DYNAMIC PATH PLANNING FOR INSPECTION ROBOTS INCORPORATING IMPROVED A* AND DYNAMIC WINDOW APPROACH

Zhiguo ZHAO[1], Dong XIE[2], Shanzhen XU[3,*], Jianhui ZHU[4], Zhen XU[5]

With the rapid advancement of industrial automation and intelligent inspection, patrol robots have become indispensable tools for equipment monitoring and safety assurance. To enhance the performance of traditional A algorithms – including path discontinuity, redundant nodes, and poor adaptability to dynamic environments – this study proposes a hybrid navigation framework integrating the improved A* algorithm with the optimized Dynamic Window Approach (DWA). Simulation experiments demonstrate significant performance improvements: Node count, cumulative turning angle, and path length decreased by 11.11%, 40.93%, and 3.45% respectively in 20×20 grid maps. Corresponding reductions reached 41.67%, 69.02%, and 17.97% in 40×40 grid maps.*

Keywords: Inspection robots; path planning; A* algorithm; dynamic window approach; fusion algorithm

1. Introduction

As artificial intelligence and robotic technologies continue to advance at an unprecedented pace, mobile robots are being increasingly integrated into diverse industries, with path planning algorithms emerging as critical enablers for their safe and efficient navigation in complex operational scenarios [1,2]. Especially the inspection robot [3], as an intelligent mobile device, has become an indispensable and important tool in industry, logistics, security and other fields. Path planning serves as a critical enabler for autonomous navigation in robotic inspection systems, forming a cornerstone of their operational capability [4,5]. Global path planning focuses on the optimality and efficiency of the overall path, such as A* algorithm [6], rapidly-exploring random trees (RRT) algorithm[7], Dijkstra's algorithm [8], and ant colony algorithm [9], etc., while local path planning pays more attention to

[1] Prof., Faculty of Transportation Engineering, Huaiyin Institute of Technology, Huai'an, China, e-mail: zhaozg@hyit.edu.cn
[2] PG., Faculty of Transportation Engineering, Huaiyin Institute of Technology, Huai'an, China, e-mail: 1410101122@qq.com
[3] A/Prof., Faculty of Transportation Engineering, Huaiyin Institute of Technology, Huai'an, China, e-mail: xushzh@hyit.edu.cn
[4] Lect., Faculty of Transportation Engineering, Huaiyin Institute of Technology,Huai'an, China, e-mail: zhujianhui@hyit.edu.cn
[5] PG., Faculty of Transportation Engineering, Huaiyin Institute of Technology, Huai'an, China, e-mail: 1558463117@qq.com

real-time and dynamic obstacle avoidance capability, such as dynamic window approach(DWA) [10], timed elastic band(TEB) algorithm [11], and artificial potential field(APF) approach [12], etc.

The A* algorithm excels in fast computation and minimal path length but suffers from redundant nodes, frequent turns, and proximity to obstacles in large environments. To address these, researchers have proposed solutions: Wang et al. [13] enhanced A* by adding time cost to the evaluation function, pruning obstacle-adjacent child nodes, and replacing zigzags with arcs to shorten paths. Lian et al. [14] introduced 24-neighbor heuristics and collision threat costs in the heuristic function, improving obstacle avoidance. Dang et al. [15] integrated collision risk and movement costs into the cost function and used Reeds-Shepp (RS) curves during path expansion to reduce directional changes and collision risks, achieving smoother paths. These approaches address A*'s limitations by optimizing node selection, refining path geometry, and incorporating safety constraints.

While existing A* improvements show moderate gains, unmanned vehicles still face unknown obstacles during travel, necessitating real-time dynamic avoidance. Researchers have enhanced the DWA algorithm: Wang et al. [16] proposed a pre-trajectory selection mechanism using differential manifolds, integrating obstacle density and angular change into the cost function to improve path smoothness and obstacle avoidance efficiency. Zhang et al. [17] developed a distributed collision avoidance framework combining trajectory prediction, weighted risk metrics, and maneuver constraints in the cost function, enhancing both obstacle negotiation and system safety. These modifications address real-time dynamic challenges while optimizing DWA's performance.

Despite improvements to A* and DWA algorithms, challenges remain: A* struggles with redundant nodes and excessive turns, while DWA falters in high-density obstacle environments due to local optimization. To address this, this paper proposes a synergistic framework combining enhanced A* and DWA. The A* algorithm integrates obstacle weights into its heuristic function to reduce redundancy and improve trajectory quality. The DWA algorithm is modified to include a combined speed metric, enabling dynamic tracking of A*-generated path nodes. This fusion minimizes disruptions from dynamic obstacles during inspection robot navigation.

