IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 23 March 2024, accepted 3 May 2024, date of publication 9 May 2024, date of current version 17 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3399192

RESEARCH ARTICLE

Indoor Localization of Mobile Robots Based on the Fusion of an Improved AMCL Algorithm and a Collision Algorithm

HONGDA ZHU AND QIANG LUO

Department of Mechanical Engineering, Chongqing Three Gorges University, Chongqing 404100, China

Corresponding author: Qiang Luo (qluo@sanxiau.edu.cn)

This work was supported in part by the Wanzhou District Science and Technology Bureau under Grant wzstc-20230218, and in part by the General Project of the Natural Science Foundation of Chongqing Science and Technology Commission under Grant cstc2020jcyj-msxmX0143.

ABSTRACT The complexity of the environment limits the accuracy of the traditional Adaptive Monte Carlo Localization(AMCL) algorithm, which also suffers from high computational effort and particle degradation due to laser model limitations. To address these issues, an optimized AMCL algorithm with a bounding box is proposed. The AMCL algorithm is first parameterized and initialized to the particle swarm. During the particle iteration process, collision detection is performed on the bounding box. If a collision occurs, the particle filter is not updated and its particle weight is set to 1. If there is no collision, the particle filter is updated normally and the particle weight is set to 0. Then, the particles are resampled and updated based on the measurement data and motion model. After experimental verification, this method's self-localization trajectory is closer to the actual path, and the measurement error fluctuation is smaller. The RVIZ simulation experiments revealed that the overall positioning time was optimized by 18.25% compared to the original AMCL, and by 9.28% compared to the improved AMCL. The optimization algorithm effectively improved the positioning accuracy and robustness of the system.

INDEX TERMS Adaptive monte carlo localization, bounding box, collision algorithm, particle filter, self-localization.

I. INTRODUCTION

Indoor positioning technology for mobile robots is a crucial aspect of robot research. To achieve accurate control of a robot, its autonomous positioning ability cannot be ignored [1]. Currently, there are three main methods to improve the positioning accuracy of mobile robots: multi-sensor fusion positioning technology, equipping them with state-of-the-art sensors or signal receivers, and optimizing and improving positioning algorithms. Multi-sensor fusion positioning technology is a promising trend in navigation and positioning. However, this method faces a common issue of signal switching not being smooth, with the increase of the complexity of the mobile environment and the difficulty of the task [2], [3], particularly near transition points with large signal fluctuations during indoor and outdoor continuous operation, which can affect positioning accuracy [4], [5], [6]. Equipping the robot with state-of-the-art signal receivers or precisely calibrated sensors can reduce sensor errors and improve localization accuracy. However, this upgrade comes at an increased cost and may not be applicable to theoretical research in the laboratory stage. This can be achieved by implementing advanced positioning algorithms or optimizing existing ones to better handle uncertainty, resulting in improved accuracy and reduced input costs. Additionally, this approach can increase the feasibility and success rate of the study. Furthermore, this study investigates the potential for enhancing the accuracy and dynamic obstacle avoidance of mobile robot positioning through the optimization and improvement of the Adaptive Monte Carlo Localization (AMCL) algorithm.

The monte Carlo localization (MCL) algorithm is a particle filtering algorithm commonly used for localization. It can overcome all localization subproblems except robot

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

abduction [7]. However, it suffers from high computational complexity and imprecise localization accuracy [8]. The AMCL algorithm is an optimization and improvement of the MCL algorithm. It achieves better localization accuracy and robustness by sampling a series of assumptions about the current position of the robot, and then evaluating and updating these assumptions to estimate the robot's position. Despite its widespread use in autonomous navigation and localization applications of mobile robots, the traditional AMCL localization algorithm has some drawbacks. The AMCL algorithm requires a significant number of particle sampling and updating operations, resulting in high computational complexity. Inaccurate estimation of the initial position; can hinder efficient particle updating, leading to localization errors. Additionally, long-term motion can cause particle degradation and cumulative errors in sensor data, resulting in inaccurate weight distribution of particles. After identifying these limitations, it is important to optimize and improve them based on the specific situation in practical applications. This will enhance the positioning accuracy of mobile robots; and has significant research value.

In recent years, there have been advancements in the research on improving AMCL-based algorithms both domestically and internationally. Hanten et al. introduced an effective and resilient localization method that is suitable for large-scale indoor environments [9]. In 2011, Zapata et al. proposed an improved Monte Carlo localization algorithm that uses adaptive samples. This reduces the burden of online computation by adopting pre-cache technology and is more efficient than the traditional AMCL localization algorithm [10]. Wang et al. also proposed an improved AMCL robot localization method; that utilizes the method of alternating resampling and KLD sampling in the sampling part, enhancing the filtering efficiency [11]. FENG et al. Also contributed to this field. The authors propose an optimized AMCL algorithm that enhances stability and robustness of the system while improving localization accuracy. They achieve this by adding scan matching and discrete Fourier transform [12]. Additionally, Wang et al. introduce the idea of DNA cross mutation in genetics into the particle iteration process of AMCL, and propose an adaptive Monte Carlo localization method based on the improvement of genetic algorithm [13]. Although these studies provide reference and direction for further optimization and improvement of the AMCL algorithm, there are still significant fluctuations in positioning errors. Particle depletion results in degraded pose tracking performance and other issues. Therefore, this study explores a new method to enhance the positioning accuracy through particle iterative optimization of the AMCL algorithm.

To enhance the AMCL localization algorithm, this study proposes an improved method based on the collision algorithm. The bounding box idea is introduced into the particle iteration process of AMCL, and a collision detection process is added to detect and constrain particle motion. This process affects the efficiency of particle filter update and resampling, ultimately improving localization accuracy. Section II outlines the fundamental principles of the MCL and AMCL localization algorithms. In section III, the AMCL localization algorithm is combined with the bounding box concept. Section IV presents the simulation verification and result analysis conducted under the MATLAB and ROS simulation environment to confirm the feasibility and optimization of the improved algorithm. Section V provides a summary of the paper and discusses the limitations of the experiment. It also outlines potential areas for future optimization.

