Advances in Computer Science Research, volume 88

International Conference on Computer, Network, Communication and Information Systems (CNCI 2019)

Design of Autonomous Navigation Algorithm for Security Inspection Robot

Xiaowei Han[1, a], Changwen Liu[2, b]

[1]Institute of Science and Technology Innovation Shenyang University, Shenyang, China

[2]Graduate School Shenyang University Shenyang, China

[a]Hanxw69@163.com, [b]986481856@qq.com

Abstract. The security inspection robot is mainly used for large-scale warehouse security and automatic inspection of various warehouses, community security inspections, and industrial site monitoring and other automated inspections. The security inspection robot uses the SLAM algorithm to map the inspection target data and the odometer data obtained by the laser radar triangulation technology under the ROS operating system, and determines the position of the robot and the solid obstacle in the environment. After calibrating the main target points of the inspection and the inspection sequence on the map, the improved ant colony algorithm is used for static path planning and obstacle avoidance between each target point, and then the improved BUG algorithm is adopted in the complex dynamic environment. Obstacle avoidance, so as to achieve the inspection mission of the security inspection robot.

Keywords: Security inspection robot, path planning, SLAM algorithm, ant colony algorithm, BUG algorithm.

1. Introduction

In today's society, large-scale warehouses that store important materials such as grain and industrial products, as well as daily community security, industrial field equipment and daily operations, are all facing material security issues. In view of this, a security inspection robot that can be applied to large warehouse security and automatic inspection of various warehouses, community security inspection, industrial site monitoring and other automatic inspections is designed. This paper focuses on the design of autonomous navigation algorithms for security inspection robots.

At present, the commonly used path planning and obstacle avoidance algorithms of robots mainly have two aspects .One is global path planning and obstacle avoidance, and the other is local path planning and obstacle avoidance. Global path planning can be explained as the mobile robot constructs the surrounding environment in a static environment and seeks the optimal path problem between two target points. The specific algorithms [1] are: artificial potential field method, fence method, A* algorithm, View method, free space method, topological method, etc., local path planning [2] obstacle avoidance can be interpreted as local unknown in the environment of the robot or new obstacles appear, real-time path planning and obstacle avoidance based on the returned data, the specific algorithm is genetic Algorithms, fuzzy logic algorithms, neural network algorithms, etc. Since the security inspection robot has to inspect all important corners of the entire environment, the above methods cannot fully meet our needs.

In the actual complex environment, the security inspection robot in this paper needs to complete three steps: the first step is to construct the two-position map of the inspection environment, the second is the robot's own positioning, and the third is the path planning. The first two steps are specifically: in the unknown environment, using the SLAM algorithm to generate the data of the inspection target through the data obtained by the laser radar triangulation technology and the odometer data under the ROS operating system, and determine the robot and the solid obstacle in the environment. s position. The third step of the path planning goal is to calibrate the main target points of the inspection and the inspection sequence on the map, and to implement static path planning and obstacle avoidance through improved ant colony algorithm between each target point,

Copyright © 2019, the Authors. Published by Atlantis Press.
This is an open access article under the CC BY-NC license (http://creativecommons.org/licenses/by-nc/4.0/).

and then in the complex dynamic environment. The improved BUG algorithm is used for dynamic obstacle avoidance, and a collision-free motion path is quickly searched in the working space where obstacles exist, so that the robot can safely and collision-freely bypass all obstacles to reach the target point, thereby achieving inspection. The inspection task of the robot.

2. Positioning and Two-Position Map Construction of Security Inspection Robot

The Synchronization and Map Construction [3] (SLAM) problem can be described as: the mobile robot gradually builds a map of the surrounding environment while moving in an unknown environment, and uses this map to estimate the position and posture of the robot. This paper relies on laser radar to realize the synchronous positioning and composition of the robot.

The position and motion track of the security inspection robot can be expressed as:

$$a_1 = (x_1, y_1, \theta_1); \tag{1}$$

$$A = \{a_1, a_2, a_3, a_4 \dots, a_t\}; \tag{2}$$

Among them, t is time, a_t is the pose of the robot, x_t, y_t is the two-dimensional plane coordinate of the inspection robot, θ is the deflection angle of the robot, and A is the motion trajectory of the robot.

The relative motion of the security inspection robot can be expressed as:

$$B_t = \{b_1, b_2, b_3, b_4 \dots, b_5\} \tag{3}$$

Among them, $b_t$is the motion odometer obtained by the robot from time t to time t+1 in the control variable of the robot wheels.

