Journal of Beijing Institute of Technology, 2026, Vol. 35, No. 2

Efficient Adaptive Collision-Free Path Planning Algorithm for Indoor Mobile Robots: Optimization and Application

Xiangyang Lu✉, Yandan Wang, Guangyi Zang, Junqi Wang
(School of Information and Communication Engineering, Zhongyuan University of Technology, Zhengzhou 451191, China)

Abstract: An adaptive path planning algorithm was proposed, which improves upon traditional A* by integrating an improved A* algorithm with the Dynamic Window Approach (DWA). This addresses the problems of slow search speed, unsmooth paths, and poor dynamic obstacle avoidance capability. Through an "8+5" neighborhood screening, a 16-neighborhood evaluation function, and a second-order then third-order Bézier curve optimization process, a Jetson Nano+ROS (Robot Operating System) is deployed to meet the requirements of efficient and safe navigation for fire inspection robots in complex environments. The results show that, compared with the original algorithm, the proposed algorithm reduces the average number of traversed nodes by 49.23%, the number of turns in the optimized path has decreased by approximately 28.82%, decreases curvature by 66.6%, and eliminates path tangency with obstacles. This also supports real-time obstacle avoidance with integration DWA, and outperforms traditional methods.

Keywords: adaptive path planning; improved A* algorithm; Bézier curve optimization; fire inspection robot; Jetson Nano+ROS system.

1 Introduction

Path planning serves as a cornerstone of inspection robotics, assuming heightened significance in firefighting scenarios, such as post-disaster search-and-rescue missions [1]. Currently, domestic research is predominantly oriented toward practical applications, emphasizing refinements to classical algorithms and the development of lightweight models, whereas international studies are centered on theoretical explorations within complex systems and multi-robot collaboration [2].

Path planning algorithms can be categorized into three primary classes. The first class comprises biomimetic algorithms, which emulate behavioral patterns of natural organisms, including ant colony optimization [3], particle swarm optimization [4], whale optimization algorithm [5], artificial bee colony algorithm [6], genetic algorithm [7], and gray wolf optimization algorithm [8]. These algorithms exhibit robust adaptability to complex nonlinear problems but are constrained by limitations such as parameter-dependent performance and high computational complexity.

One important area of advancement is the second class, denoted as intelligent algorithms, which encompasses deep learning [9,10], reinforcement learning [11], deep reinforcement learning [12,13], and hybrid approaches combining deep learning with traditional path planning algorithms [14–16]. Liu et al. [17] utilized residual Convolutional Neural Networks to train

Manuscript received Oct. 12, 2025; revised Dec. 18, 2025; accepted Mar. 18, 2026. The associate editor coordinating the review of this manuscript was Dr. Lijuan Jia. This work was supported by the National Natural Science Foundation of China (Nos. 61975015, 62375017).
✉ Corresponding author. Email: 5801@zut.edu.cn
DOI: 10.15918/j.jbit1004-0579.2025.068

unmanned aerial vehicles (UAVS) for path planning, integrating limited map information to generate feasible paths that approximate global optimality. Wang et al. [18] proposed a neural network-driven predictive heuristic method, which combines Convolutional Neural Networks (CNNS) to predict potential search areas in path planning and guide the search direction. However, the performance of this model is contingent upon the quality and diversity of inspection data, making it potentially inadequate to accurately predict optimal paths in complex or dynamic environments. Xu et al. [19] proposed a deep reinforcement learning-based path planning algorithm, optimizing paths via a Deep Q-Network (DQN) and validating the efficacy and superiority of deep reinforcement learning in this domain.

Classical planners are categorized into global methods (Dijkstra,A*,D*) and local methods (DWA, APF) [20]. Global approaches generate optimal paths in static, known maps but are hindered by high node expansion, fixed heuristics, and obstacle-tangent trajectories [21]. Local planners, conversely, respond rapidly to unknown obstacles yet risk local minima and lack global optimality [22]; neither operates adequately in isolation [23].