2. Research Foundation

2.1. Environmental modeling

The environment model includes map data, obstacle/target/start positions, providing essential environmental data. Raster-based mapping was selected for its simplicity and compatibility with the experiment. A Cartesian coordinate system is established using the raster's orthogonal edges as x-y axes, enabling precise point localization via $Map(x,y)$ coordinates.

2.2. Traditional A* algorithm

As a milestone in search algorithms, the A* algorithm [18] establishes itself as the gold-standard heuristic search technique for optimal pathfinding in discrete topological networks. It combines Dijkstra's completeness with best-first efficiency via a cost framework that merges verified path costs and admissible heuristics. This dual-cost design ensures optimality and polynomial-time performance in grid-based navigation, with trajectory generation driven by its objective cost model:

$$f(n) = g(n) + h(n), \tag{1}$$

where the evaluation function $f(n)$ comprises two fundamental components: $g(n)$ denoting the accumulated traversal cost from the origin node to current position n, $h(n)$ representing the heuristic estimate from n to the destination. The heuristic computation uses metrics like Manhattan distance (Eq. (2)) for orthogonal movement constraints:

$$dist = \left|x_{curr} - x_{next}\right| + \left|y_{curr} - y_{next}\right|, \tag{2}$$

where $dist$ is the function distance, (x_{curr}, y_{curr}) is the position of the current node in the raster map, and (x_{next}, y_{next}) is the position of the next node in the raster map.

2.3. Improvement of A* algorithm

The A* algorithm finds the shortest path by prioritizing paths with the lowest cumulative cost via a heuristic function. However, traditional A* often produces zigzag, non-smooth paths in path search. Moreover, heuristic underestimation can cause suboptimal paths or higher computational costs. To address this, this paper introduces environmental obstacle information μ, as shown in Eq. (5), representing obstacle information between the current node and the target node, and uses an atan function to adaptively adjust the heuristic weight, achieving efficient global path planning with dynamic cost estimation.

$$f(n) = g(n) + \alpha h(n), \tag{3}$$

$$\alpha = \tan^{-1}\mu, \tag{4}$$

$$\mu = \frac{\sum_{i=x_{curr}}^{x_{goal}} \sum_{j=y_{curr}}^{y_{goal}} obstacle(i,j)}{\sqrt{(x_{star} - x_{goal})^2 + (y_{star} - y_{goal})^2}}, \tag{5}$$

where (x_{star}, y_{star}) is the start node of path planning, (x_{goal}, y_{goal}) is the goal node of path planning, (x_{curr}, y_{curr}) is the currently active node in the path planning process.

$$obstacle(i,j) = \begin{cases} 1, & (i,j) \in C_{obs} \\ 0, & (i,j) \in C_{free} \end{cases}, \tag{6}$$

where C_{obs} identifies the presence/absence of an obstacle at the node, C_{free} is the absence of an obstacle at the node, obstacle(i, j) is the binary state of obstacle presence or absence at the node and the value is recorded.

2.4. Route optimization

The improved A* algorithm still produces non-smooth nodes and redundant segments in its global path, as shown in Fig.1 and 2. Here, the red node G denotes the goal, the green node S is the start, and the solid black line represents the initial global path. Green dashed lines indicate connections between nodes where the occupied grid cells contain no obstacles, while red dashed lines indicate connections where obstacles exist within the grid cells. To address this, after obtaining the initial global path, we start from the goal node G. If the connection between nodes occupies grid cells with no obstacles, we replace the parent node of the current node with its grandparent node, continuing this process until obstacles are encountered or the start node S is reached.