The security inspection robot uses the RPLIDAR radar interface data to process the data to generate a real-time map. The particle filter Fast-SLAM algorithm with the least performance loss is used to simplify the algorithm into distance calculation and map update. At different times, the Fast-SLAM algorithm can maintain K particles. The specific form of the particles is:

$$A_t^{[K]},\ \xi_{t,1}^{[K]}, \dots, \xi_{t,N}^{[K]}, \Sigma_{t,1}^{[K]}, \dots, \Sigma_{t,N}^{[K]} \tag{4}$$

Where [K] is the number of the particle, $A_t^{[K]}$ is the sample trajectory, $\xi_{t,1}^{[K]}$is the mean, $\Sigma_{t,N}^{[K]}$ is N two Dimensional Gaussian collection. K particles have K trajectory samples $A_t^{[K]}$ and KN Gaussian distributions. Gaussian distribution models the road signs of particles. When the particle collection capacity tends to infinity, the particle filter can approximate the true posterior probability. Distribution, whose accuracy is close to the optimal estimate, in the update step, the scanned line is added to the map, and when the obstacle occurs, the set of adjustment points is drawn around the obstacle point instead of a single point.

Fig. 1 Laser processing process

Each node of ROS implements [4] its own separate functions and its calculation process is also performed by nodes. The communication between them is transmitted through messages, and the messages are transmitted through themes. The topics are published and subscribed. Function, the above is the entire calculation process. The laser radar scan obtains the original data and the odometer of the robot and the data is processed by the improved Fast-SLAM algorithm under the ROS system, thereby directly constructing the two-dimensional map. After the modeling and drawing of the robot's inspection environment is completed, the key points and directions that the robot needs to complete the inspection will be determined for the completed icon, and the robot will perform the inspection task according to the calibration points, as shown in Fig. 2 below.

Fig. 2 Map drawn under the ROS system

3. Path Planning for Security Inspection Robot

3.1 Path Planning Obstacle Avoidance Problem Description

The path planning of the security inspection robot [5] can be roughly divided into three steps: the first step is to calibrate the key environmental locations (which can be regarded as a point) that must be inspected in order in the environment in which the robot constructs the required inspection. The inspection order A→B→C→D→...→N. In the second step, we first use the grid model to establish the obstacle model of the global static environment of the robot. In the environment of calibrating the inspection sequence A→B→C→D→...→N, we use the improved ant colony algorithm to find one without considering it. The collision-free optimal or sub-optimal path in the case of dynamic obstacles, that is, the optimal between A→B, the optimal between B→C, and so on, and finally find the global optimal path in static environment planning. In the third step, it is parallel with the second step. When the robot walks according to the global path planning, there will be local unknown dynamic obstacles. At this time, we use the improved BUG algorithm to avoid obstacles and avoid obstacles in time. Specifically, it can be represented by the formula (5) and the formula (6).

$$\mathrm{min} f(x) x \in R^n \tag{5}$$

$$\mathrm{s.t.}\, h_i(x) \leq A_i \quad i = 1,2,3,\ldots,n \tag{6}$$

Among them: f(x) is the objective function;

$h_i(x)$ is a nonlinear constraint.

n is the number of inequalities.

The flow chart of the inspection route of the security inspection robot is as follows:

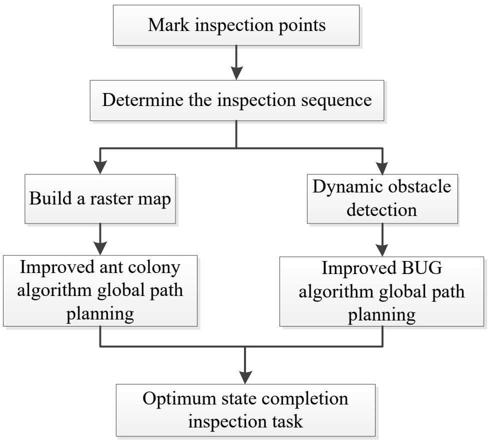

Fig. 3 patrol path planning flow chart

This paper establishes a 25*25 two-dimensional grid space[6], which is mainly used to find an optimal or sub-optimal path from node A to node B with the improved ant colony algorithm. The two-dimensional map is planned in the whole environment. Static obstacles, the robot two-dimensional space model is shown in Fig. 4.