II. AMCL LOCALIZATION ALGORITHM

A. MCL ALGORITHM

Monte Carlo localization algorithm is an algorithm based on particle filtering algorithm, which is widely used to estimate the pose of a mobile robot in a known environment [14], [15]. It represents bel (x_t) as a collection of n particles.

$$\mathrm{bel}\,(\mathrm{x_t}) = \mathrm{p}\,(\mathrm{x_t}|\mathrm{z}_{1:t}, \mathrm{u}_{1:t}) \tag{1}$$

where bel (x_t) is the posterior probability distribution of state x_t;$z_{1:t}$ denotes all sensor data up to time step t; *and* $u_{1:t}$ denotes the control data.The model will generate a measurement probability p $(z_t|x_t^{[n]})$ for each particle $x_t^{[n]}$.The formula for the weights of the particles is as follows:

$$W_t^{[n]} = \mathrm{p}(z_t|x_t^{[n]})W_{t-1}^{[n]} \tag{2}$$

The weight $W_t^{[n]}$ is the weight of the nth particle,which represents the integral of the measurement probability over time and satisfies a normalization condition: $\sum_{i=1}^{n} W_t^{[i]} = 1$. Thus,the posterior reliability of robot pose x_t can be roughly expressed as:

$$\mathrm{bel}(x_t) \approx \sum\nolimits_{i=1}^{n} W_t^{[i]}\delta(x_t - x_t^{[i]}) \tag{3}$$

where δ represents the sample estimated probability.

MCL represents the posterior via a set of weighted particles, however, the MCL algorithm does not solve the robot abduction problem, and localization fails as soon as the positional changes are discontinuous. To improve the localization accuracy, a large number of particles need to be added, which slows down the localization convergence.

B. AMCL ALGORITHM

The AMCL is a common technique for mobile robot localization problem [16], and its visual interpretation is shown below.

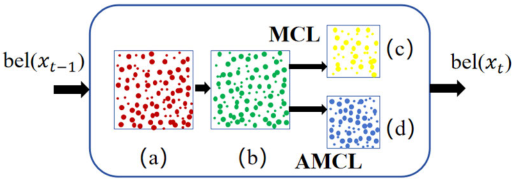


FIGURE 1. MCL and AMCL visual explanation.

In Fig.1, the (a) process represents the sampling process,where the red circles indicate the particles distributed on the map with respect to bel (x_{t-1}); the (b) process represents the weighted weighing stage,where the larger green circles indicate the larger particle mass,and vice versa; the (c) process represents the MCL resampling process for a constant number of particles N; and the (d) process represents the AMCL resampling process for a varying number of particles N. Continuing with this recursion, the particles are eventually concentrated around the actual location of the robot.

The principle of AMCL algorithm is to improve MCL by adaptively adjusting the number of particle samples, which determines whether the robot is abducted or not based on the long term estimation weight ω_{slow} and the short term estimation weight ω_{fast}, and if ω_{fast} is inferior to ω_{slow}, the robot recovers its relocation from abduction by adding random particles in the resampling, the expression is:

$$\begin{cases} \omega_{slow} = \omega_{slow} + \alpha_{slow}(\omega_{avg} - \omega_{slow}) \\ \omega_{fast} = \omega_{fast} + \alpha_{fast}(\omega_{avg} - \omega_{fast}) \end{cases} \tag{4}$$

where ω_{avg} denotes the average weight of all particles,and the parameters α_{slow} and α_{fast} are the attenuation rates of the exponential filters averaged over the long-term and short-term estimates, respectively ($0 \leq \alpha_{slow} \leq \alpha_{fast}$). The basic flow of the AMCL localization algorithm is shown in the following figure.

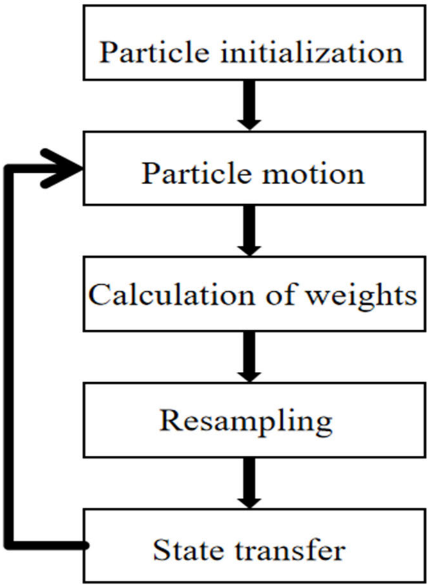


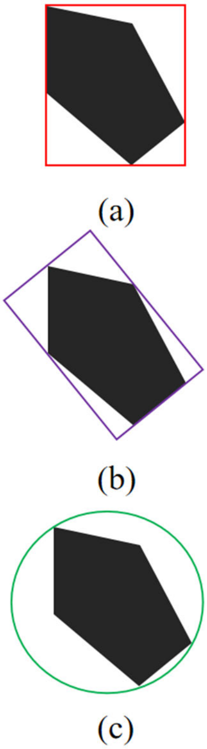
FIGURE 2. Basic steps of the AMCL algorithm.

The traditional AMCL localization algorithm is highly adaptive to particles, but its position estimation accuracy is not optimized. Therefore, how to improve it to increase the positioning accuracy in terrains prone to positioning failures and apply the optimized AMCL to mobile robotic systems is the next research focus of this paper [17], [18].

III. GIMPROVEMENT OF AMCL BASED ON BOUNDING BOX

A. BOUNDING BOX IDEA

The bounding box algorithm is a technique used to solve the optimal wrap-around space of a discrete set of points. It is widely used in collision detection, real-time obstacle avoidance, and other functions. The basic idea is to replace complex geometric objects with slightly larger geometric objects with simple characteristics [19]. The most common bounding box algorithms are *AABB*(axis-aligned bounding box), bounding sphere, *OBB*(oriented bounding box), *FDH*(fixed directions hulls).


(a)

(b)

(c)

FIGURE 3. (a) AABB; (b) OBB; (c) bounding sphere.

Figure 3 presents a simple bounding box model. The black area represents an irregular object. The red line in the figure represents *AABB*, which has a simple construction and requires less storage space. The purple line in the figure represents *OBB*, which has better tightness. Finally, the green line in the figure represents the bounding sphere.