$$path(G,S) = G \to N_3 \to N_4 \to N_8 \to N_9 \to S\,, \tag{7}$$

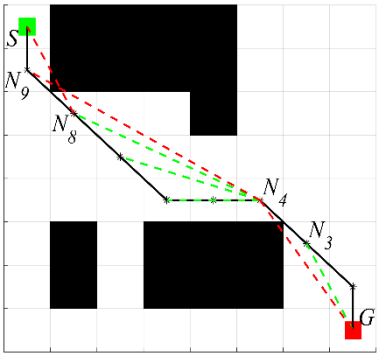


Fig.1. Optimization path map

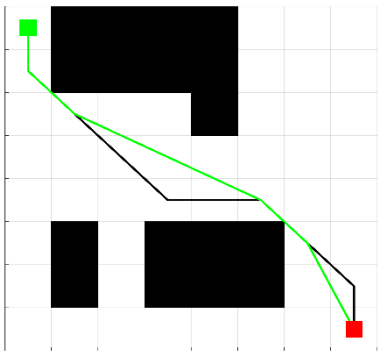


Fig.2. Optimized path result map

2.5. Traditional DWA algorithm

The DWA algorithm [19], a short-term motion planner, selects optimal velocity and direction to avoid obstacles and reach the target by predicting future trajectories based on the robot's motion model and state. It dynamically evaluates multiple candidate paths within a time window, choosing the best one for real-time navigation. The core of DWA lies in three factors [20]: (1) environmental constraints and robot capabilities; (2) velocity-based trajectory simulation; and (3) a scoring function to evaluate and select the optimal velocity. The transformation equation from position-based control to velocity control for the inspection robot is formulated as follows:

$$\begin{cases} \Delta x_y = -v_y \Delta t \cdot \sin(\theta_t) \\ \Delta y_y = -v_y \Delta t \cdot \cos(\theta_t) \end{cases}, \tag{8}$$

The kinematic relationship governing the inspection robot's motion can be formulated as follows:

$$\begin{cases} x = x + v_x \Delta t \cdot \cos(\theta_t) - v_y \Delta t \cdot \sin(\theta_t) \\ y = y + v_x \Delta t \cdot \sin(\theta_t) - v_y \Delta t \cdot \cos(\theta_t) \\ \theta_t = \theta_t + \omega \Delta t \end{cases}, \tag{9}$$

where Δx_y is the amount of change in x-coordinate in the forward direction, Δy_y is the amount of change in y-coordinate in the forward direction, v_y is the forward speed, Δt is the sample time step for motion control, θ_t is the amount of change in the angle, and ω denotes the yaw angle of the mobile inspection robot.

$$\begin{cases} V_s = \{(v,\omega) \mid v_{\min} \le v \le v_{\max}, \omega_{\min} \le \omega \le \omega_{\max}\} \\ V_a = \{(v,\omega) \mid v \le \sqrt{2dist(v,\omega)\dot{v}}, \omega \le \sqrt{2dist(v,\omega)\dot{\omega}}\} \\ V_d = \{(v,\omega) \mid v \in [v_c - \dot{v}\Delta t, v_c + \dot{v}\Delta t], \omega \in [\omega_c - \dot{\omega}\Delta t, \omega_c + \dot{\omega}\Delta t]\} \end{cases}, \tag{10}$$

where V_s represents the robot's kinematic constraint equation, V_a is the acceleration constraint formula of the robot and V_d is the dynamic window formula of the robot. The final velocity window of the robot is V_r, which is the intersection of the three velocities, as shown in Eq.(11)

$$V_r = V_s \bigcap V_a \bigcap V_d, \tag{11}$$

The rating function for the inspection robot is:

$$G(v,\omega) = \sigma\left(\alpha \cdot heading(v,\omega) + \beta \cdot dist(v,\omega) + \gamma \cdot velocity(v,\omega)\right), \tag{12}$$

Normalization of the rating function is:

$$normal\_head(i) = \frac{heading(i)}{\sum_{i=1}^{n} heading(i)}, \tag{13}$$

$$normal\_dist(i) = \frac{dist(i)}{\sum_{i=1}^{n} dist(i)}, \tag{14}$$

$$normal\_velo(i) = \frac{velocity(i)}{\sum_{i=1}^{n} velocity(i)}, \tag{15}$$

where $velocity(v,\omega)$ measures forward velocity, $dist(v,\omega)$ assesses obstacle proximity, and $heading(v,\omega)$ evaluates angular alignment with the target.

2.6. Improvement of DWA algorithm

The main improvement in dynamic path planning for inspection robots is the addition of dynamic obstacle prediction, specifically by incorporating a resultant velocity ($barr(v_c, v_b)$) into the DWA's evaluation function. This function assesses the angle of velocity between the inspection robot's movement velocity and the obstacle's movement velocity.