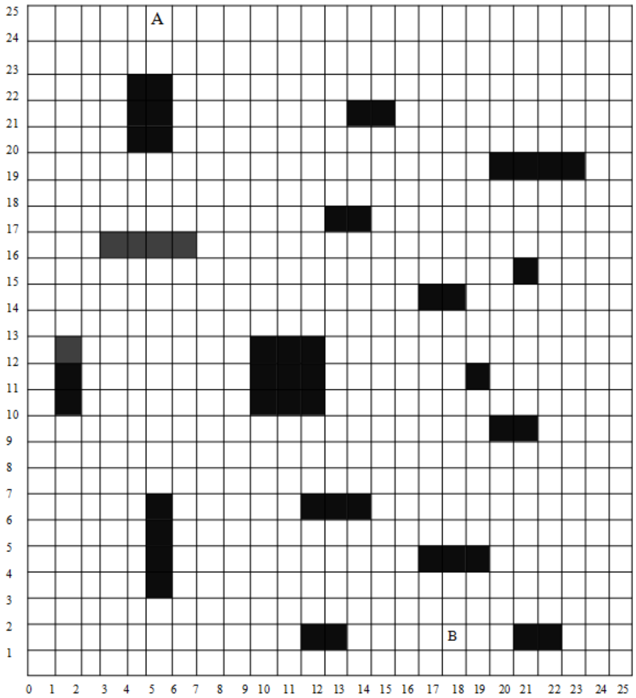

Fig. 4 2D workspace of the robot

3.2 Static Global Path Planning Based on Improved ant Colony Algorithm

Initially, Marco Dorigo, a European scholar, got the ant colony algorithm based on the ant's behavior of walking out of the optimal path in the process of searching for food [7] . When they find food, they release a secretion in the environment, called pheromones, which attracts other ants to feed. The concentration of pheromone in the foraging environment represents the distance from the food source (target). When other ants are looking for a target source, they will take different paths, there will be better paths, they will give feedback, they will attract more ants to take a more optimal path, and so on, eventually there will be an optimal path.

The ant colony algorithm is an extension of the foundation. On the basis of not considering other factors, we improve the ant colony system mentioned above, and improve its selection method, pheromone and optimal solution. We need to optimize the inspection route A→B→C→D→...→N of the security inspection robot. First, we carry out static global path planning for the path between A→B. The specific way is:

First, the pheromone on the path of the global optimal solution is increased. The specific formula is as follows:

$$P_{ij}(t+1) = (1-\xi).P_{ij}(t) + \xi\sum_{k=1}^{n} 1/L_{AB} \tag{7}$$

Among them, P_{ij} represents the value of the pheromone at time t;

ξ represents the pheromone volatilization coefficient, the range is $0 < \xi < 1$;

n represents the number of ants, k = 1, 2, 3, ..., n.

L_{AB} represents the length of the global optimal path.

Secondly, we must enhance its anti-interference ability. Under the action of the feedback mechanism, the ants walk according to the path of pheromone. When it moves to the next node, we will weaken the pheromone of the path, in this way to change the local changes of the pheromone, which can eliminate most of the paths that have been taken, and enhance the anti-interference ability. The specific formula is as follows:

$$P_{ij} = (1-\rho)\ \partial_{ij}+\rho P_0 \tag{8}$$

Among them,∂_{ij} indicates the concentration of the pheromone;

P0 is the initial concentration of the pheromone;

ρ is the residual coefficient of the pheromone, $0 < \rho < 1$.

Finally, after the anti-interference ability of the ant colony algorithm is enhanced, the algorithm is still not suitable for processing large amounts of computational data, so we need to improve it on this basis, because the pheromone information on the better path is continuously updated. Therefore, we perform a certain sorting weighting process on the intensity of each ant releasing pheromone to reduce the time for optimal path selection. The weighted formula is as follows:

$$P_{ij}(t+1) = (1-\xi).\partial_{ij}(t) + \sum_{\varepsilon=1}^{\beta}(\beta-\varepsilon).\sum_{k=1}^{n} 1/L_{AB} + \beta.\sum_{k=1}^{n} 1/L_{AB} \quad (9)$$

Among them,β represents the number of pheromones, $\varepsilon = 1,2,3，\ldots\beta$。

The final path we get is sorted as:

$$L_{AB1} > L_{AB2} > L_{AB3} > \cdots > L_{AB\beta} \quad (10)$$

Select the static global optimal path between A and B we need to complete our planning.

