_4, P_5, P_6, P_7, y), f(P_8, P_9, P_{10}, P_{11}, y), f(P_{12}, P_{13}, P_{14}, P_{15}, y))$$

$$= f(a, b, c, d, x)$$

Among them, $M(P)$ represents the possibility that point $P$ may be used in the constructed map, $P_0 \sim P_5$ represents the probability that the 16 points with the shortest distance from $P$ points are used, which can be obtained by looking up the table and interpolating the first column to get.

$$f(P_0, P_1, P_2, P_3, y) = \left(\frac{1}{2}P_0 + \frac{3}{2}P_1 - \frac{3}{2}P_2 + \frac{1}{2}P_3\right) y + \left(P_0 - \frac{5}{2}P_1 + 2P_2 - \frac{1}{2}P_3\right) y^2 + \left(-\frac{1}{2}P_0 + \frac{1}{2}P_3\right) y^3 + P_3$$

The subsequent process uses the framework of formula (13), and the partial derivative at the end of the laser on the map can be calculated after double three-line interpolation.
Feature matching uses Gauss-Newton iteration method to optimize the objective function, and the output of AMCL is used as the initial value $\xi$ of the objective function. After layer-by-layer iteration, the laser scanning endpoints and the edges of obstacles are kept in a neat and unified state, and the deviation $\Delta \xi$ obtained after optimization minimizes the equation (8), and finally we can get the optimal solution $\xi + \Delta \xi$. Therefore, The improved AMCL is more accurate in positioning itself than the unimproved AMCL.

4. Experiment Analysis

Aiming at the improved positioning algorithm in this article, we conduct experimental tests and analysis indoors. The experiment is expected to analyze the AMCL and the optimized AMCL from the two perspectives of global positioning and positioning after position replacement.

4.1. Global positioning experiment

After the experiment is ready, start the robot and move it about 5M in the same direction. Record the current time when the particle set starts to maintain a stable state, and then control the robot to return to the corresponding position of the initial starting point. Repeat 10 sets of tests. Global positioning experiment data comparison statistics are shown in the table1.

| Robot pose point | Error of original amcl | Error of optimization |
|------------------|------------------------|-----------------------|
|                  | X/m   | Y/m   | $\theta$/rad | X/m   | Y/m   | $\theta$/rad |
| Starting point   | 0.039 | 0.051 | -0.031       | 0.029 | 0.007 | 0.001       |
| Terminal point 1 | -0.033 | -0.012 | -0.022       | -0.013 | -0.009 | -0.012       |
| Terminal point 2 | -0.047 | -0.031 | -0.024       | -0.027 | -0.004 | -0.013       |
| Terminal point 3 | -0.065 | -0.028 | -0.015       | -0.015 | -0.017 | -0.006       |
| Terminal point 4 | 0.027  | 0.022  | -0.008       | -0.008 | -0.029 | 0.005        |
| Terminal point 5 | 0.045  | 0.027  | 0.012        | 0.019  | 0.012  | 0.009        |
| Terminal point 6 | 0.064  | 0.047  | 0.028        | 0.034  | 0.025  | 0.019        |
| Terminal point 7 | -0.059 | -0.050 | -0.017       | -0.019 | 0.018  | 0.008        |
| Terminal point 8 | 0.037  | -0.035 | -0.014       | -0.018 | -0.016 | 0.002        |
| Terminal point 9 | 0.024  | -0.016 | 0.012        | 0.025  | 0.018  | -0.012       |
| Terminal point 10| -0.028 | -0.033 | -0.017       | 0.015  | -0.018 | 0.007        |
| Max error | 0.065 | 0.051 | 0.031 | 0.034 | 0.029 | 0.019 |
|------------|--------|--------|--------|--------|--------|--------|
| Min error  | 0.024  | 0.012  | 0.008  | 0.008  | 0.004  | 0.001  |
| Mean error | 0.043  | 0.032  | 0.018  | 0.019  | 0.016  | 0.080  |