The rating function for the inspection robot is:

$$G(v,\omega)=K\left(\alpha\cdot heading(v,\omega)+\beta\cdot dist(v,\omega)+\gamma\cdot velocity(v,\omega)\right), \tag{16}$$

Normalization of the rating function gives:

$$normal\_barr(i)=\frac{barr(i)}{\sum_{i=1}^{n} barr(i)}, \tag{17}$$

where K is $barr(v_c,v_b)$; $barr(v_c,v_b)$ function evaluates the combined velocity between the inspection robot (v_{c1}) and a moving obstacle (v_{b1}), including the angle between their resultant velocity $v_{h1}=v_{c1}+v_{b1}$ and the obstacle's velocity v_{b1}. This is achieved via continuous LiDAR scanning to track the obstacle's velocity. When the angle is acute, a normal vector v_{h2} is derived from v_{h1}. By setting the robot's adjusted velocity v_{c2} such that $v_{c2}-v_{b1}=v_{h2}$, the optimized v_{c2} is calculated. The target velocity then determines the robot's wheel speeds.

3. Method

Global planners often struggle with real-time moving obstacle avoidance, while local planners may get stuck in local minima or dead-ends near U-shaped obstacles. This paper proposes a hybrid framework combining enhanced A* for global path optimization and modified DWA for local obstacle avoidance (Fig. 3).

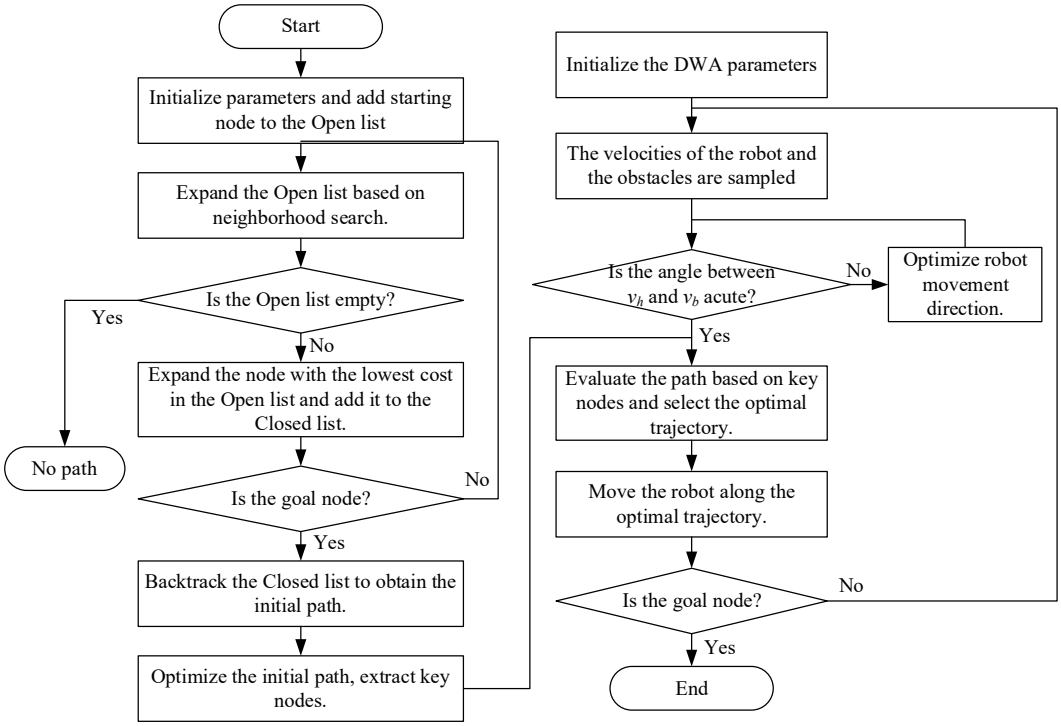
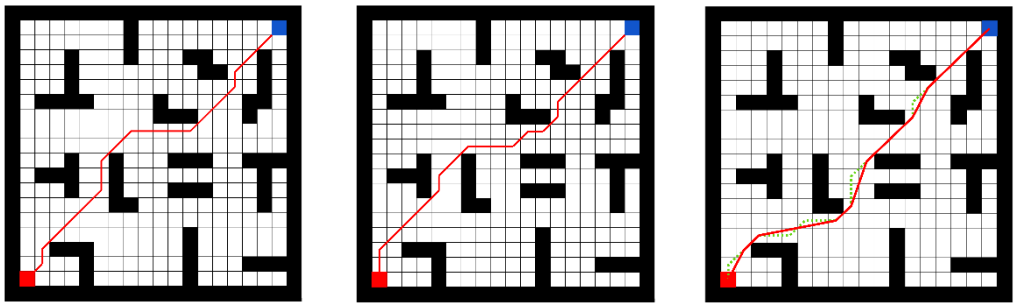


Fig.3. Flowchart of the fusion algorithm

The global layer generates an optimal path with key waypoints to avoid U-shaped obstacle traps, while the local layer enables real-time obstacle avoidance and trajectory adjustments between waypoints, ensuring dynamic evasion and quick target convergence.