Similarly, the optimal path of B→C, C→D, D→E..., the final global optimal path is available:

$$L = L_{AB} + L_{BC} + L_{CD} + \ldots\ldots + L_{.N} \quad (11)$$

3.3 Dynamic Path Planning of Robot Based On BUG Algorithm

The BUG algorithm is a classical method for the dynamic obstacle avoidance of robots in an unknown environment [8]. The derivation process and theoretical analysis are relatively simple compared to other algorithms. The behavior is mainly divided into two states: “going along the main line” and “walking along the wall”. As shown in Fig. 5, the connection between the starting point A and the ending point B is the main line of the algorithm. The starting state is that the robot walks according to the main line. When the robot detects an obstacle, it starts to open the "walking along the wall" state. Until the robot moves to the main line, the "walk along the main line" state is turned on again, and so on, until the robot completes the inspection to reach the target point B. The BUG algorithm mainly includes algorithms such as BUG1, BUG2, Alg1, Alg2, Tangent Bug, and DistBug. The algorithm can effectively find the effective path we need and perform dynamic obstacle avoidance, but there may be local redundancy of the path during this period, especially in the case of irregular obstacles.

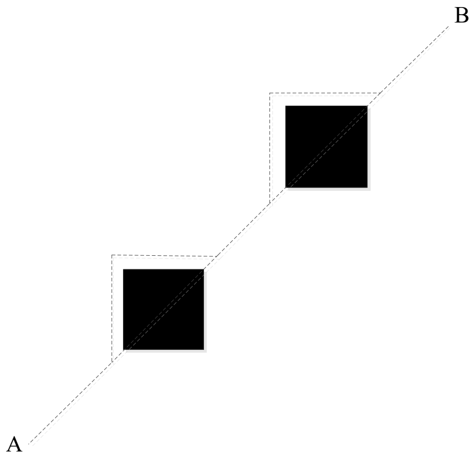

Fig. 5 Schematic diagram of path planning obtained by the traditional BUG algorithm

The classic BUG algorithm will not change once the direction of the main line and the detour is determined, so that the walking route will form a certain redundancy, which is not optimal. In this paper, combined with the characteristics of the security inspection robot itself, the "main line" and "bypass line" problems are improved, and the Dynamic main-Bug algorithm is proposed. The specific performance is as follows: the robot first follows the BUG algorithm. The main line running from the starting point to the end point generated together with the static optimal path, and calculating the slope k and the angle θ between the two points, and performing the "walking along the wall" mode when encountering an obstacle, reading in real time during the running process Take its own position angle, when the deviation rate k_1 of its own position is equal to K, exit the mode of “walking along the wall”, take the position of the point as a new starting point, re-plan the new main line, and so on, continuously Change the dynamic main line and continue to drive along the new main line until the end point.

In a complex environment, there may be more than one unknown dynamic irregular obstacle, so there will be new and new mainline replacements. This article only introduces one and two unknown irregular obstacles. In the case of an irregular obstacle, as shown in Fig. 8, the starting point of the robot inspection is $M(X_M, Y_M, \theta_M)$, the end point is $N(X_N, Y_N, \theta_N)$, and the starting main line is MN. When the robot moves from the M point along the main line MN to the F point, it starts to “walk along the wall”. When the robot reaches the Q point, it monitors its own deviation rate $k_1=\tan\theta_1=K_M=\tan\theta_M$, At this point, the robot exits the “walking along the wall” mode, starts the new main line QM planning with the Q point as the starting point, and finally moves along the main line to the end point N, completing the inspection task of the point M to N.

Among them, X_M, Y_M are two dimensional coordinate values.

θ_M is the angle of the robot.

K_M is the mainline slope.

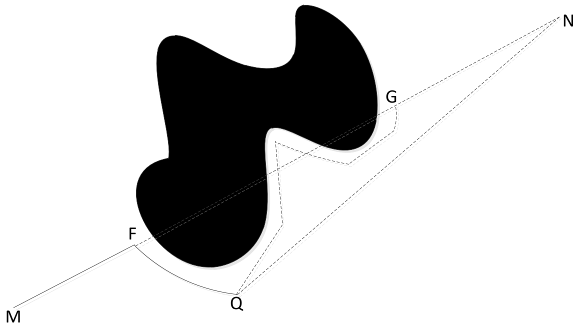

Fig. 6 Schematic diagram of the robot once obstacle BUG algorithm

It can be seen from Fig. 6 that in ΔQGN, it is obvious that QG+GN>QN, and the length of the curve between QN is larger than the linear length of QN. Therefore, the improved BUG algorithm greatly shortens the main line length of the traditional BUG algorithm. Improve the inspection efficiency of the robot.