Recent works accelerate search efficiency (Mi et al. [24]), shorten paths (Li et al. [25], Ma et al. [26]), or smooth trajectories via Bézier/B-spline curves (Cao et al. [27], Lai et al. [28]). Most, however, overlook dynamic avoidance or obstacle tangency. Yang et al. [29] integrate a refined A* with DWA but employ a 5-neighborhood, and thus remain susceptible to tangential contacts.

Section 2 thus presents our 16-neighborhood adaptive A*,which enables global optimality. Section 3 details the simulation validation of the proposed adaptive A*-DWA hybrid approach, focusing on achieving real-time dynamic avoidance. Section 4 demonstrates its deployment on a fire-inspection robot; and Section 5 concludes and outlines future work.

2 Improved A* Global Path Planning Algorithm

2.1 Adaptive Child Node Search and Filtering

The fixed eight-neighborhood search in conventional A* often leads to inefficient node exploration and creates safety risks due to the path's proximity to obstacles. This paper addresses these limitations by presenting an adaptive "8+5" neighborhood search and selection technique. Unlike the rigid traditional model, this adaptive approach filters nodes more effectively, streamlining the search process and significantly enhancing the safety margins of the resulting path.

2.1.1 Title"8+5" Neighborhood Search Strategy

First, a vector $(\mathrm{d}x, \mathrm{d}y)$ is defined to determine the quadrant position of the target node relative to the current node, as shown in Equation.1

$$\begin{cases} \mathrm{d}x = x_{\mathrm{goal}} - x_{\mathrm{cur}} \\ \mathrm{d}y = y_{\mathrm{goal}} - y_{\mathrm{cur}} \end{cases} \tag{1}$$

In the equation, goal and cur denote the target and current nodes, respectively, with $\mathrm{d}x$ and $\mathrm{d}y$ representing their relative x- and y-axis positions. For the four quadrants: First quadrant: $\mathrm{d}x > 0, \mathrm{d}y > 0$ Second quadrant: $\mathrm{d}x < 0, \mathrm{d}y > 0$ Third quadrant: $\mathrm{d}x < 0, \mathrm{d}y < 0$ Fourth quadrant: $\mathrm{d}x > 0, \mathrm{d}y < 0$ A five-neighbor search is used in all four cases. Compared to the eight-neighbor search, this eliminates three nodes opposite the target-current direction, simplifying calculations while improving search relevance. When $\mathrm{d}x = 0$ or $\mathrm{d}y = 0$ (nodes aligned on a coordinate axis), traditional eight-neighbor search applies.

2.1.2 Child Node Screening Rules

Building on the improved search strategy, child node screening is required to enhance adaptability to complex environments and address tangency and intersection issues. This paper thus proposes an 8+5 neighborhood-based adaptive search and screening method (see Fig. 1). It illustrates the 8+5 neighborhood-based adaptive search and screening method.

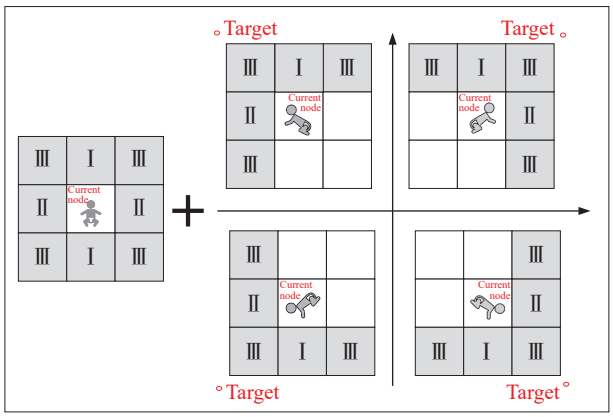


Fig. 1 Adaptive search and screening method based on 8+5 neighborhoods

For both five- and eight-neighborhoods, the current node's neighborhood is divided into three regions, each with distinct filtering rules:

Type I: Obstacle presence discards optional child node regions left and right of the obstacle grid.

Type II: Obstacle presence discards optional child node regions above and below the obstacle grid.

Type III: Obstacle presence requires no special treatment.

2.1.3 Handling Special Cases

A key innovation lies in the adaptive switch from five- to eight-neighborhood search when obstacles flank the current node (either vertically or horizontally), eliminating path generation failures in such edge cases.