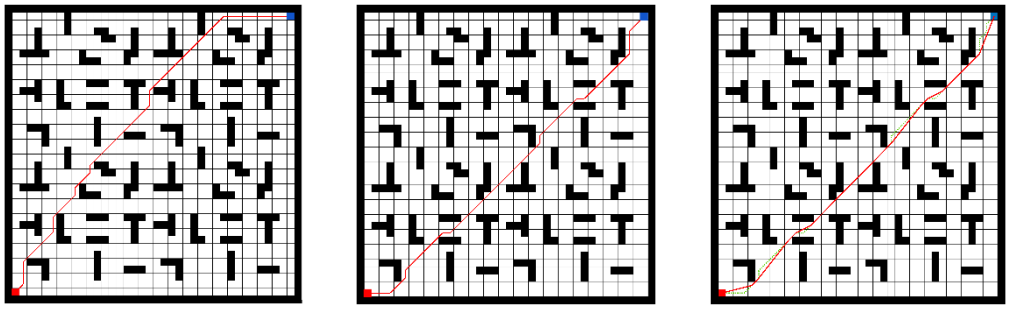
3.1. Simulation Verification of Improved A* Algorithm

The framework was validated via MATLAB 2022b simulations on an Intel® Core™ i5-12400H CPU@2.50GHz, using 20×20 and 40×40 grid maps (black lines: obstacles; white areas: passable space; S=start at bottom-left; G=target at top-right). Performance comparisons were conducted between the proposed hybrid algorithm, conventional A*, and Dijkstra's algorithm. Results are shown in Fig. 4. To assess the robustness of the enhanced A* algorithm, this paper conducted a comparative analysis on a 40×40 grid map, evaluating performance against the conventional A* pathfinding algorithm, Dijkstra's algorithm, and the proposed method. Experimental outcomes are depicted in Fig. 5.



(a) Traditional A* algorithm (b) The Dijkstra algorithm (c) Improved algorithms

Fig.4. Comparison of three route planning methods in 20×20 grid-based maps



(a) Traditional A* algorithm (b) The Dijkstra algorithm (c) Improved algorithms

Fig.5. Comparison of three path planning algorithms in 40×40 raster map

Table 1 presents the comparative performance metrics of the three algorithms.

Table 1 demonstrates the comparative performance of the improved fusion algorithm against the traditional A* algorithm across two grid map scenarios. On a 20×20 grid, the proposed method achieved reductions of 11.11% in node count, 40.93% in cumulative angular deviation, and 3.45% in path length.

Table 1

Computational outcomes of the algorithm's numerical experiments

Map size	Algorithm	Number of nodes	Total degrees of corners	Length of path to the target node
20×20	Traditional A* algorithm	9	405°	26.38
	The Dijkstra algorithm	10	495°	26.37
	Improved fusion algorithms	8	239.25°	25.46
40×40	Traditional A* algorithm	12	540°	57.60
	The Dijkstra algorithm	12	495°	55.25
	Improved fusion algorithms	7	167.30°	47.25

When scaled to a 40×40 grid, these improvements were amplified to 41.67%, 69.02%, and 17.97%, respectively. These results indicate that the algorithm effectively minimizes computational complexity (via reduced nodes), motion energy consumption (via fewer turns), and path redundancy in complex environments, thereby providing inspection robots with an optimized navigation strategy.

3.2. Fusion Algorithm Experimental Validation

To evaluate the capability of the enhanced fusion algorithm to determine optimal path trajectories and enhance obstacle avoidance for inspection robots in cluttered environments, the enhanced fusion algorithm was evaluated on grid maps of varying sizes (20×20 and 40×40) to demonstrate its scalability and path optimization performance in complex environments. The algorithmic parameters for the fusion method are summarized in the subsequent tables (Tables 2–4).

Table 2.

