Research on dynamic path planning method of moving single target based on visual AGV

Meng Sun¹ · Liqun Lu¹ · Huiting Ni¹ · Yi Wang¹ · Janghui Gao¹

Received: 29 June 2021 / Accepted: 13 January 2022

Published online: 22 February 2022

© The Author(s) 2022 **OPEN**

Abstract

In order to research the ability of AGV to cope with the dynamic path planning in mobile single-object dynamic environment, the dynamic path planning methods based on reduced point and non-reduced point were proposed. This paper focuses on the dynamic path planning method of non-reduced point moving targets, adopts the method of combining historical data and real-time data, establishing the NAR neural network prediction model to predict velocity of moving target, and connecting with the A* algorithm, through the example simulation experiment, the results show that the effective success rate of the algorithm reaches 77.5% in the map environment set in this paper.

Keywords Visual AGV · Dynamic path planning · MATLAB · Simulation experiment · Complex environment

1 Introduction

AGV (Automated Guided Vehicle) has the advantages of high efficiency, stable operation and long working time. It is widely used in the field of warehousing and logistics and pipelined production line, but it can only be competent for simple static path navigation planning tasks or line patrol tasks, and its application field is single. Therefore, improving the ability of AGV to deal with dynamic path navigation in complex environment is of great significance for its development. At present, path planning technology is mainly divided into two categories: Global Path Planning based on known map information and Dynamic Path Planning based on unknown map environment information. Global Path Planning, namely Static Path Planning, is a navigation path planning based on mastering the overall map information. Common planning algorithms include Dijkstra algorithm, A* algorithm [1, 2], Ant colony algorithm [3–5], Simulated annealing algorithm [6, 7], Artificial

potential field algorithm [8–11], Particle swarm optimization method [12, 13], Genetic algorithm [14, 15]; Dynamic Path Planning is also called Local Path Planning in a narrow sense, compared with Global Path Planning, it pays attention to the obstacle information and its own position information of mobile robot in real-time state. Mobile robots use sensors to read the surrounding environment information in real time. When obstacles appear, they complete obstacle avoidance through local path planning algorithm. Common Local Path Planning algorithms include Artificial Potential Field method, Vector Field of Histogram (VFH) method and DWA (Dynamic Window Approach) method. The Artificial Potential Field method simulates the environment map as a gravitational field model, during the operation of the mobile robot, it is always attracted by the target point, and the obstacles in the path environment are always accompanied by the repulsion force. Finally, the resultant force between gravity and repulsion is calculated to guide the robot to move, which has the advantages of

✉ Liqun Lu, luliqun@sdu.edu.cn; Meng Sun, sunmeng0121@gmail.com; Huiting Ni, 837089704@qq.com; Yi Wang, 1586664861@qq.com; Janghui Gao, 1546396991@qq.com | ¹School of Transportation and Vehicle Engineering, Shandong University of Technology, No.266, West Xincun Road, Zhangdian District, Zibo City, Shandong Province, China.

easy realization and good control effect, but it is easy to produce optimal layout and lock. The core idea of VFH method [16, 17] is to use histogram to represent obstacles and constantly update the information of obstacles through the sampling data obtained by the upper sensor, so as to drive the robot to complete obstacle avoidance path planning. This method has high flexibility and strong adaptability to the environment, but it has high requirements for the ability of the host computer to process environmental information data. The full English name of DWA is Dynamic Window Approach. This method samples multiple groups of velocity samples in the velocity (v, ω) space, simulates the trajectory of the mobile robot in the next time domain, comprehensively evaluates the simulated trajectory, and selects the speed of the optimal trajectory as the driving speed of the mobile robot, which makes DWA Local Path Planning algorithm have efficient and stable obstacle avoidance function [18, 19]. In recent years, with the development of communication technology, dynamic path planning has developed from local planning to global planning. At present, the typical research on Global Dynamic Path Planning is as follows: Hu Bin's Dynamic Path Planning method based on time window [20]; Ge Yan and Sun Haipeng, respectively, proposed the construction method of traffic network model based on graph theory and the method based on real-time traffic [21, 22]; Wang Rong's method of applying Beidou Positioning system to robot dynamic path planning [23]; Liu Jingkun proposed a method for mobile robot to update the map according to obstacles in motion and complete dynamic path planning using A* algorithm [24].