Similarly, as shown in Fig. 7, the robot encounters two obstacles. The schematic shows that the starting main line is AB. When the inspection robot moves to point C, it starts to “walk along the wall” and reaches D point. The deviation rate of this point is the same as the slope of the main line. Exit the “walking along the wall” mode and start the inspection according to the main line 2. After reaching the I point, the deviation rate is the same as the slope of the main line 2. According to the main line 3 inspection, the target is finally reached. Point B completed the task of inspection. And the path is optimized, and the path is optimized.

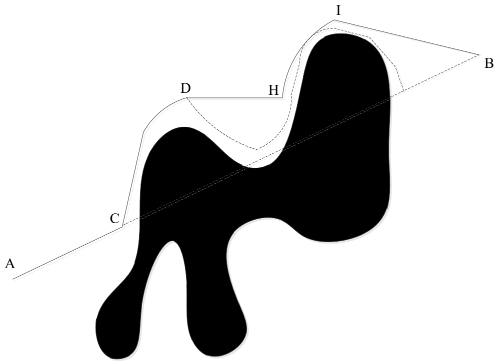

Fig. 7 Schematic diagram of the robot's two obstacles BUG algorithm

The dynamic main-Bug algorithm is compared with the traditional algorithm as follows:

Table 1 Comparative analysis of various BUG algorithms

BUG algorithm	Algorithm characteristics
BUG2	A parallel main line from the start point to the end point is introduced.
Alg2	After the direction of detour is determined, the route cannot be changed, and the route is redundant.
DistBug	The cumulative parameter Dir is introduced to judge the motion direction of the robot
Parallel obstacle boundary walking BUG	Reduced redundancy in path planning, but obstacles can only be rectangular
Dynamic main-Bug	Multi-mainline replacement, considering various shape obstacle path planning problems, solving the problem of route redundancy

4. Simulation Analysis

In order to verify whether the above method is effective, the MATLAB platform is used for simulation experiments [9]. Firstly, the improved ant colony algorithm is simulated in the established grid environment. The initial coordinates are set to A (5.6, 24.5) and the end point coordinates are B (17.2, 1.6), the population size and population are 30 and 80 respectively, the pheromone calculation parameters and heuristic parameters are set to 3 and 1.8, and the number of cycles is 800. Fig. 8 shows the initial path planning of the robot, and Fig. 9 shows the improvement. Path planning, it can be clearly seen that both paths can achieve the effect of inspection, but the improved path is significantly more optimized than the initial path.