This is an example of the OBB bounding box, which demonstrates the basic principle of the algorithm [20]. OBB refers to the smallest cuboid that contains the detected objects in any direction with respect to the coordinate axis, and its direction is arbitrary [21]. To calculate the OBB bounding box, find the center point of the rectangle formed by the given points and solve for it using the covariance matrix. The covariance matrix represents the correlation of the points when combined together. It measures the extent to which each dimension deviates from its mean. Use the following formula to calculate the covariance:

$$\text{cov(x,y)} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{n-1} \tag{5}$$

And satisfies the relation.

$$\text{cov(x,y)} = \text{cov(y,x)} \tag{6}$$

The covariance matrix can be calculated by applying the above equation.

$$\text{C} = \begin{matrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{matrix} \tag{7}$$

Upon calculating the covariance matrix, the eigenvectors and eigenvalues are obtained by diagonalizing the matrix. The eigenvectors are then utilized to derive the coordinate axes of the OBB. Based on the known points, the center coordinates of the OBB coordinate axes, as well as the length and width of the rectangle, can be derived. Finally, the OBB bounding box can be determined.

B. IMPROVED AMCL ALGORITHM

To enhance location accuracy and efficiency, we introduce the bounding box concept of collision algorithm into the particle iteration process of AMCL, resulting in an improved adaptive Monte Carlo localization method. Please refer to the following figure for a detailed procedure.

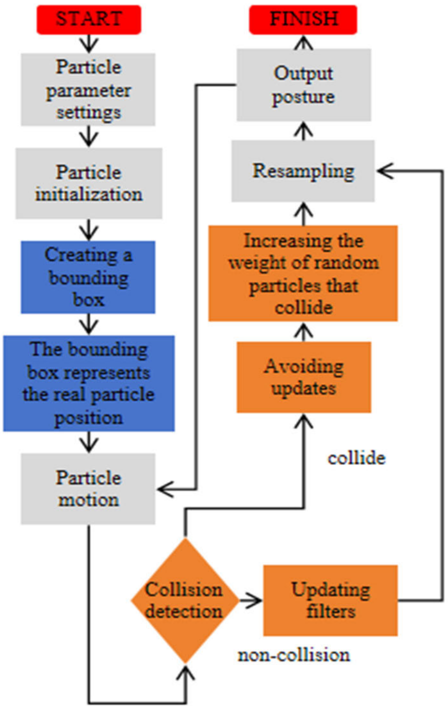


FIGURE 4. Flowchart of improved AMCL localization algorithm based on the idea of bounding box.

Figure 4 shows the process used in this study. Firstly, the AMCL localization algorithm is parameterized and the particle swarm is initialized. Then, a bounding box is created for the positional pose where the real particles are located. The box is rectangular and centered on the real position of the particles, with the geometric centers of all the particle swarms also located inside the box. Collision detection is then performed on the moving particles. The specific collision detection process will be elaborated in the next section. Here, the box is used to detect whether the real particles collide with other random particles (obstacles) or not.

The bounding box is the process of aligning the real position of particles with the existing map, and this study adopts the AABB bounding box, the AABB is a smallest hexahedron that contains a given object, whose important characteristic is that each face of the rectangle corresponding to the bounding box is parallel to the plane of a certain coordinate axis, based on which, only six values are needed to determine the AABB bounding box [22], which represent the minimum and maximum values of the bounding box on each coordinate axis, i.e., x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, and z_{max}. In other words,both the real position of the particle and its surrounding similar random particles must satisfy the following conditions.

$$\begin{cases} x_{min} \leq x \leq x_{max} \\ y_{min} \leq y \leq y_{max} \\ z_{min} \leq z \leq z_{max} \end{cases} \tag{8}$$

In addition, the six parameters representing the AABB bounding boxes can be categorized into the following two groups.

$$\begin{cases} P_{min} = [x_{min}, y_{min}, z_{min}] \\ P_{max} = [x_{max}, y_{max}, z_{max}] \end{cases} \tag{9}$$

where P_{min} is the set of minimum values of 3 axis coordinates; P_{max} is the set of maximum values of 3 axis coordinates, based on which we can get the coordinates of the geometric center of the AABB bounding box, i.e., the real orientation of the particle. The formula is as follows.

$$\text{C} = (P_{min} + P_{max})/2 \tag{10}$$

The bounding box is introduced into the particle iteration process of AMCL, The initial state diagram of random particles is shown in the following figure, in this study, it is assumed that the actual location of the particle under investigation is known, where the red dots represent the actual positions of the particles, the subsequent position of the robot's movement is analogous to the initial position; the black dots represent all 500 random particles, and the yellow dots represent the geometric center points of all the particles.

As illustrated in the accompanying figure, both the actual position of the particle and the geometric center of the random particle are within the bounding box. The actual position of the particle is a fixed coordinate(50,42). The subsequent particle motion and collision detection is based on the bounding box. The subsequent section will concentrate on the particle collision detection process, which represents the most crucial aspect of this study.

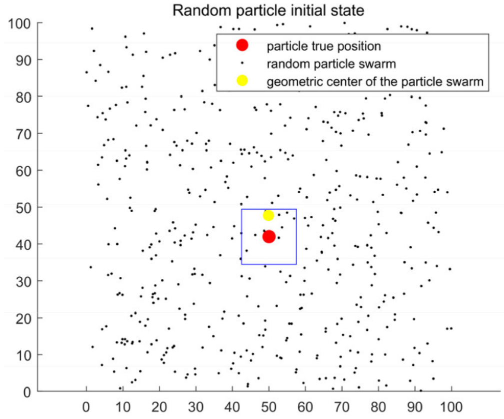


FIGURE 5. Random particle initial state diagram.

C. ADD COLLISION DETECTION PROCESS

Collision detection is a crucial technique used in many robotic applications, including motion planning, trajectory optimization, and simulation. It determines whether a robot has collided with an obstacle [23]. By adding a collision detection process to the AMCL algorithm, the robot can detect its surrounding environment while performing real-time localization to determine if it has collided with an obstacle.