The academic research results on static path planning are quite rich, and the research on dynamic path planning focuses on local path planning in unknown environment (obstacle avoidance and local optimization prevention). Taking visual AGV as the research object, this paper focuses on the in-depth research on the path planning method in the mobile single target dynamic environment, and puts forward the dynamic path planning method based on point reduction and nonpoint reduction, in order to get the best path, make AGV accurately complete the path guidance task for uncertain targets, improve the path planning ability of AGV in complex environment, and expand the application field of AGV.

2 Positioning principle and map selection of visual AGV

Localization and map are the basis of path planning. AGV located in local environment is implemented based on SLAM, which means Simultaneous Localization and map construction. Under the condition of no prior information

of the environment, AGV can determine its position in the environment and estimate its motion position through the mounted sensor. In the global situation, its position can be determined under the environment map by obtaining external steady-state information. The localization model of SLAM and the characteristics of various maps lay a foundation for the research of AGV path planning.

2.1 Basic structure and mathematical expression of visual SLAM

Visual SLAM realizes the construction of environment map in the process of movement with the help of camera, and the specific structure framework is shown in Fig. 1, including: reading sensor information (using sensors to read image information and preprocess), visual odometer (calculating the camera's movement and pose and obtaining the appearance of local map through the relationship between adjacent images), and image processing non-linear optimization (processing the camera pose information and loop detection information accumulated by the visual odometer at different times, and optimizing these information to obtain the motion track), loop detection (judging whether it has been to the previous position), mapping (building the environment map by estimating the camera motion trajectory). The main principle is to use the difference between adjacent image frames to evaluate the camera's own motion, and then obtain the local positioning information of AGV.

2.2 Camera mathematical model and pose representation

RGBD camera can obtain the depth information of the collected image when collecting the surrounding environment data, complete the matching of RGB image pixel and image depth, and input the corresponding RGB image and depth map. RGBD cameras are divided into two categories according to their principles: measuring pixel distance based on structured light principle, the typical products are Microsoft Kinect1.0 and Aubi Astra; measuring pixel

Fig. 1 Classic visual slam framework

distance based on time-of-flight principle, the typical products is Microsoft's Kinect 2.0 camera.

RGBD cameras based on structured light and ToF usually include a normal camera, a transmitter and a receiver. When the RGBD camera works, the emitter usually emits infrared light to the target position detected. For the structured light RGBD camera, the distance between its position and the target position will be calculated according to the received structural light pattern information. For the ToF type RGBD camera, the pixel depth information of the whole image is obtained by comparing the time difference between the pulse light emitter and the pulse light receiver, namely the time difference of infrared light flight, as shown in Figs. 2 and 3.

2.2.1 Mathematical models of cameras

Either can be as a pinhole camera model, its principle as shown in Fig. 4. First, building the world coordinate system (XYZ), camera coordinate system (XYZ) and image coordinate system (UV). Set the coordinates of point P to be (u, v) in the image coordinate system and (x, y, z) in the camera coordinate system, according to the principle of similarity, the relationship between them is [25]:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{z} \cdot \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (1)$$

Fig. 2 Structured light rgbd camera

Fig. 3 TOF type rgbd camera

where: f_x and f_y are the focal length of the camera, c_x and c_y are the coordinates of the optical center of the camera in the image coordinate system.