The improved A* algorithm innovatively reduces traversed nodes, avoids tangency or intersection issues with obstacles, and resolves path generation failures in special scenarios (Fig. 2).

Fig. 2 shows pre- and post-improvement changes in evaluation metrics: traversed nodes dropped by 61.29% (from 341 to 132), obstacle-tangent trajectories were eliminated, and a slight path length increase-attributed to dangerous area avoidance-was observed.

2.2 Adaptive Heuristic Function

To balance navigational safety with operational efficiency, this study introduces an adaptive heuristic 16-neighborhood framework. This design specifically mitigates the path-length inflation typically triggered by obstacle-heavy environments while maintaining low latency. A key drawback of traditional A* is its arbitrary selection of equivalent nodes, as seen in Equation 2, which yields unpredictable path lengths. Our refined evaluation function resolves this by systematically prioritizing shorter paths, transforming a theoretically sound algorithm into a reliable tool for practical robot navigation.

$$f(n) = g(n) + h(n) \tag{2}$$

Here, $g(n)$ denotes the actual cost from the start node to currently nodes n, and $h(n)$ represents the estimated cost from n to the target node (heuristic function). This paper employs Manhattan distance as $h(n)$, calculated via

$$h(n) = |x_{\text{goal}} - x_{\text{cur}}| + |y_{\text{goal}} - y_{\text{cur}}| \tag{3}$$

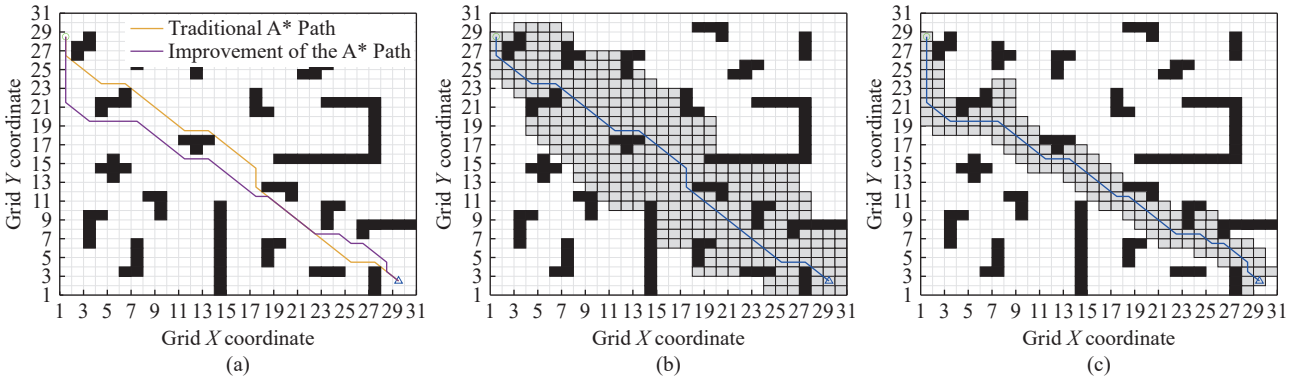


Fig. 2 Comparison of effects before and after adding adaptive search and filtering: (a) comparison of paths before and after improvement; (b) rendering before improvement; (c) rendering after improvement

To resolve the aforementioned issues and boost adaptability in complex environments, this paper optimizes child node selection by dynamically tuning the heuristic function $h(n)$ weight based on obstacle conditions within the 16-neighborhood of the current node,via the obstacle rate p. The 16-neighborhood grid features two layers: an inner 8 neighborhood (high threat, high weight) and an outer layer of four diagonally extended 8-neighborhoods (low threat, low weight).The obstacle rate p is defined as

$$p = \psi(x_1, x_2) = \frac{k_1 x_1 + k_2 x_2 + c}{16} \tag{4}$$

In the formula, x_1, x_2 denote the number of obstacle grids in the first and second layers, respectively (ranging from 0~8), k_1 and k_2 represent the weights of the first and second layers, c is a constant, and the denominator 16 corresponds to the total number of grids in the area. With k_1=1, k_2=0.857 and c=1, p varies within (0.062 5,1].

$$f(n) = g(n) + \varepsilon^{(1-\ln p)} h(n) \tag{5}$$

In the equation, ε is a constant greater than 1. Given the range of p,$\varepsilon^{(1-\ln p)}$ spans $[1,\infty)$, The obstacle rate p is inversely correlated with the adaptive weight:

In safe areas (low p, fewer obstacle):The weight $\varepsilon^{(1-\ln p)}$ increases,directing the algorithm toward the target to accelerate searching.