After initializing the particles and creating the bounding box, a collision detection process is added for the moving random particles. Collision detection involves calculating the bounding parameters of the box and the coordinates of the random particles. It is assumed that the center coordinates of the box are (x_a, y_a), and the coordinates of the real particles are (x_a, y_a). The width and height of the box are w_a and h_a, respectively. And set the center coordinates of the random particles as (x_p, y_p), so as to compute the bounding parameters of the box.

$$\begin{cases} left_a = x_a - w_a/2 \\ right_a = x_a + w_a/2 \\ top_a = y_a + h_a/2 \\ bottom_a = y_a - h_a/2 \end{cases} \tag{11}$$

Then determine whether the random particle collides with the bounding box, if $left_a \leq x_p \leq right_a$ and $bottom_a \leq y_p \leq top_a$, the random particle collides with the bounding box, otherwise, no collision occurs. During the movement of the bounding box, if the random particle collides with the bounding box, then choose not to update the AMCL filter in order to avoid incorporating the wrong measurement data into the filter and set its particle weight to 1, which indicates that the position is credible if the random particle does not collide with the bounding box, then update the AMCL filter normally and set its particle weight to 0, which indicates that the position is not credible. After completing the collision detection, continue to resample and update the particles based on the measurement data and motion model.

By introducing collision detection of the bounding box, the AMCL particle iteration remains close to the true position. This reduces the computational amount of the AMCL algorithm; and improves the efficiency of subsequent particle weight calculation and resampling. As a result, the localization accuracy and stability are improved.

The specific step-by-step process of the improved AMCL proposed in this study is as follows:

1. Parameterization: The parameters to be defined include the total number of particles N, process noise Q, measurement noise R, random particle population P, particle weights W, and any other relevant parameters. It is important to note that the observed state of the initial system is the true bitmap superimposed on Gaussian noise.

2. Initialize the swarm and create a wraparound box for the real positions of the particles: N randomly distributed particles on the map are generated to form an initial random particle population P, and their distances from the measured positions and initial weights are found. Then the enclosing box is created at the real particles, as detailed in Section III-B.

3. Simulation of particle motion: Iterating over the initialized random particle swarm, the particles constantly update their velocities and positions by referring to the individual extremum T and the global extremum O. The expressions are as follows.

$$\begin{cases} V_i^{(t+1)} = \omega V_i^{(t)} + c_1 r_1(\mathrm{T}-x_i^{(t)}) + c_2 r_2(\mathrm{O}-x_i^{(t)}) \\ x_i^{(t+1)} = x_i^{(t)} + v_i^{(t+1)} \end{cases} \tag{12}$$

where $V_i^{(t+1)}$ is the velocity of the particle at the next moment; $x_i^{(t+1)}$ is the position of the particle at the next moment;ω is the inertia weight; c_1, c_2 are the acceleration constants, which usually take the value of 2; and r_1, r_2 are the uniformly distributed random numbers in the interval(0,1).

4. Collision detection for the enclosing box and surrounding particles.

5. Classify the particle swarm: classify the random particles according to the collision detection results, assign larger weights to the particles that have collided and keep them, indicating that these random particles are closer to the real particles and more credible; assign smaller weights to the particles that have not collided.

$$x_k^i \in \begin{cases} C_L, & w_k^i \leq W_T \\ C_H, & w_k^i > W_T \end{cases} \tag{13}$$

where C_L is the set of small weighted particles, C_H is the set of large weighted particles, w_k denotes the normalized weights,and W_T is the threshold for particle weight classification.

6. Resampling and updating of particles after recalculating weights.

IV. SIMULATION EXPERIMENT VERIFICATION AND ANALYSIS

A. EXPERIMENTAL ENVIRONMENT

The comparison between the traditional AMCL localization algorithm and the optimized AMCL localization algorithm was conducted using ROS Noetic on Ubuntu 20.04. The robot model, which is equipped with LIDAR and camera sensors, is depicted in the figure below. It is a four-wheel differential steering robot model.

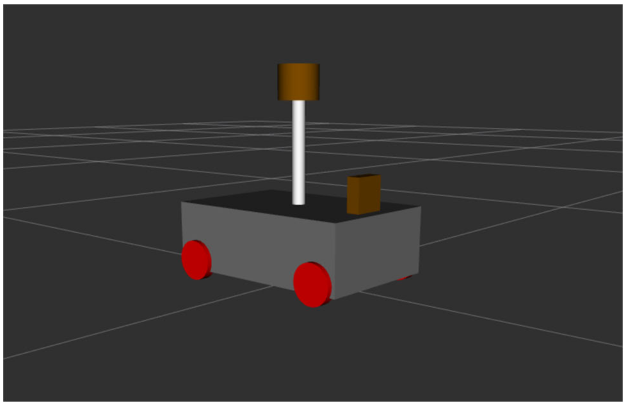


FIGURE 6. Robot global model diagram.

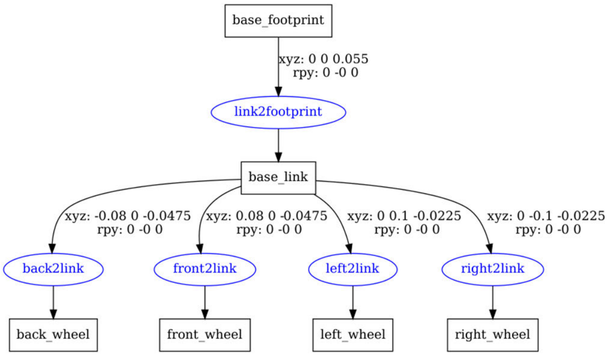


FIGURE 7. Robot model chassis tree diagram.

The robot model is constructed by configuring the connecting rod and joint parameters. The connecting rod refers to the rigid part of the robot, which includes the robot chassis, wheels, LIDAR, camera and other sensors. The joints serve as connection points between two connecting rods. Figure 7 shows the design of the connecting rods and joints for the specific chassis model.

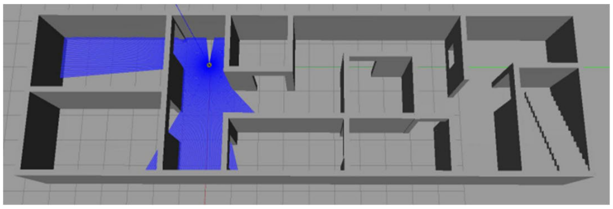


FIGURE 8. Simulation environment map.

The simulation experiment utilized Gazebo software to simulate the physical and collision characteristics of the real world. The robot's movement was controlled through the keyboard control node. Figure 8 shows the house-like interior environment created using Gazebo software, which includes components such as the ground plane, walls, and stairs. The initial pose of the robot model is fixed in a specific location within the simulated indoor environment. Subsequent positioning experiments are conducted based on this initial configuration. The figure shows the scanning area of the LIDAR, represented by the blue area. The scanning angle range is from −3 rad to 3 rad.