2.2.2 Pose representation of the camera

The pose of the camera represents the process of converting the position of the camera coordinate system to the world coordinate system. Description of camera position and posture by constructing a position and posture transformation matrix based on six degrees of freedom (three degrees of freedom of rotation and three degrees of freedom of translation) in the camera position and posture with a visual odometer. Due to the constraint of pose transformation matrix, lie algebra is introduced to express pose of camera. Let the camera coordinate system be C and the world coordinate system be O , and the transformation relationship between them is as follows:

$$T_C^O = \begin{bmatrix} r_{00} & r_{01} & r_{02} & t_0 \\ r_{10} & r_{11} & r_{12} & t_1 \\ r_{20} & r_{21} & r_{22} & t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R_C^O & t_C^O \\ 0 & 1 \end{bmatrix} \quad (2)$$

Fig. 4 Schematic diagram of pinhole camera

where: $T_C^O \in SE(3)$, $R_C^O \in SO(3)$ is rotation matrix, $t_C^O \in R^3$ is translation matrix.

Using the transformation matrix of camera pose, the position relationship between the point p_C in the camera image coordinate system and the point p_O in the world coordinate system is as follows:

$$p_O = T_C^O p_C \quad (3)$$

When the lie algebra $\mathfrak{se}(3)$ corresponding to lie group $SE(3)$ is introduced into the nonlinear optimization of pose estimation, there exists $\zeta \in \mathfrak{se}(3)$, which is a six dimensional vector consisting of a translation vector and a rotation vector. For Lie algebra $\mathfrak{se}(3)$, the exponential mapping is lie group $SE(3)$:

$$\exp(\zeta^\wedge) = \begin{bmatrix} \exp(w^\wedge) J_V & \\ 0^T & 1 \end{bmatrix} \in SE(3) \quad (4)$$

$$\text{where : } J = \sum_{n=0}^{+\infty} \frac{1}{(n+1)!} (w^\wedge)^n$$

In the optimization pose solution, a function related to the camera position and posture is also constructed and used to solve the derivative of the position and posture transformation to adjust the current estimate. Generally, using the disturbance model to solve the derivative of camera pose. Assuming function $F = TP$ (where T is the pose transformation matrix, lie algebra is ζ , and map coordinate is P), a disturbance $\Delta T = \exp(\delta\hat{\zeta})$ is applied on T ($\delta\hat{\zeta}$ is the disturbance quantity of Lie algebra P) to obtain;

$$\frac{\partial F}{\partial \delta\zeta} = \frac{\partial(T \cdot P)}{\partial \delta\zeta} = \lim_{\delta\zeta \rightarrow 0} \frac{\begin{bmatrix} \delta w^\wedge (R \cdot P + t) + \delta v \\ 0 \end{bmatrix}}{\delta\zeta} = \begin{bmatrix} I & -(R \cdot P + t)^\wedge \\ 0^T & 0^T \end{bmatrix} \quad (5)$$

Through the above formula, we can know the advantages of Lie algebra in nonlinear optimization, which lays a foundation for the accurate solution of camera pose estimation.

2.3 Map selection

2.3.1 Global location and local location analysis

Although positioning information of AGV can be obtained in local environment based on camera, there are two disadvantages of obtaining positioning information by using visual odometer in long distance and large range environment. One is that the redundant information of odometer increases, the calculation amount increases, and the accuracy of pose solution decreases gradually. The other is to cooperate with loopback detection, which requires

a lot of computing resources. In addition, if the working path is not loopback mode, loopback detection cannot be completed to eliminate errors. Therefore, the best way to eliminate positioning errors is to integrate steady-state position information, such as GPS, QR code and UWB.

For AGV, global positioning and local positioning are very different in principle and complement each other. Local positioning focuses on avoiding obstacles in unknown environment; however, global positioning tends to control its own positioning information under the overall environment map in order to evaluate the path and measure the movement step. When AGV is in an uncertain and complex environment, due to the absence of prior map, local localization plays a role in guiding AGV to avoid obstacles, while global localization evaluates the relationship between its own position and target position for AGV.

However, when AGV is in a state known to static or prior maps and has the ability of global positioning information feedback, due to the limitations of camera pose evaluation (large amount of calculation and low accuracy), its local positioning effect is not obvious, while the global positioning based on steady-state information can play a leading role.

2.3.2 Analysis of various map models

The acquisition of location information solves the problem of evaluating its location in local or global environment in path planning, and the selection of appropriate planning map is the key step to realize path planning. The current mainstream map models include point cloud map, octree map, topology map and raster map.