Robot kinematic parameters

Parameters	Value
Minimum Linear Velocity /(m/s)	0.00
Maximum Linear Velocity /(m/s)	2.00
Minimum angular velocity /(rad/s)	-20.00
Maximum angular velocity /(rad/s)	20.00
linear acceleration /(m/s^2)	1.00
angular acceleration /(rad/s^2)	50.00
Robot initial angle /(rad)	45.00

Table 3.

Operational Parameters for the DWA algorithm

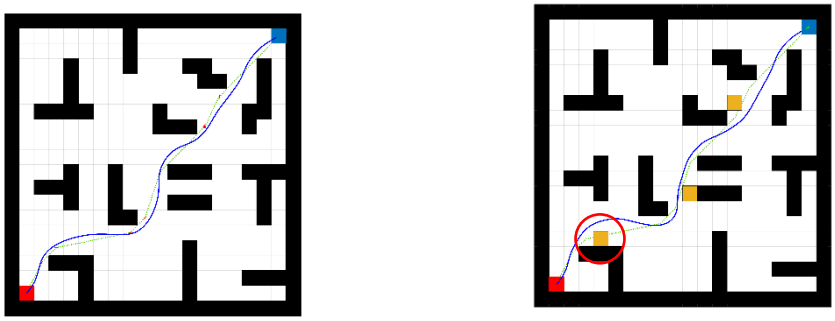
Parameters	Value
Linear velocity resolution /(m/s)	0.02
Angular velocity resolution /(rad/s)	1.00
time resolution /(s)	0.10
Trajectory prediction time /(s)	3.00
Obstacle Distance Threshold /(m)	0.50

Table 4.

Parameters of the evaluation function

Parameters	Value
combined velocity weight	0.40
azimuthal weight	0.07
distance weight	0.20
velocity weight	0.30

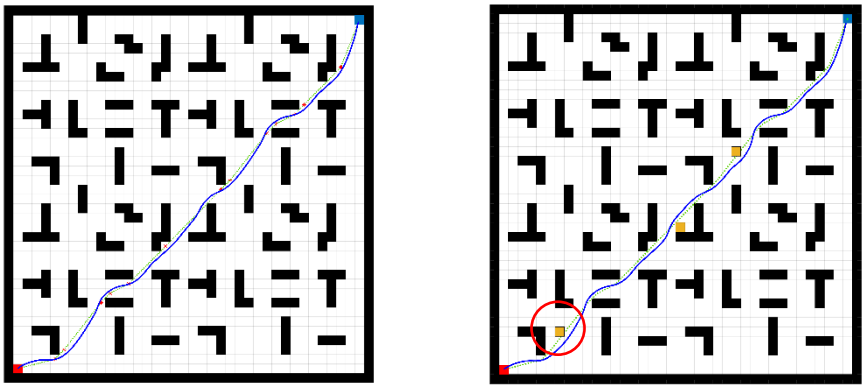
The enhanced fusion algorithm is depicted in Fig. 6 and 7, showcasing its performance advantages in complex navigation tasks.



(a) Simulation of improved algorithm (b) Improved Algorithm Sim with Obstacles

Fig.6. Comparison of accessibility improvement algorithms in 20×20 raster maps

Fig. 6 compares the proposed algorithm's performance in a 20×20 grid map (15% obstacles). The method generated 8 critical nodes to guide DWA toward the target. In Fig. 6(a), the hybrid algorithm navigated a U-shaped obstacle cluster near the start via node-based path optimization. In Fig. 6(b), newly added obstacles along the global path triggered local trajectory recalculation between nodes, ensuring collision-free navigation to the target.



(a) Simulation of improved algorithm (b) Improved Algorithm Sim with Obstacles

Fig.7. Comparison of accessibility improvement algorithms in 40×40 raster maps

Fig. 7 compares the proposed algorithm's performance in a 40×40 grid map with 15% static obstacles. The method generated 14 critical nodes to guide DWA,

avoiding obstacles in Figs. 7(a) and 7(b). The enhanced fusion approach, validated in 20×20/40×40 grids, avoids U-shaped dead-ends and improves inspection robot safety. Three static obstacles (red circles in Fig. 7b) were added; the path was reoptimized at the nearest critical nodes, demonstrating adaptive trajectory adjustments for safe navigation.

3.3. Inspection Robot Experimental Validation

The inspection robot runs on ROS-Melodic. Within the Navigation framework, its global and local path planners are replaced to the improved A* algorithm and the improved DWA algorithm, respectively. To validate the algorithm's feasibility, the inspection robot was outfitted with a suite of sensors. Fig. 8 illustrates the robotic system and its experimental setup, while Table 5 details the technical specifications of the inspection robot.