B. SIMULATION EXPERIMENT ANALYSIS

To assess the effectiveness of the proposed method in this paper, we conducted simulation studies using MATLAB 2023a and RVIZ simulation software. We first constructed a regional environment in MATLAB; and specified the simulation parameters, which are detailed in Table 1.

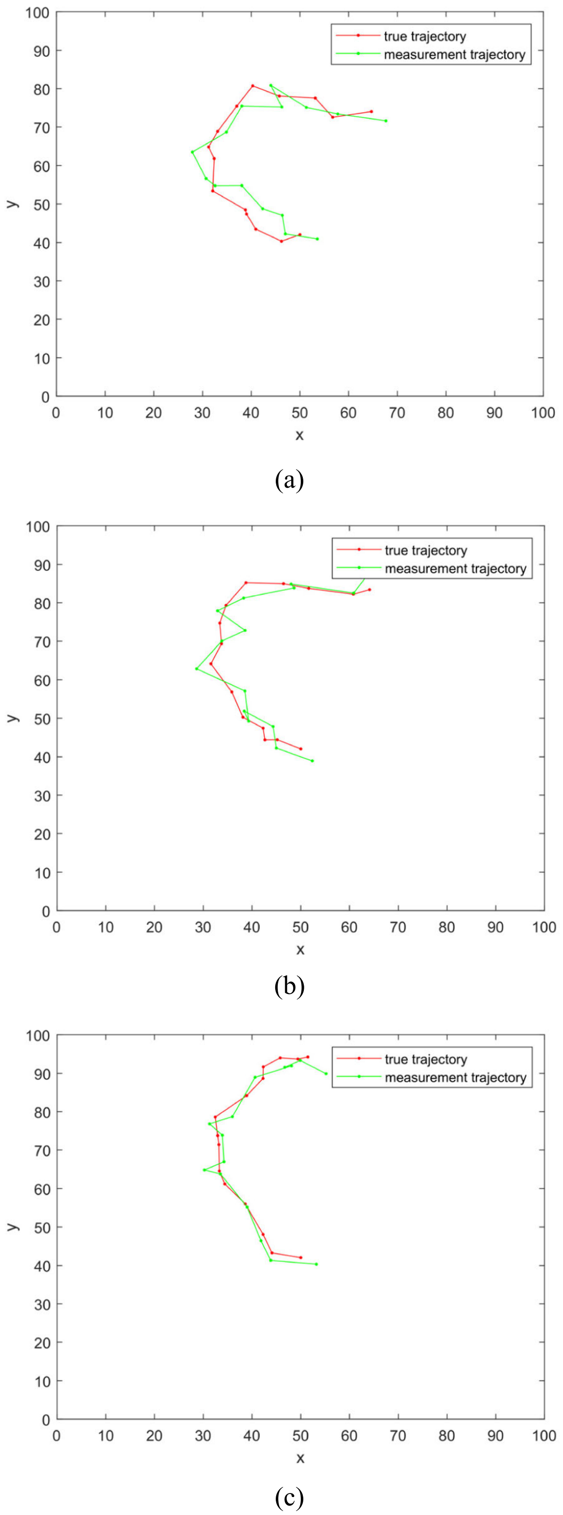
TABLE 1. Key parameter settings.

Parameters	Values
Total particles	500
Process noise	6
Measured noise	6
Measuring time	15
Rotation angle	Pi/T
Distance	80/T
Worldsize	100

During the simulation process, the initial value of the particle is set to the real value, and the positioning error is set to zero. At each new moment, 500 random particles must undergo one random movement according to the model without exceeding the region. Subsequently, the predicted, corrected, resampled, and filtered random particles are generated by the original AMCL algorithm, the improved AMCL algorithm, and the improved AMCL algorithm in this paper, respectively.

Under the conditions of the above simulation environment, the anti-jamming performance of the traditional AMCL positioning algorithm, the improved AMCL positioning algorithm in reference 24 and the optimized AMCL positioning algorithm in this paper are studied respectively. Firstly, the process noise Q = 6 and the measurement noise R = 6 are set, and then the self-positioning simulation experiments of random particles are carried out using these three algorithms respectively [24].

As can be seen from the figure above, the coordinates of the real position of the particle after 15 movements are recorded and connected with red thin lines as the real trajectory of the particle. The coordinates of the measurement of different algorithm models are recorded and connected with green thin lines as measurement trajectories. Conversely, the closer the red line is to the green line, the more accurate the self-positioning trajectory is likely to be. Conversely, a larger interval indicates a lager error in the algorithm model. As illustrated in (a), there are discernible contrasts between the two trajectories.(b) is more optimized than (a), and the



(a)

(b)

(c)

FIGURE 9. Comparison diagram of trajectory simulation: (a) Original AMCL algorithm; (b) Contrastive AMCL algorithm(The figure references the 2018 article by Cao et al. for comparison); (c) Improved AMCL algorithm fused with collision algorithm.

self-positioning trajectory in (c) is most closely aligned with the real trajectory. The enhanced algorithm has been demonstrated to exhibit robust anti-interference capabilities.

Furthermore, the figure below depicts the particle swarm error and measurement error of the three algorithms. The particle swarm error is defined as the distance between the geometric center of all random particles and the actual particle. The measurement error is defined as the distance between the particle pose predicted by different algorithm models and the real pose.