In complex areas (high p, more obstacles): The weight decreases, slowing the search to improve accuracy and ensure path safety and stability.

To identify the optimal values, we conducted a sensitivity analysis using 50 Monte Carlo runs across three map types (sparse, medium, and dense).

Weight Ratio (k_1/k_2): We varied $k_1 \in [0.8, 1.2]$ while keeping $k_1 > k_2$. Results indicated that increasing the ratio enhances safety by penalizing inner obstacles but extends path length. A ratio of $k_1/k_2 \approx 1.15$ (specifically $k_1 = 1$, k_2=0.875) provided the best trade-off between optimality andsafety. Ratios below 1.1 led to premature convergence, while ratios above 1.2 caused excessive detours.

Normalization Constant (c): The constant c ensures $p \in (0, 1]$. Tests varying $c \in [0.5, 2]$ showed negligible influence ($< 1.5\%$) on path metrics, so c=1 was selected for simplicity and interpretability.

In summary, the 16-neighborhood-based adaptive heuristic function effectively balances search efficiency with environmental safety.

2.3 Trajectory Smoothing

Motion stability and safety in patrol robots are heavily contingent on trajectory smoothness. Standard A* algorithms typically yield jagged paths with sharp vertices, triggering abrupt velocity shifts and inefficient power consumption. To mitigate these mechanical stresses, we implement a sequential second-to-third-order Bézier optimization. This dual-stage refinement smoothes the A* output while strictly maintaining obstacle clearance.

Since a Bézier curve's geometry hinges on its control points, we opt for a balanced approach: while higher-order curves offer superior flexibility, they also elevate computational and control complexity. Consequently, the three-point second-order curve, defined below, serves as our foundational model:

$$B(t) = (1-t)^2 P_0 + 2(1-t)tP_1 + t^2 P_2 \tag{6}$$

Here, P_0, P_1, P_2 denote the three control points of the second-order Bézier curve, with the curve parameter $t \in [0,1]$. The curve is generated via linear interpolation as defined as

$$\begin{aligned} A &= (1-t)P_0 + tP_1 \\ B &= (1-t)P_1 + tP_2 \\ P &= (1-t)A + tB = \\ &\quad (1-t)^2 P_0 + 2(1-t)tP_1 + t^2 P_2 \end{aligned} \tag{7}$$

2.3.1 Comparison of Optimization Effects

The conventional optimization approach Bézier 2nd (Strategy A) solely employs two-step quadratic Bézier curve processing. While this

mitigates some turns, residual sharp corners persist, hindering motion control. In contrast, the proposed strategy Bézier 2nd follow 3 rd (Strategy B) adopts a two-stage refinement: initial path smoothing via second-order curves, followed by third-order curves to polish remaining sharp segments —effectively eliminating abrupt angles (see Fig. 3 and Fig. 4).

Relative to Strategy A, Strategy B exhibits a significantly narrower and more stable curvature envelope. This eliminates sharp turns, generating trajectories that are inherently smoother and more tractable. The cascaded second- and third-order Bézier refinement thus endows inspection robots with paths meeting rigorous smoothness and drivability requirements.

3 Simulation Experiments and Analysis

This paper comprehensively validates the improved A* algorithm via the Matlab Simulation platform, covering its adaptability across diverse scenarios, comparative analysis with existing algorithms, and dynamic obstacle avoidance testing. Results demonstrate that the improved A* algorithm outperforms in path planning efficiency, safety, path quality and dynamic adaptability, making it suitable for practical applications in fire inspection robots.

3.1 Experimental Foundation

3.1.1 Simulation Environment

Experiments were conducted on the Matlab platform using a 30 m × 30 m grid map with multiple fixed obstacles, simulating residential, office, and complex environments. The start point is marked by a triangle, the target by a circle; white grids represent navigable areas, and gray grids denote traversed nodes. The diversity of experimental scenarios ensures the algorithm's applicability across varied environments.