It can be seen from the chart that the 10 sets of global positioning experiments have been completed. The maximum positioning errors of the unoptimized AMCL algorithm are 6.5 cm, 5.1 cm and 0.031 rad, and the average positioning is 4.3 cm, 3.2 cm and 0.018 rad, respectively. After fusing the results and features of the EKF odometer and IMU into AMCL, the optimized algorithm obtained has a maximum positioning error of 3.4 cm, 2.9 cm and 0.019 rad, and the average positioning error is reduced to 1.9 cm, 1.6 cm and 0.080 rad. It can be seen that the position and angle of the improved and optimized algorithm have been greatly improved.

4.2. Relocation experiment
In order to also verify the ability of the improved algorithm to regain positioning when the position of the robot is changed, experiments are designed and compared with the unimproved AMCL algorithm. Verify the ability of the algorithm to respond to the situation of robot kidnapping. Relocation experiment data comparison statistics are shown in the table 2.

| Algorithm  | X/m   | Y/m   | θ/rad |
|------------|-------|-------|-------|
| Original AMCL | 0.049 | 0.037 | 0.017 |
| Optimization | 0.025 | 0.020 | 0.009 |

This experiment can verify that the optimized algorithm also has better recovery ability when the position changes suddenly, but the real-time performance in the relocation process is poor when there is a problem with positioning and start positioning again. It may need to be further improved in future work.

5. Conclusion
This paper studies the positioning accuracy of AGV robots. First, We analyze the principle of the sensor used, and then input IMU and odometer data into the extended Kalman filter, and use the fused state as a KLD-based adaptive Monte Carlo motion model to predict the pose of the particle set. Feature matching is added, and the weighted average output of AMCL is used as the initial value of scan matching. By constructing a lidar observation point and a priori map matching function model, the Gauss Newton method is used to optimize the solution to improve the accuracy. Then we use global positioning and relocation to verify the optimized algorithm. The experimental results show that under the condition that the path of the robot's movement is consistent, the optimized positioning algorithm should be greatly improved compared with the traditional AMCL algorithm. When the position of the robot that has already been positioned is changed, the optimization algorithm still has the ability to re-position, and the positioning accuracy after re-positioning can still maintain a certain degree of credibility. However, it takes a long time to restore the positioning, and it may require future efforts to further optimize the algorithm.

References
[1] Xiao chuanguJin, Meng zhiZhong, Xiong wenQuan, et al. (2016) Dynamic scheduling of mobile-robotic warehouse logistics system. In: Control Conference., 234-240
[2] Bei Xuying, Ping Xueliang, Gao Wenyan. (2018) Calibration of systematic odometry errors for wheeled mobile robots[J]. Application Research of Computers, 35(9):2696-2699+2703.
[3] Liu Shengpan,WANG Wenju,RAO Xingqiao. (2018) Error Analysis and Rotation Scheme Research of Single-axis Rotary SINS., [J].Acta Armam ,39(09):1749-1755.
[4] Yang Shuang, Zeng Bi, He Weiting. (2020) Research on SLAM Technology Combining Semantic Laser and Landmark Information [J/OL]. Computer Engineering and Applications:1-14.

[5] Li Junmin, Li Libo. (2014) Application of Artificial Fish School and Monte Carlo Hybrid Algorithm. Journal of Xi'an University of Science and Technology, 34(2): 224-227

[6] Zheng Chenyao, Dong Zhenjie, Zhang Weiquan. (2014) Comparison of particle filter and no-track particle filter. Ship Electronics Engineering, (12): 47-49

[7] Luo Yuan, Pang Dongxue, Zhang Yi, et al. (2016) Monte Carlo positioning based on adaptive multi-proposal distribution particle filter Algorithm. Computer Applications, 36(8): 2352-2356

[8] Zhang Baoxian, Liu Jun, Chen Haoyao. (2013) AMCL based map fusion for multi-robot SLAM with heterogenous sensors. In: IEEE International Conference on Information & Automation. 822-827