(1) Point cloud map

Point cloud map is a point cloud map which uses RGBD camera to convert the data into point cloud, and then stitches it together. The representation of the map is a three-dimensional environment space point cloud map. Due to device limitations, in order to reflect the construction effect of point cloud map, this paper completes the construction of dense point cloud based on TUM dataset, as shown in Fig. 5.

Point cloud map is not suitable for navigation path planning because it often takes up a lot of storage space and contains some unnecessary information, which reduces the resolution. To solve this problem, point cloud map is usually converted into octree map. As shown in Fig. 6, octree itself is a 3 D grid map, its main idea is to use three-dimensional space as the root node of the tree, divide each parent node into eight child nodes, and then divide regularly until a resolution-compliant node appears, octree map by integrating the parent node, simplify the 3 D grid

Fig. 5 Point cloud map

Fig. 6 Octree map

structure, can reduce the storage space and improve efficiency.

(2) The topology map

A topological map is essentially a statistical map, as shown in Fig. 7. It is a highly abstract map composed of points and lines to maintain correct relative positions. Topology map is a concise and efficient way to express maps among all maps. Due to the fast search speed of topology map, it play an important role in path planning from the macro point of view, but the description of local environment is poor, which cannot reflect the accurate local environment information.

(3) Raster map

The form of raster map is 2D occupied raster map, as shown in Fig. 8. The raster map mainly divides the environment into a series of grids, and each grid is assigned a possible value, which is used to represent the probability of the grid being occupied. In the representation of scale map, 2D map is processed in grid. For environmental features, the value 0 is used

Fig. 7 Topological map

Fig. 8 Grid map

to indicate that there are no obstacles and the path is unobstructed; the value 1 indicates that there is an obstacle and the path is blocked.

For AGV and other robots that move flat, much of the information in 3D maps is redundant. On the contrary, 2D raster map requires less memory, higher efficiency and faster map construction, which is extremely suitable for path planning research.

3 Research on dynamic path planning method of 2-mobile single objective

3.1 Pose representation of camera

The method framework based on reduction point is shown in Fig. 9. AGV and mobile target use communication tools

to agree the specific point of the task as the node for both sides to "meet". In order to avoid waiting too long, they constantly update the agreed node on the way until both sides have no objection. In addition, the path planning algorithms in motion adopt A* algorithm.

Based on the non-reductive method framework, as shown in Fig. 10, the database of NAR neural network prediction model is established by using the historical speed information set of the moving target (the average speed of the moving target when it completely passes the given path each time), and the predicted speed of the moving target is obtained and regarded as the preliminary basic speed to predict the initial "cutoff" point. Because of the high degree of freedom of the moving target, it can not move according to the predicted speed. Taking T as the real-time speed of the moving target, the NAR neural network is established to predict the next time speed, which is used as the reference speed to evaluate the initial best "cutoff" point and update the "cutoff" point according to the rules; finally, using A* algorithm to construct the path planning, and predicting and updating constantly the "cutoff" points in the movement until the distance between AGV and the moving target is less than the limited distance.

3.2 Basic principle of A* algorithm

A* algorithm uses heuristic path node search method, uses evaluation function prediction and combines the

Fig. 9 Framework of reduced point method

Fig. 10 Framework of non-reduced point method

advantages of BFS (best first search) algorithm and Dijkstra algorithm to select the least cost node as the next search node. The evaluation function consists of $h(n)$ and $g(n)$:

$$f(n) = h(n) + g(n) \quad (6)$$

where: $f(n)$ is the evaluation function from the starting point to the target point, $h(n)$ is the heuristic function, which is the shortest distance from the current search node n to the target node with the least cost, and $g(n)$ is the actual distance from the starting node to the current node.