3.1.2 Evaluation Metrics

1) Traversed nodes: Defined as the total number of nodes visited during the search process, calculated via

$$N = |N_{\text{CLOSED}}| \tag{8}$$

$|N_{\text{CLOSED}}|$ denotes the number of closed nodes in the list,i.e.,the total count of nodes that have been accessed and processed

2) Cumulative turning angle: In firefighting robot path planning, frequent turns reduce travel speed and risk stalls, overheating. Cumulative Turning Angle (CTA)-the sum of absolute heading changes from start to goal-quantifies path tortuosity and correlates directly with energy cost, execution time, and smoothness, expressed as

$$\Theta = \sum_{i=1}^{n-1} |\Delta\theta_i| \tag{9}$$

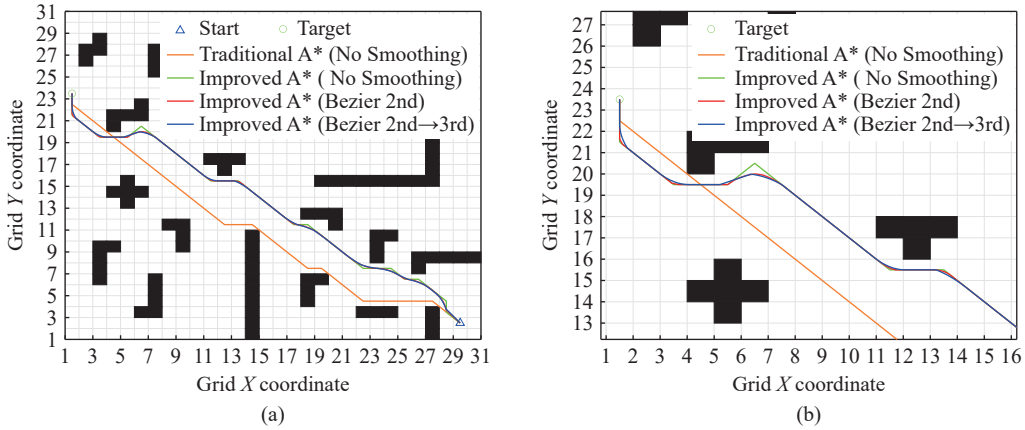


Fig. 3 Comparison of smooth paths under different optimization strategies: (a) comparison of optimized and unoptimized paths; (b) differences in the details of different optimization strategies.

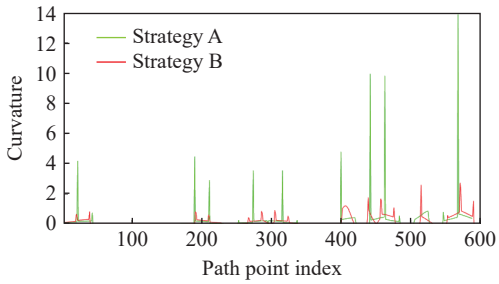


Fig. 4 Comparison of curvature changes under different optimization strategies

where $\Delta\theta_i$ is the direction change angle of the robot between the i-th and $(i+1)$-th path segments, and n is the total number of path segments. This parameter reflects the geometric complexity of the path, especially in structured or narrow spaces. The higher its value, the more tortuous the path is.

3) Number of turns: It's a concise, effective discrete metric for evaluating path quality in robot path planning. Widely used in path optimization, algorithm evaluation, and motion control modeling, it is defined (in grid maps or topological path representations, where paths consist of adjacent nodes or segments) as the count of path segments with directional changes, expressed as

$$N_{\text{turn}} = \sum_{i=2}^{n-1} \mathbb{I}(\theta_i \neq \theta_{i-1}) \tag{10}$$

Here, θ_i denotes the direction of the i-th path segment (e.g., up, down, left, right, or diagonal), and $\mathbb{I}(\cdot)$ is an indicator function, taking 1 if there is a direction change and 0 otherwise.