Fig.8. Inspection robot and experimental environment

Table 5

Specification Parameters of Inspection robot

Parameters	Value	Parameters	Value
host computer	Jetson Nano B01	speed	0~1.0m/s
ROM	32GB	Lower Computer	STM32F407
RAM	4GB	IMU	ICM20948
RGB-D camera	Astra Pro	size	21*20*18cm
lidar	rplidar A1	weight	2.1KG

Using the remote PC, we launch the LiDAR node, Cartographer node and RViz interface, enabling real-time control of the inspection robot to map the experimental site. the map building effect is shown in Fig. 9. To enable normal operation, a large environment is required for the inspection robot’s dynamic obstacle avoidance experiment. The improved algorithm uses pedestrians as obstacles and activates the robot’s path planning. Fig.10 marks the green arrow’s start as the target and its direction as the robot’s pose. Fig.11 shows the algorithm proactively replanning paths based on obstacle motion trends, enabling adaptive trajectory adjustments for enhanced safety.

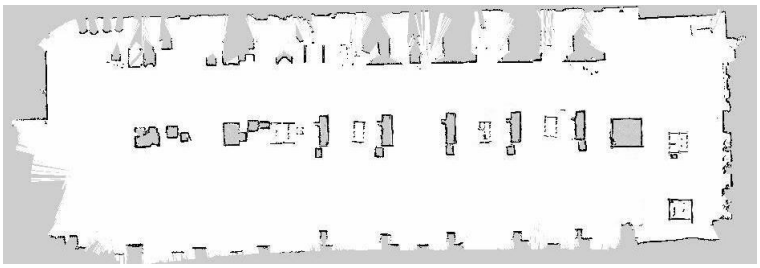


Fig.9. Map created by inspection robot

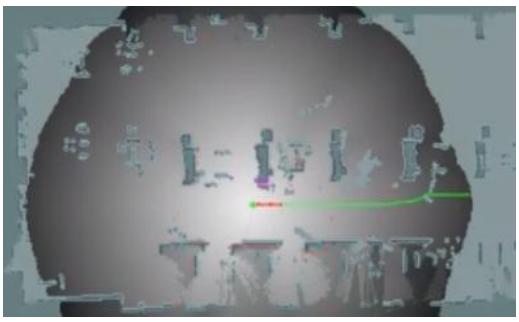


Fig.10. Inspection robot path planning diagram

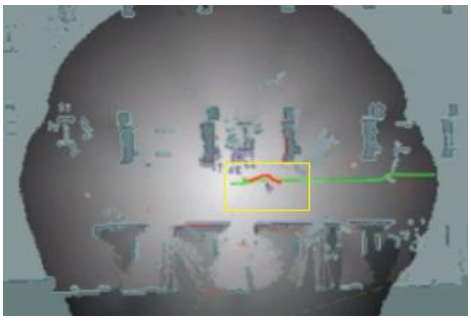


Fig.11. Dynamic obstacle avoidance diagram

As demonstrated by Figs. 10 and 11, the enhanced path planning algorithm prevents the inspection robot from halting in response to dynamic obstacles by proactively adjusting its trajectory. In addition, its total running time was much less; its overall running time was reduced by 15%. Table 6 compares the performance metrics of the baseline method and the enhanced path-planning algorithm.

Table 6.

Route Planning Methodology Comparison

	Path length /m	Number of turns /pc	runtime /s	average speed /m/s
original algorithm	4.03	1	26	0.15
improved algorithm	4.15	2	22	0.19

4. Conclusion

This paper addresses the A* algorithm's issues of node redundancy, low path smoothness, and safety, as well as DWA's tendency to fall into local optima. The algorithms were improved and fused to enable integrated obstacle avoidance for inspection robots in dynamic environments. Obstacle information was incorporated into A*'s cost function, and an atan function adaptively adjusted the heuristic weight for efficient global optimization. Node pruning enhanced path smoothness and safety. For DWA, path nodes were integrated into local planning, and velocity evaluation improved computational efficiency. Experiments showed reduced redundancy, smoother paths, and stronger obstacle avoidance while ensuring safety.