The definition of function $h(n)$ is the precondition to get the shortest path. The closer the estimated value of heuristic function $h(n)$ is to the real value of target search node n , the higher the efficiency of node search. Generally, there are three definitions: based on Euclidean distance, based on Manhattan distance and based on Chebyshev distance, it is verified that the Manhattan distance definition can ensure the stability of optimal path acquisition. If the two-dimensional coordinate value of the current position in the grid map is (x_1, y_1) , and the two-dimensional coordinate value of the target search node is (x, y) , then the expression of $h(n)$ is:

$$h(n) = |x - x_1| + |y - y_1| \quad (7)$$

A * algorithm will set open list and close list in the search optimal path node. The specific steps are shown in Fig. 11:

1. Set the OPEN list and CLOSE list, put the starting node 0 into the Open list and grid the search area.
2. Add eight non CLOSE list nodes around the search node to the OPEN list, calculate the $f(n)$, $h(n)$ and $g(n)$ values for the nodes in the Open list, and select the node with the smallest $f(n)$ value as the current node.
3. Save the node that completes the search to the CLOSE list and delete it from the OPEN list.
4. Traverse the OPEN list and repeat the above steps to select the minimum search node of the evaluation function $f(n)$.
5. Until the target node is stored in the CLOSE list, the shortest path search is completed, and the final path is obtained through reverse order.

3.3 Simple logical verification of the point-like method

A logical inference-based approach is used to better illustrate the advantages of the dynamic route method based on the approximate points over the static route method based on the approximate points. Assume that the initial fixed point of AGV and the moving target is P_0 , the path planned by using A* algorithm is L_0 , and the time is t_0 (the running time of the path planned based on A*). Assume

that one party cannot reach the appointed point P_0 on time, then one party will wait for the other party. If the point is rearranged (approximately point P_1), the AGV uses A* algorithm to plan the path as L_1 and integrates the real-time information of the moving target and its own position to eliminate the waiting time difference. In this way, the appointed points are constantly refreshed, so that the AGV can efficiently complete the task.

3.4 Implementation of non-reductive point dynamic path planning method

The dynamic path method based on non-reductive point moving target point can be simply divided into: through the historical data set to build the NAR neural network to predict the average speed of moving targets, the sampling speed of moving targets in time through the NAR neural network to predict the future speed, combined with historical data to predict the average speed and based on the prediction of real-time speed build cutoff point selection and update the rules, and constantly refresh cutoff point position over time. When the distance between the AGV and the moving target is a safe distance, it will not refresh.

3.4.1 Realization of NAR neural network algorithm for predicting future speed

The neural network is composed of input layer, hidden layer, output delay layer and output layer. As shown in Fig. 12, left and right $y(t)$ represent input and output, respectively; 1: 5 is the order of delay; W is connection weight; b is the threshold value; the "10" in the hidden layer represents the number of nodes in the hidden layer [26–28], the specific mathematical expression is as follows:

$$y(t) = f(y(t-1), y(t-2), \dots, y(t-k)) \quad (8)$$

where: f is a nonlinear regression function; k is the order of delay.

Fig. 11 A* algorithm flowchart

Fig. 12 NAR neural network structure

Import dataset DATA (moving target historical average speed dataset and moving target sampling speed dataset in cycle T) into segmented data (for example, 70% are training data, 20% are validation data and 10% are test data) using MATLAB and NAR model, and the number of nodes in the hidden layer and the order of lag delay were constantly adjusted until the model error reached the requirement and the prediction was made.

3.4.2 Rules for initial selection and update of cutoff point

The dynamic path navigation planning of non-reducing point moving target is to introduce real-time distance information to ternary process the binary position coordinates on the premise of deterministic path. The specific steps are as follows: let the running path of the moving target be L , the starting position coordinates be (x_0, y_0) , and the ending position be (x_n, y_n) (if it is loop motion, the ending position is consistent with the starting position). The coordinates of moving object in real-time motion are (x, y) . In the world coordinate system, the mileage from the starting coordinates (x_0, y_0) to the real-time motion coordinates (x, y) is unique. In the world coordinate system, the mileage from the starting coordinate (x_0, y_0) to the real-time motion coordinate position (x, y) is unique. If the mileage is S , the real-time coordinate is (x, y, S) .