4) Curvature: For 2D plane curves, curvature quantifies the degree of bending. Typically, path planning requires continuous curvature without abrupt changes. Curvature is denoted by K, with its formula given as

$$K = \frac{|y'y''|}{(1+y'^2)^{\frac{3}{2}}} = \frac{|x'y'' - y'x''|}{(x'^2+y'^2)^{\frac{3}{2}}} \tag{11}$$

In the equation, y', y'' are the first and second derivatives of the curve equation, representing the tangent direction of the curve and the rate of change of this tangent direction, respectively. A large curvature value indicates sharp bending, manifesting as abrupt turns or directional changes (non-smooth segments). Conversely, a small curvature value corresponds to smooth curves with gradual directional transitions (smooth segments).

3.2 Adaptive Experiment

The traditional A* algorithm (A) and the improved A* algorithm (B) were compared across three environments: complex obstacle, residential, and office settings. Experimental results demonstrate that Algorithm B outperforms Algorithm A in traversed node count, cumulative turning angle, and number of turns, while generating smoother, obstacle-free paths (see Fig. 5 and Tab. 1).

Fig. 5 and Tab. 1 (CTA=cumulative turning angle; Nodes = number of nodes traversed.) reveal that the traditional A* algorithm employs an undirected 8-neighborhood search, leading to excessive redundant nodes, low efficiency, and potential obstacle intersections. In contrast, the improved algorithm adopts an "8+5" neighborhood strategy, reducing the average number of traversed nodes by 49.23% while eliminating safety risks.

Coupled with an adaptive evaluation function for final node screening, the improved algorithm achieves a 29.16% reduction in average cumulative turning angle and a 28.82% decrease in the number of turns, with zero obstacle intersections—validating the strategy's effectiveness. Significantly improved path quality, enhancing the efficiency, stability, and safety of agents in complex environments.

3.3 Comparative Experiment

The ant colony optimization (ACO) algorithm, Dijkstra algorithm, and rapidly exploring random tree (RRT) algorithm were included for comparison. Experimental results indicate that the improved A* algorithm outperforms these alternatives in search time, path length, and traversed node count, with enhanced path safety