REFERENCES

[1] Carvalho J.P., Aguiar A.P., Deep reinforcement learning for zero-shot coverage path planning with mobile robots, IEEE/CAA Journal of Automatica Sinica, 2025, 1-16

[2] Ibrahim I., Decré W., Swevers J., Accelerated Reeds-Shepp and Under-Specified Reeds-Shepp Algorithms for Mobile Robot Path Planning, IEEE Transactions on Robotics, 2025, 2691-2708

[3] Li J., Zhou X., Gui C., Yang M., Xu F., Wang X., Adaptive climbing and automatic inspection robot for variable curvature walls of industrial storage tank facilities, Automation in Construction, 172, 2025, 106049.

[4] Annadata Y.S., Thazhathethil A., Moganarengam V., Nikoubin T., Enhancing Drone-Based Precision Agriculture: Performance Optimization of TinyML Models on Edge Devices and Adaptive Path Planning, SN Computer Science, 6(2), 2025, 132.

[5] Abou-Bakr, E., Alnajim, A.M., Alashwal, M. and Elmanfaloty, R.A., Chaotic sequence-driven path planning for autonomous robot terrain coverage. Computers and Electrical Engineering, 123, 2025, 110032.

[6] Erke S., Bin D., Yiming N., Qi Z., Liang X., Dawei Z., An improved A-Star based path planning algorithm for autonomous land vehicles, International Journal of Advanced Robotic Systems, 17(5), 2020, 1729881420962263.

[7] Hariprasath V., Mukesh S., Aruna T., Gayana R., Akshya J., Choudhry M.D., Sundarrajan M., Path Planning Solution for Intelligent Robots Using Ex* RRT Algorithm in Disaster Relief. 2024 5th International Conference on Recent Trends in Computer Science and Technology (ICRTCST): IEEE, 2024, 413-418.

[8] Dhouib S., Faster than Dijkstra and A* methods for the mobile robot path planning problem using four movement directions: the Dhouib-Matrix-SPP-4. Mechatronics and Automation Technology: IOS Press, 2024, 284-290.

[9] Lyridis D.V., An improved ant colony optimization algorithm for unmanned surface vehicle local path planning with multi-modality constraints, Ocean Engineering, 241, 2021, 109890.

[10] Ayalew W., Menebo M., Merga C., Negash L., Optimal path planning using bidirectional rapidly-exploring random tree star-dynamic window approach (BRRT*-DWA) with adaptive Monte Carlo localization (AMCL) for mobile robot, Engineering Research Express, 6(3), 2024, 035212.

[11] Turnip A., Faridhan M.A., Wibawa B.M., Anggriani N., Autonomous Medical Robot Trajectory Planning with Local Planner Time Elastic Band Algorithm, Electronics, 14(1), 2025, 183.

[12] Fu B., Chen Y., Quan Y., Zhou X., Li C., Bidirectional artificial potential field-based ant colony optimization for robot path planning, Robotics and Autonomous Systems, 183, 2025, 104834.

[13] Wang M., Yan L., Li Y., Path Planning of AGV in Workshop Based on Improved A* Algorithm, Autom Instrum, 38, 2023, 45-49.

[14] Lian X., Zhu X., Chen Y., Wang S., Research on path optimization method of substation intelligent inspection robot based on improved A* algorithm, Electronic Design Engineering, 31(18), 2023, 95-99.

[15] Dang C.V., Ahn H., Lee D.S., Lee S.C., Improved analytic expansions in hybrid a-star path planning for non-holonomic robots, Applied Sciences, 12(12), 2022, 5999.

[16] Wang H., Ma X., Dai W., Jin W., Research on obstacle avoidance of mobile robot based on improved DWA algorithm, Computer Engineering and Applications, 59(6), 2023, 326-332.

[17] Zhang W., Shan L., Chang L., Distributed collision avoidance algorithm for multiple unmanned surface vessels based on improved DWA, Control and Decision, 38(4), 2023, 951-962.

[18] Veisi O., Moradi M.A., Gharaei B., Maleki F.J., Rahbar M., Sustainable forestry logistics: Using modified A-star algorithm for efficient timber transportation route optimization, Forest Policy and Economics, 173, 2025, 103456.

[19] Bian Y., Ji P., Zhou Y., Yang M., Obstacle avoidance path planning of mobile robot based on improved DWA. Chinese Journal of Construction Machinery, 2021, 44-49.

[20] Chen Q., Lu Y., Wang Y., Zhu B., From topological map to local cognitive map: a new opportunity of local path planning, Intelligent Service Robotics, 14, 2021, 285-301.