First selection rule of cutoff point: before the cutoff point is selected, NAR neural network is constructed by using the historical average speed set of the moving target to predict the average speed of the moving target. Let the average speed of moving target prediction be $\bar{v}$, the stable moving speed of AGV be v , and the cutoff point position be the dynamic point (x, y) , which is traversed from the starting point, and based on A* algorithm, the time taken for AGV to reach this point from the immediate position is t . At the same time, the predicted average speed $\bar{v}$ is the basic speed, and the time kt (t value is selected according to the actual situation, k is 1, 2... N) is the prediction time length to compare the time difference between AGV and moving target to the predicted position one by one. When $t_0 \leq nt$ and the distance difference is less than the limit value, the predicted position (x, y) is taken as the initial "cutoff" point. If not, the initial "cutoff" point can not be obtained, and the path planning task fails.

Update rule of "cutoff" point: after obtaining the initial "cutoff" point, AGV plans the path motion according to A* algorithm, takes time T as a cycle, reappears and evaluates the "cutoff" point, refreshes the "cutoff" point, and plans the path. After one cycle, AGV obtains the sampling speed set of the moving target, and constructs the NAR neural

network prediction model to predict the future motion speed v_n :

When $v_n \geq \bar{v}$ is selected, in order to prevent "escape", AGV selects the moving target speed v_n as the reference speed. The idea of "cutoff" point selection refers to the principle of initial selection, and the cutoff point position is dynamic point (x, y) , which is traversed and selected from the immediate position point.

When $v_n < \bar{v}$, in order to balance the return and risk, weight coefficients λ and δ are introduced, and the future expected speed of the next cycle is expressed as:

$$\hat{v} = \lambda v_n + \delta \bar{v} \quad (9)$$

$$\text{where: } \lambda = \frac{\bar{v}}{v_n + \bar{v}}, \delta = \frac{v_n}{v_n + \bar{v}}.$$

According to rules (1) and (2), AGV refreshes the "cutoff" points and uses A* algorithm to plan the dynamic global path until it reaches a safe distance from the moving target.

3.5 Simulation experiment verification

According to the above analysis, 2D map is more conducive to dynamic path planning, so 2D map is used in the experimental simulation process.

According to the above, the dynamic path method based on the historical statistics data and global map information of non-reduced-point is proposed, using simulation experiments to verify the effectiveness of the algorithm in MATLAB environment, and its positioning mode is the global positioning in the known prior map environment. Before the experiment, we need to build an environment map of 50 m*50 m with several accessible routes, as shown in Fig. 13. The moving target moves from the upper right along the boundary in a U-shape, and the initial position of the moving target is set as (48,48), the corresponding ternary coordinate is (48,48,0). The end-point coordinate is (2,48), and the corresponding ternary coordinate is (2,48,138). Set the AGV's position before responding to a task to be (25,25) with a stable speed of 2.5 m/s-2.55 m/s. In order to establish the prediction model of historical data, the historical average moving speed of the moving target was set between (2,3) m/s by MATLAB, and 30 groups of data values were obtained as the historical data set of the average speed of the indefinite moving target by example simulation.

Establish a NAR neural network prediction model and set the hidden layer to 9, threshold and weight were randomly generated. In order to prevent the prediction results

Fig. 13 Planar map

Fig. 14 Neural network training results

from being disturbed by abnormal data, the lag order was set as 3, 70% of the data were training data, 20% were test data and 10% were validation data. As shown in Fig. 14, when the neural network is trained to step 7, the fitting variance is minimum.

As shown in Fig. 15, after NAR neural network training, the error between the predicted value and the real value is small, basically meeting the accuracy requirements. As shown in Fig. 16, the average speed of a moving target passing through a given path is 2.4 m/s in the immediate state.