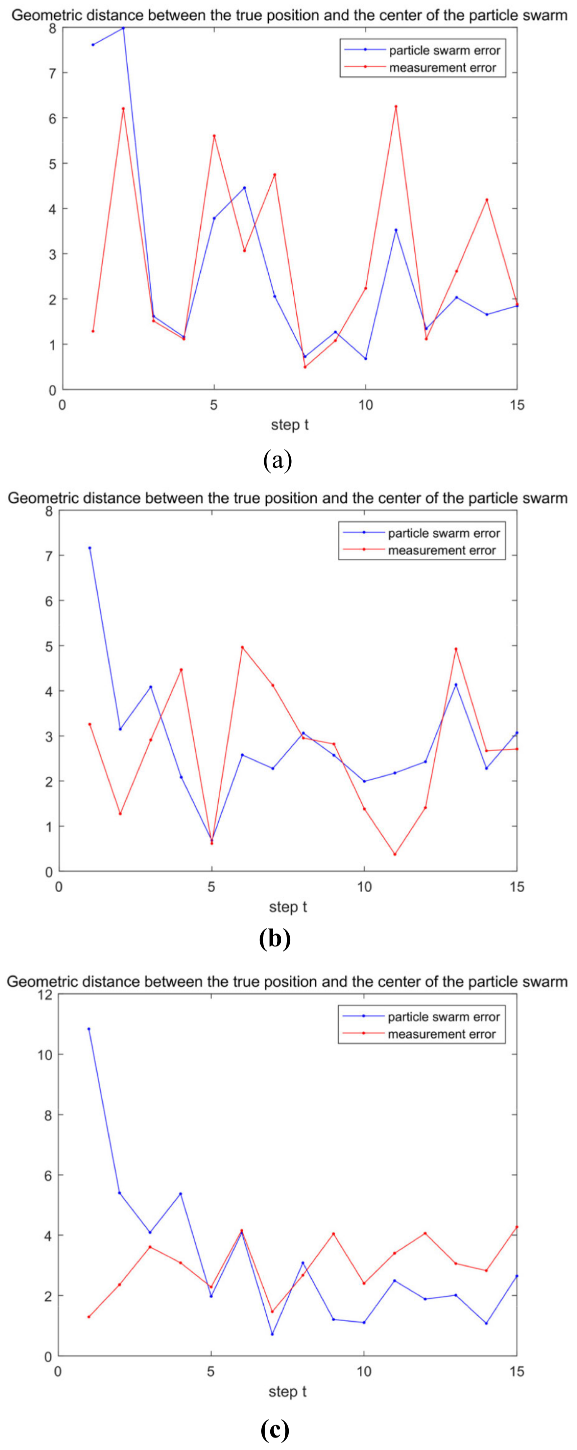
Figure 10 (a) shows the particle swarm error and measurement error of the traditional AMCL positioning algorithm; (b) The error graph of the improved AMCL algorithm in reference 24 is shown; (c) The error graph of the AMCL algorithm optimized in this paper is shown. The blue line represents particle swarm error. As can be seen from the figure above, the positioning error of the traditional AMCL positioning algorithm is about 3.78m when tracking from the initial position of the real particle to its movement to 5s. (b) Approximately 0.68m in the figure; (c) Approximately 1.97m in the figure. After 2s of self-positioning, due to the addition of closed frame and collision detection process in the optimization algorithm in this paper, the particle swarm error in figure (c) has some fluctuations, which is a normal phenomenon. The measurement errors of the three algorithms are shown by the red line in the figure. (a) The measurement error of the traditional AMCL positioning algorithm in the figure is maintained between 0m and 7m, which fluctuates greatly; (b) The measurement error of the improved positioning algorithm in the figure is maintained between 0m and 6m; (c) The measurement error in the figure is kept between 1m and 5m with small fluctuation and basically converges to a stable state.

Repeated localization experiments were carried out on the three algorithms, and the improved algorithm in reference 24 was set as the contrastive group. Each experimental group was randomly conducted 15 particle iterations, including 15 groups of data, and 15 parallel repeated positioning experiments were carried out for each experimental group. Subsequently, the mean value of the 15 groups of data was calculated to obtain the mean value of particle errors at different times, as illustrated in the following table.

The three groups of particle measurement error data were visualized in the simulation software, and the average particle error graph of the three groups of experiments were obtained, as shown in the figure below.

As can be seen from the figure above, the optimized positioning algorithm has the smallest average particle measurement error. It can be seen that when the bounding box and collision detection process are introduced for real particles, the measurement error of the improved algorithm is smaller and the localization anti-interference ability is stronger.

The three localization algorithms are simulated in RVIZ software, and a complete raster map and its path planning are obtained. The following figure shows the cost map model in robot model positioning and navigation. (a) The figure shows the simulation results of traditional AMCL positioning algorithm; (b) The figure shows the simulation results of the



(a)

(b)

(c)

FIGURE 10. Comparison diagram of localization error: (a) Original AMCL algorithm; (b) Contrastive AMCL algorithm(The figure references the 2018 article by Cao et al. for comparison); (c) Improved AMCL algorithm fused with collision algorithm.

TABLE 2. Particle measurement error data of three algorithms.

Steps	Original AMCL	Contrastive AMCL	Optimized AMCL
1	0.394177239	0.397999778	0.127198971
2	3.064232509	2.806851127	2.714052092
3	3.155254738	3.158165865	2.313427035
4	2.573441495	3.230066502	2.621056716
5	3.13448224	2.567609684	3.082326389
6	2.997655088	3.18401902	3.027539975
7	2.468242141	2.817903609	2.156029733
8	2.370886677	2.684147221	2.586000512
9	3.012538583	2.63504594	2.739044339
10	3.096057898	2.90861126	2.494676514
11	3.118400729	2.832078789	2.194417959
12	2.838994867	2.868419472	2.784914331
13	3.144156301	3.462025257	2.888148493
14	2.989545282	3.3765752	2.200305657
15	4.152825963	3.531450137	3.362097224

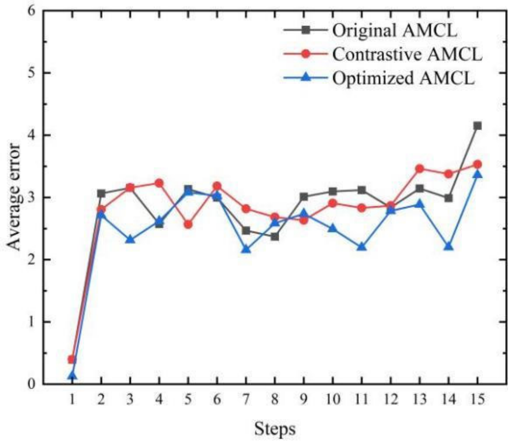


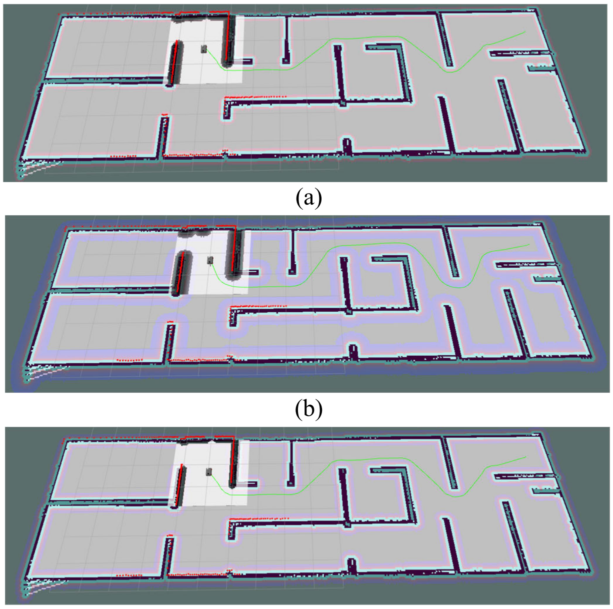
FIGURE 11. Positioning experiment simulation.