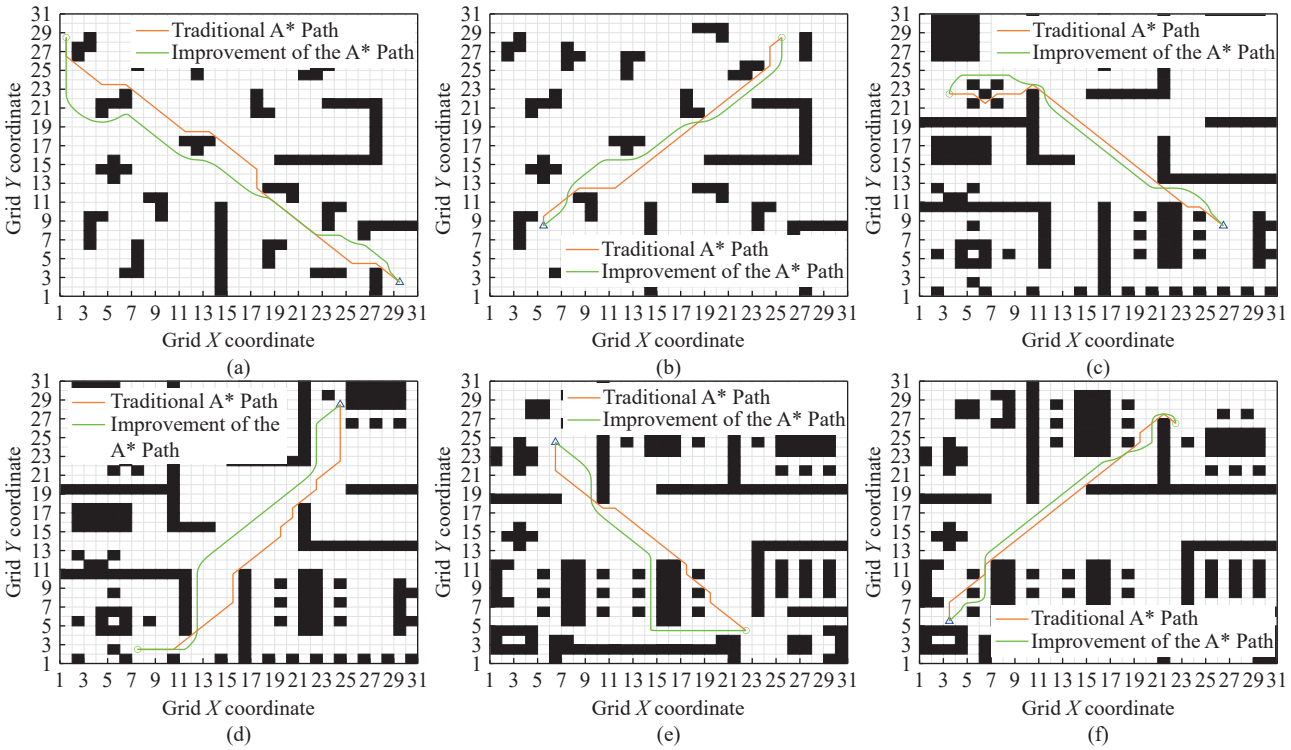


Fig. 5 Comparison of the effects of the A* algorithm before and after improvement in different environments: (a, b) complex obstacle environment; (c, d) home indoor environment; (e, f) office indoor environment; CAT represents cumulative turning angle

Tab. 1 Performance comparison of the A* Algorithm before and after improvement in different environments

Env.	Start	Goal point	Algorithm	Nodes	CTA	Number of turns	Turning or not
Complex	(1,28)	(29,2)	A	341	405	9	Yes
			B	130	345	7.7	No
	(29,2)	(1,28)	A	216	301	6	Yes
			B	98	225	5	No
Home	(26,8)	(3,23)	A	253	405	9	Yes
			B	96	289	6.4	No
	(24,28)	(7,2)	A	265	450	10	Yes
			B	131	235	5	No
Office	(6,24)	(22,4)	A	132	315	7	Yes
			B	95	242	5.4	No
	(3,5)	(22,26)	A	136	212	5	Yes
			B	84	135	3	No

(Fig. 6).Prior to experimentation, parameters for the relevant algorithms were configured as detailed in Tab. 2.

Fig. 6(a) (logarithmic time scale) shows ACO,RRT, and Dijkstra have much longer median search times (6 835 ms,1 154.7 ms,1 557.5 ms) than traditional and improved A*,with the improved A* being slightly faster.

Fig. 6(b) reveals ACO and RRT have unstable path lengths and poor obstacle safety due to randomness; Dijkstra's path is longest; improved A*'s median path is 0.4 m shorter than traditional A*.

Fig. 6(c) indicates Dijkstra traverses the most nodes (619); traditional A* (8 neighborhood) has a median of 178,while improved A* (8+5-neighborhood) reduces this to 125.

Fig. 6(d) shows traditional A* and ACO paths colliding with obstacles, whereas improved A*'s path is smooth and collision-free. In sum-