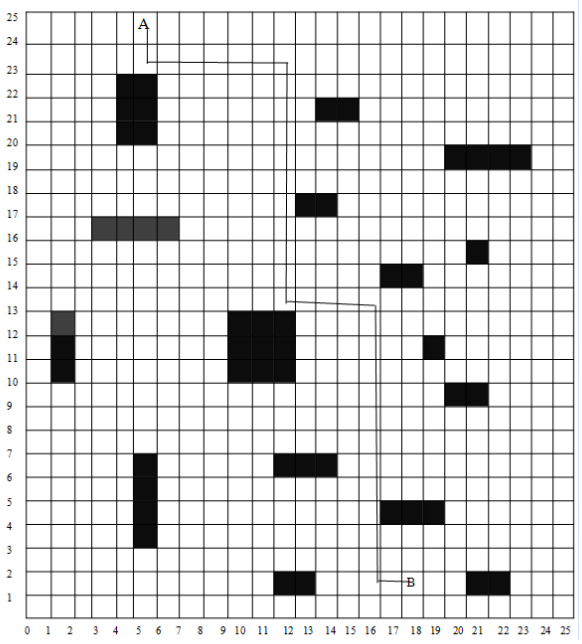

Fig. 8 Robot initial path planning

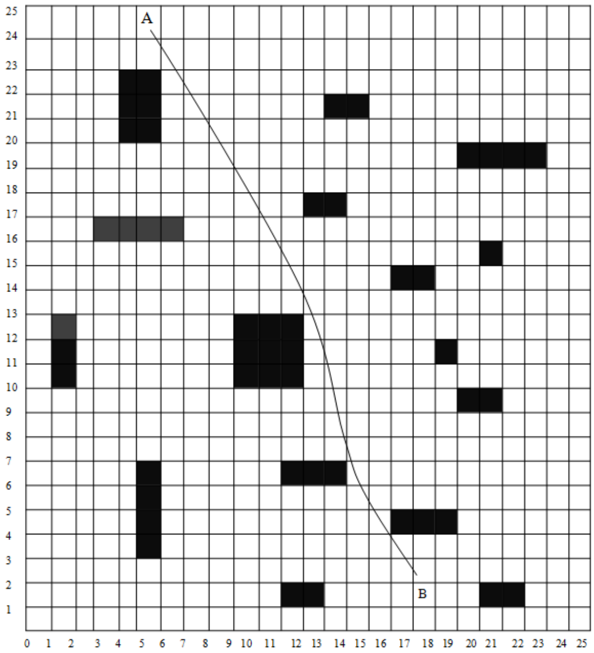

Fig. 9 Static path planning of robot improved ant colony algorithm

Next is the simulation of the Dynamic main-Bug algorithm. We set up two simple block obstacles [10]. The simulation results are shown in Fig. 12. The solid line is the actual motion trajectory of the robot, and the dotted line is the main line of the traditional BUG algorithm. Through simulation, we can clearly see that the improved BUG algorithm reduces a lot of redundancy compared with the traditional BUG algorithm, and it can be clearly seen from Table 2 that the more obstacles, the improved BUG algorithm for the robot The efficiency of inspections is getting higher and higher.

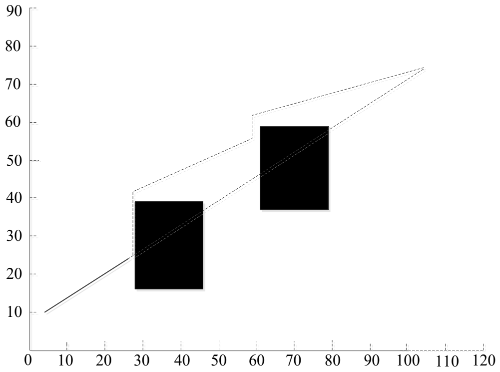

Fig. 10 Schematic diagram of improved BUG algorithm simulation

Table 2 Comparison of BUG algorithm paths

Number of obstacles	Traditional BUG algorithm	Dynamic main-Bug algorithm
1	36.789	33.694
2	41.262	34.567
3	48.374	36.231
4	55.035	37.256

5. Conclusion

In the operating environment of the security inspection robot, this paper proposes the mapping of the data and odometer data obtained by the laser radar triangulation technology using the SLAM algorithm under the ROS operating system, and the static target path through the improved ant colony algorithm. Planning and avoiding obstacles, then using the improved BUG algorithm for dynamic obstacle avoidance and path planning in complex dynamic environment. Through simulation experim ents, these algorithms can effectively and efficiently achieve the inspection tasks of security inspecti on robots.

References

[1]. Zhang yongni. Research on obstacle avoidance path planning algorithm of intelligent robot [M]. Chongqing. Information technology, 2017

[2]. Wang shuai. Path planning method for mobile robot in dynamic uncertain environment [J]. Electrical technology, 2010, (1): 18-21.

[3]. Sun yuliang. Mobile robot indoor instant map construction and autonomous navigation [thesis of master of engineering of dalian university of technology]. Dalian university of technology, 2011

[4]. Chen Zhuo, Su Weihua, Comfort Ning, Qin Xiaoli. Realization of SLAM and Path Planning of Mobile Robot in the Framework of ROS [J].monograph, 2017.02.109.

[5]. Yang xing. Research on indoor autonomous navigation mobile robot path planning [D]. Zhongbei University, 2016.

[6]. Wang xingce, zhang rubo, gu guochang. Global path planning of robots based on potential field grid method [J]. Journal of Harbin engineering university, 2003, 24 (2): 170-174.

[7]. Liu jianhua, Yang jianguo, liu hua ping. Global path planning method for mobile robots based on potential field ant colony algorithm. Journal of agricultural machinery. 2015, 46 (9):18-27.

[8]. Du wen. Research and implementation of real-time path planning algorithm in dynamic environment [master of engineering thesis of university of electronic science and technology]. Chengdu: university of electronic science and technology, 2011

[9]. Xie yuanyuan, zhu qingbao. Robot path planning based on ant colony algorithm in dynamic environment [J]. Journal of nanjing normal university (engineering technology edition), 2006, 6(3): 45-50.

[10]. Chen shaobin. Research on path planning and trajectory tracking of autonomous mobile robots [D]. Hangzhou: zhejiang university, 2008.