control group; (c) The figure shows the simulation results optimized in this paper. The white area is the local cost map display area, the black area represents the expansion radius of the obstacle, the red area is the radar scanning area of the robot model, and the green line represents the final planned path of the positioning experiment.

TABLE 3. Operational data of three algorithms.

Optimized or not	Original AMCL	Contrastive AMCL	Optimized AMCL
Time /s	26.30	23.70	21.50
Path length /m	7.61	7.92	7.67
Number of collisions	2	0	0

As can be seen from Figure 12, although the path planned by the traditional AMCL positioning method is short, it does not take into account the size of the robot and obstacle avoidance and other issues, and it is easy to collide with obstacles in the autonomous navigation, thus causing the robot to fall into a state of suspension, and repositioning requires time to generate a new path, resulting in low overall positioning efficiency. In the control group, the expansion radius was added to the obstacle, so that the robot model maintained a safe distance when turning or approaching the obstacle, which ensured the safety and fluency of the overall operation, but increased the length of the path, making the time consumed too long. By adding the bounding box and collision detection process, the optimization algorithm in this paper can detect the distance between the robot and the obstacle in real time during the operation, realize the dynamic obstacle avoidance function, and ensure the efficiency and safety of the operation.



(a)

(b)

(c)

FIGURE 12. Positioning experiment simulation: (a) Original AMCL algorithm; (b) Contrastive AMCL algorithm (The figure references the 2018 article by Cao et al. for comparison); (c) Improved AMCL algorithm fused with collision algorithm.

Through data processing, the specific data in Table 3 is obtained. From the data in the above table, it can be found that although the original AMCL positioning algorithm has the shortest path length, it has experienced two collisions and does not meet the relevant requirements of positioning and navigation. The path length of the control group was the longest and the localization efficiency was not high. Although the path planning length of the optimized algorithm in this paper is not significantly optimized, only 0.25m shorter than that of the control group, the overall positioning time optimization effect is significant, and the efficiency is increased by 18.25% compared with the original AMCL positioning algorithm. Compared with the control group, the efficiency was improved by 9.28%. It can be seen that the optimized AMCL positioning algorithm proposed in this paper is feasible and effective in the simulation model, and also lays a theoretical foundation for further improving the positioning and navigation efficiency of the solid robot.

V. CONCLUSION

This paper proposes a global indoor localization method based on the collision algorithm with improved AMCL. The traditional AMCL localization algorithm has several issues, including large computation, degradation of position tracking performance due to particle depletion, and slow recovery of localization. The proposed method aims to address these issues.

1. The idea of bounding box in collision algorithm is introduced into the particle iteration process of AMCL, by creating a suitable bounding box for the real position of particles, and the geometric centers of all the particle swarms are also located in the bounding box, and the bounding box will move accordingly with the trajectories of the real particles in the subsequent random particle movement process.

2. Collision detection on the motion of random particles, which is also the core part of this study. By adding the collision detection algorithm, determine whether the random particles collide with the bounding box, if the collision occurs, it means that the random particles are closer to the position of the real particles, so set their particle weights to 1, indicating that the position is highly credible; if the collision does not occur, update the AMCL particle filter normally, and set their particle weights to 0, indicating that the position is not to be trusted. After completing the collision detection, continue to resample and update the particles according to the measurement data and the motion model, and so on recursively, the random particles are finally concentrated around the real particles.

3. The improved AMCL localization algorithm was simulated and verified. Firstly, simulation experiments were carried out in MATLAB, and the program was written through the steps of parameter setting, initializing particle swarm, creating bounding box, particle movement, collision detection, resampling, etc. The trajectory and error comparison charts were finally obtained, and it can be seen through the charts that the improved localization algorithm has feasibility. Then the improved algorithm was verified again in RVIZ simulation software, firstly, a robot model was built and an indoor environment was built in gazebo simulation software, and then indoor localization experiments were carried out for the three methods respectively. According to the data analysis, compared with the original AMCL positioning algorithm, the complete positioning time is optimized by 18.25%, and the path length is shortened by 0.25m compared with the contrastive AMCL positioning algorithm. The experiment verifies the optimization of the improved method.

This paper presents a novel approach to improving the positioning efficiency and stability of a robotics system. The algorithm combines the AMCL positioning algorithm with a collision algorithm, and the experimental results demonstrate that this integration enhances the overall performance of the system. Nevertheless, the successful operation of the improved algorithm in this paper is contingent upon the known initial pose of the robot. In the future, further research will be conducted on the global positioning problem. The improved, mature algorithm will be transplanted to the solid robot in order to achieve higher positioning accuracy for the mobile robot.

DATA AVAILABILITY STATEMENT

Not applicable.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

ABBREVIATIONS

The following abbreviations are used in this manuscript:

AABB	Axis-aligned bounding box.
OBB	Oriented bounding box.
FDH	Fixed directions hulls.
AMCL	Adaptive monte carlo localization.
MCL	Monte carlo localization.

REFERENCES

[1] X. Yang, F. Huang, and S. Yan, "UWB/IMU/odometer fusion localization method based on improved UKF," *Chin. J. Inertial Technol.*, vol. 31, no. 5, pp. 462–471,2023.

[2] Q. Luo, H. B. Wang, Y. Zheng, and J. C. He, "Research on path planning of mobile robot based on improved ant colony algorithm," *Neural Comput. Appl.*, vol. 32, no. 6, pp. 1555–1566, 2020.