Fig. 15 Neural network fitting effect diagram

Fig. 16 NAR neural network average speed prediction chart

According to the NAR neural network prediction model, the average speed of the moving target passing the path is 2.4 m/s, and the stable speed of AGV is 2.5–2.55 m/s. The starting point of the moving target is (48,48,0), and the starting coordinate of AGV is (25,25). According to the initial cutoff point selection rules in the previous paper, the average speed is taken as the reference speed. T is set

as 0.1 s, take the limited distance difference of 0.5 m, and A* path planning algorithm is used to continuously select points to compare with the potential points of the moving target one by one. Finally, an initial "cutoff point" (48,25.92, 22.08) that meets the requirements is selected to complete the navigation path planning, as shown in Fig. 17.

For the convenience of simulation, the AGV refresh time is set as 3S (specific setting is set according to the environment map), and the speed of the moving target is set as: the first 3S is 2.5 m/s uniform motion; 3–8 s to do variable acceleration, acceleration of (0.08–0.12) m/s²; 8–12 s for uniform deceleration, the acceleration is set as (0.09–0.11) m/s²; and then we're going at a constant speed. Ten groups of velocity data were sampled within 3 s, and the NAR neural network was used to evaluate the moving target in real time. The mean value is used as the predicted speed when the velocity is uniform, and the predicted speed of other update cycles is shown in Figs. 18, 19 and 20. According to the cutoff update rule, complete the task of dynamic path planning for non-reducible moving targets, and the results are shown in Table 1.

To verify the success rate of the algorithm, the starting position of AGV was set as (25,25), (35,40), (18,35), (25,10), and the starting position of moving target was (48,48,0), and each group has been performed 20 simulation experiments. The moving target speed model established by Simulink is shown in Fig. 21, and the limited speed range is (2–3) m/s. The speed function of numerical generation changes of Sine Wave, Constant and Gain modules was modified in real time, the refresh time was set as 3 s, and the stable speed of AGV was set as 2.5–2.55 m/s. Finally, under the set map environment, the proposed method

Fig. 17 A* algorithm first path planning

Fig. 18 Predicted speed when $T=2$

succeeded 62 times, with a success rate of 77.5%, as shown in Table 2.

4 Conclusion

1. Based on visual SLAM technology, localization in local environment can be realized according to the conversion relationship between camera pose and AGV pose by using the calibrated camera.

Fig. 19 Predicted speed when $T=3$

Fig. 20 Predicted speed when $T=4$

Table 1 Example simulation results

Update times	AGV coordinates	Moving target coordinates	Prediction speed
0	(25,25)	(48,48,0)	2.4
1	(32.5,25)	(48,40.5,7.5)	2.5
2	(40,25)	(48,32.54,15.46)	2.82
3	(41.3,17.62)	(48,23.79,24.21)	2.93
4	(46.2,11.95)	(48,15.53,32.77)	2.61

Fig. 21 Simulink model of moving target velocity function

Table 2 Simulation success rate of each calculation example

Different starting points	(25,25)	(35,40)	(18,35)	(25,10)
Times of success	16	13	17	16

2. According to the analysis of all kinds of maps, it is concluded that 2D maps need less memory, higher efficiency and faster speed of map construction, which is very suitable for the study of path planning.

3. Compared with static path planning method, the rationality of A* algorithm is verified by logical reasoning.
4. For non-reducible moving targets, based on the combination of historical data and real-time data, NAR neural network is established and A* algorithm is used to refresh the "cutoff point" to plan dynamic path. The success rate of the algorithm is 77.5%.

Funding This journal was supported by the Korean Federation of Science and Technology Societies Grant.

Declarations

Conflict of interest The author(s) declare that they have no competing interests

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Leach AR, Lemon AP (1998) Exploring the conformational space of protein side chains using dead-end elimination and the A* algorithm[J]. *Proteins* 33(2):227–239
2. Goldberg AV, Harrelson C (2005) Computing the shortest path: A search meets graph theory[C]. In: *Symposium on Discrete Algorithms*, pp 156–165
3. Qiang L, Fuzhou D (2018) Research on path planning and motion control of omnidirectional robot based on ant colony algorithm. *Mech Des Manuf* S2:157–159
4. Changan L, Xiaohu Y, Chunyang L, Hua W (2011) Mobile robot dynamic path planning method based on improved ant colony algorithm. *Acta Electron Sin* 39(05):1220–1224
5. Ping FX, Xiong L, Sheng Y, et al (2003) Optimal path planning for mobile robots based on intensified ant colony optimization algorithm. In: *2003 IEEE International conference on robotics, intelligent systems and signal processing, proceedings. IEEE*
6. Janabisharifi F, Vinke D (1993) Integration of the artificial potential field approach with simulated annealing for robot path planning. In: *International symposium on intelligent control*, pp 536–541
7. Ming Z, Shudong S, Yanwu P (1998) Robot path planning based on genetic simulated annealing algorithm. *J Aeronaut* 01:119–121