[3] M. Yu, Q. Luo, H. Wang, and Y. Lai, "Electric logistics vehicle path planning based on the fusion of the improved A-star algorithm and dynamic window approach," *World Electr. Vehicle J.*, vol. 14, no. 8, p. 213, Aug. 2023.

[4] X. Shi, D. Zhao, and Z. Kong, "High-precision vehicle localization technology based on multi-sensor information fusion," *China Mech. Eng.*, vol. 33, no. 19, pp. 2381–2387, 2022.

[5] X. Hou, B. Xu, Y. Zhou, D. Wang, and Y. Lu, "Continuous localization based on GPS/INS/magnetometer multi-sensor fusion," *J. Sens. Technol.*, vol. 33, no. 9, pp. 1320–1326, 2020.

[6] W. Qian, X. Chen, X. Chen, and B. Sun, "Research on continuous indoor and outdoor positioning method based on odometry/LiDAR/GNSS," *J. Sens. Technol.*, vol. 35, no. 4, pp. 523–529, 2022.

[7] A. Yilmaz and H. Temeltas, "Self-adaptive Monte Carlo method for indoor localization of smart AGVs using LiDAR data," *Robot. Auto. Syst.*, vol. 122, Dec. 2019, Art. no. 103285.

[8] W. Xiaoyu, L. Caihong, S. Li, Z. Ning, and F. Hao, "On adaptive Monte Carlo localization algorithm for the mobile robot based on ROS," in *Proc. 37th Chin. Control Conf. (CCC)*, Jul. 2018, pp. 5207–5212.

[9] R. Hanten, S. Buck, S. Otte, and A. Zell, "Vector-AMCL: Vector based adaptive Monte Carlo localization for indoor maps," in *Proc. Intell. Auton. Syst. 14th Int. Conf. (IAS)*. Cham, Switzerland: Springer, 2017, pp. 403–416.

[10] L. Zhang, R. Zapata, and P. Lépinay, "Self-adaptive Monte Carlo localization for mobile robots using range finders," *Robotica*, vol. 30, no. 2, pp. 229–244, Mar. 2012.

[11] N. Wang, J. Wang, and H. Li, "An improved localization method for AMCL robot," *J. Navigat. Positioning*, vol. 7, no. 3, pp. 31–37, 2019.

[12] J. Feng, D. Pei, Y. Zou, and B. Zhang, and P. Ding, "An improved AMCL algorithm based on robot laser localization," *Adv. Laser Optoelectronics*, vol. 58, no. 20, pp. 479–487, 2021.

[13] Z. Wang, B. Yan, M. Dong, J. Wang, and P. Sun, "Localization method for wall climbing robot based on LiDAR and improved AMCL," *J. Instrum.*, vol. 43, no. 12, pp. 220–227, 2022.

[14] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust Monte Carlo localization for mobile robots," *Artif. Intell.*, vol. 128, nos. 1–2, pp. 99–141, May 2001.

[15] P. Pfaff, W. Burgard, and D. Fox, "Robust Monte-Carlo localization using adaptive likelihood models," in *Proc. Eur. Robot. Symp.* Berlin, Germany: Springer, 2006, pp. 181–194.

[16] S. He, T. Song, P. Wang, C. Ding, and X. Wu, "An enhanced adaptive Monte Carlo localization for service robots in dynamic and featureless environments," *J. Intell. Robotic Syst.*, vol. 108, no. 1, pp. 1–17, May 2023.

[17] T. Y. Lim, C. F. Yeong, E. L. M. Su, S. M. Shithil, S. F. Chik, F. Duan, and P. J. H. Chin, "Enhanced localization with adaptive normal distribution transform Monte Carlo localization for map based navigation robot," *ELEKTRIKA-J. Electr. Eng.*, vol. 18, nos. 3–2, pp. 17–24, Dec. 2019.

[18] M.-A. Chung and C.-W. Lin, "An improved localization of mobile robotic system based on AMCL algorithm," *IEEE Sensors J.*, vol. 22, no. 1, pp. 900–908, Jan. 2022.

[19] F. Lin, L. Zou, and C. Zhang, "A fast collision detection algorithm based on hybrid hierarchical enclosing box," *Comput. Simul.*, vol. 40, no. 9, pp. 454–457, 2023.

[20] S. Chen, Y. Guan, Z. Shi, and G. Wang, "Formalization of collision detection methods for robots," *J. Softw.*, vol. 33, no. 6, pp. 2246–2263, 2022.

[21] B. Gan and Q. Dong, "An improved optimal algorithm for collision detection of hybrid hierarchical bounding box," *Evol. Intell.*, vol. 15, no. 4, pp. 2515–2527, Dec. 2022.

[22] C. Tu and L. Yu, "Research on collision detection algorithm based on AABB-OBB bounding volume," in *Proc. 1st Int. Workshop Educ. Technol. Comput. Sci.*, 2009, pp. 331–333.

[23] L. Montaut, Q. L. Lidec, A. Bambade, V. Petrik, J. Sivic, and J. Carpentier, "Differentiable collision detection: A randomized smoothing approach," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2023, pp. 3240–3246.

[24] F. Cao and Q. Fan, "Research on real-time localization based on adaptive Monte Carlo algorithm," *Comput. Eng.*, vol. 44, no. 9, pp. 28–32, 2018.

HONGDA ZHU was born in Dazhou, Sichuan, China, in 2000. He received the B.Eng. degree from North China Electric Power University, Baoding, China, in 2022. He is currently pursuing the M.Eng. degree in electronic technology of electromechanical system with Chongqing Three Gorges University, Chongqing, China.

A review paper was published, in 2023. His research interests include multi-sensor fusion positioning technology and mobile robot indoor positioning research.

QIANG LUO received the B.Eng. degree from Tiangong University, Tianjin, China, in 2010, and the M.Eng. degree from Beijing Jiaotong University, Beijing, China, in 2013. From 2017 to 2018, he was enrolled at the University of Maryland, pursuing studies in the field of robotics.

Since 2020, he has been an Associate Professor with the Department of Mechanical Engineering, Chongqing Three Gorges University. He has published more than ten academic articles, applied for more than 20 patents and presided more than ten scientific research projects. His research interests include intelligent agricultural machinery equipment, automatic control, and intelligent robot. He is a member of the Artificial Intelligence Branch of Chongqing Agricultural Machinery Society and China Image and Graphics Society.