8. Jianying Z, Zhiping Z, Huang L (2006) Robot path planning based on artificial potential field method. *J Harbin Inst Technol* 08:1306–1309
9. Zhenzhong Yu, Jihong Y, Jie Z, Zhifeng C, Yanhe Z (2011) Path planning of mobile robot based on improved artificial potential field method. *J Harbin Inst Technol* 43(01):50–55
10. Bo CY, Chen LG, Song MY et al (2014) UAV path planning using artificial potential field method updated by optimal control theory. *Int J Syst Sci* 47(6):1–14
11. Igarashi H, Kakikura M (2004) Path and posture planning for walking robots by artificial potential field method. *IEEE international conference on robotics and automation*. IEEE
12. Li Qing Xu, Yinmei ZD, Yixin Y (2010) Global path planning strategy for mobile robot based on particle swarm optimization. *J Beijing Univ Sci Technol* 32(03):397–402
13. Bin Wu (2008) Research and application of particle swarm optimization algorithm for vehicle routing problem. *Zhejiang University of Technology*, Hangzhou
14. Dong SS, Mao L (2000) Coordinated path planning for multiple mobile robots based on genetic algorithm. *J Autom* 05:672–676
15. Dong LG, Bin XH, Guang LC (2003) Path planning method of mobile robot based on Genetic Algorithm in dynamic environment. *Robot* 04:327–330, 343
16. Kumar JS, Kaleeswari R (2016) Implementation of vector field histogram based obstacle avoidance wheeled robot. In: *Online international conference on green engineering and technologies*, pp 1–6
17. Borenstein J, Koren Y (1991) The vector field histogram-fast obstacle avoidance for mobile robots[C]. *Int Conf Robot Autom* 7(3):278–288
18. Xiong WY, Yong TY, Xuan L, Hua LL (2019) Adaptive dynamic window method for crossing dense obstacles. *Control Decis Mak* 34(05):927–936
19. Hui QC (2017) Research on indoor mobile robot navigation technology based on ROS [D]. *Hebei University of Technology*, Hongqiao
20. Bin H, Bing W, Chunxiang W, Ming Y (2012) A dynamic path planning method for AGV based on time window. *J Shanghai Jiaotong Univ* 46(06):967–971
21. Yan G, Jian W, Youxin M, Feng J (2010) Research progress of vehicle navigation dynamic path planning. *Highw Transp Sci Technol* 27(11):113–117
22. Haipeng S, Zhaichuanrun, Xingqun Z, Yanhua Z (2007) Dynamic path planning technology based on real-time traffic information. *Microcomput Inf* 24:177–178
23. Rong W (2016) Robot dynamic path planning based on Beidou position information. *Southwest University of science and technology*, Mianyang
24. Jingkun L (2014) Dynamic path planning of indoor mobile robot. *Zhengzhou University*, Zhengzhou
25. Yanshan Li (2019) Research on indoor mobile robot SLAM based on RGB-D camera [D]. *Changchun University of Science and Technology*, Changchun
26. Fuhua S (2019) Prediction of wheat yield in Henan province based on NAR neural network. *J Xuchang Univ* 38(05):118–122
27. Xiang W, Jixiang He, Lei S, Jing Z (2015) Research on prediction model of nitrite in aquaculture water based on NAR neural network. *Fish Mod* 42(04):30–34
28. Lushan Y (2016) Speed prediction and application based on NAR neural network. *Dalian University of Technology*, Dalian

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.