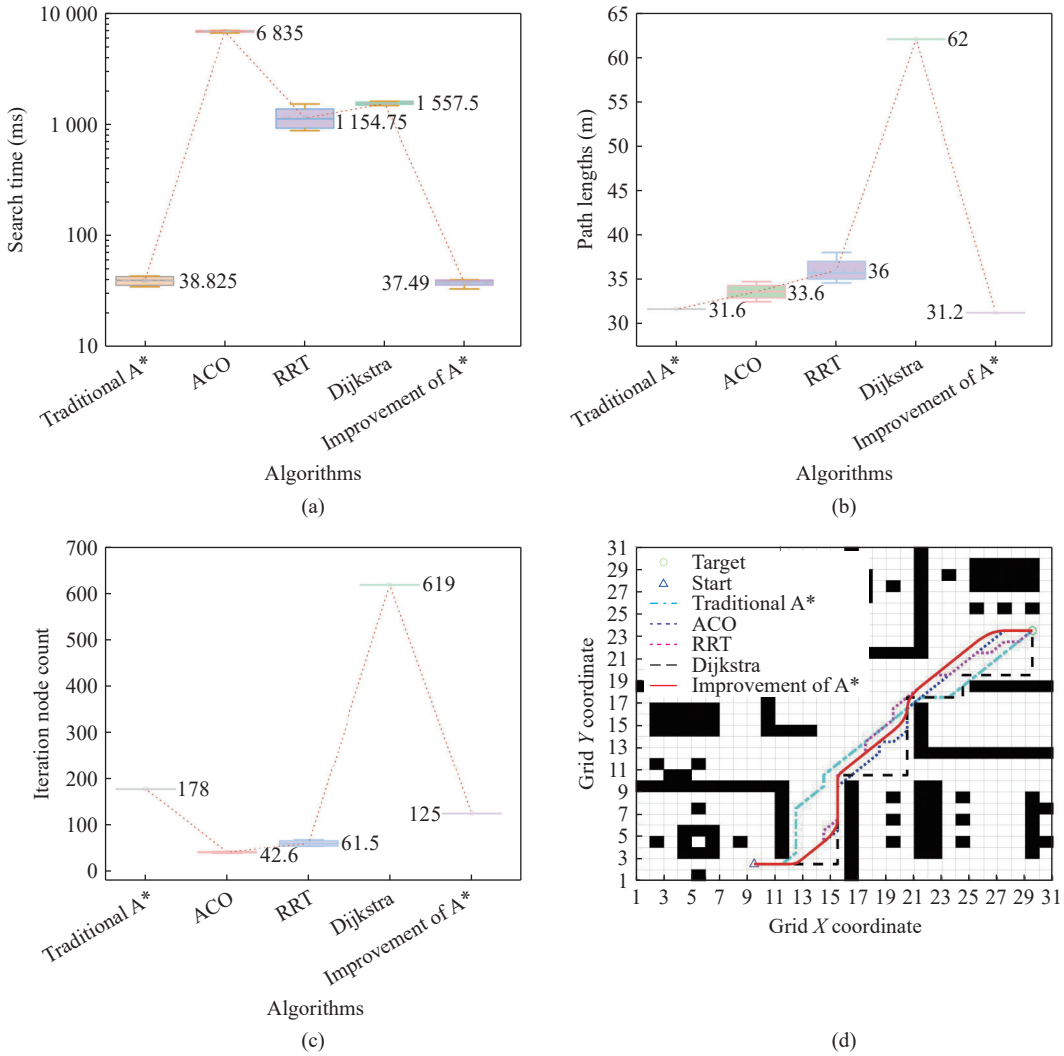


Fig. 6 Comparison of curvature changes under different optimization strategies: (a) search time comparison; (b) path length comparison; (c) comparison of node traversal counts; (d) generate path comparison

Tab. 2 Comparison of experimental algorithm parameter settings

Parameter name	Parameter value	Parameter name	Parameter value
Ant Count	20	Maximum Iteration Count t	100
Pheromone Factor α	1	RRT Steps	1
Heuristic function factor β	5	RRT random sampling probability	0.5
Pheromone evaporation factor ρ	0.3	RRT distance threshold	1
Pheromone constant Q	1	Map size	30 m×30 m

mary, the improved A* outperforms traditional A*,RRT, ACO, and Dijkstra in time, length, smoothness, and safety.

3.4 Dynamic Obstacle Avoidance Experiment

Path planning is critical to inspection robots' task execution, with its efficiency and safety directly determining mission success. Intelligent fire inspection robots operate mainly in narrow indoor environments with dynamic obstacles. The A* algorithm, a classic global path planner, generates optimal start-to-end paths but lacks dynamic obstacle avoidance. In contrast, the DWA algorithm excels at real-time planning and dynamic avoidance yet risks local optima. To address this, this paper proposes an adaptive hybrid algorithm integrating improved A* and

DWA, aiming to achieve dynamic obstacle avoidance while ensuring global optimality.

The experimental results showed that the combined algorithm performed better in terms of angular velocity, linear velocity, and path curvature, generating smoother and safer paths that are suitable for complex environments (Fig. 7). Fig. 7 demonstrates that the proposed A*–DWA fusion eliminates obstacle tangency, reduces curvature median by 66.6%,and narrows the interquartile range from 0.417 to 0.257.Angular-velocity dispersion is compressed while its median

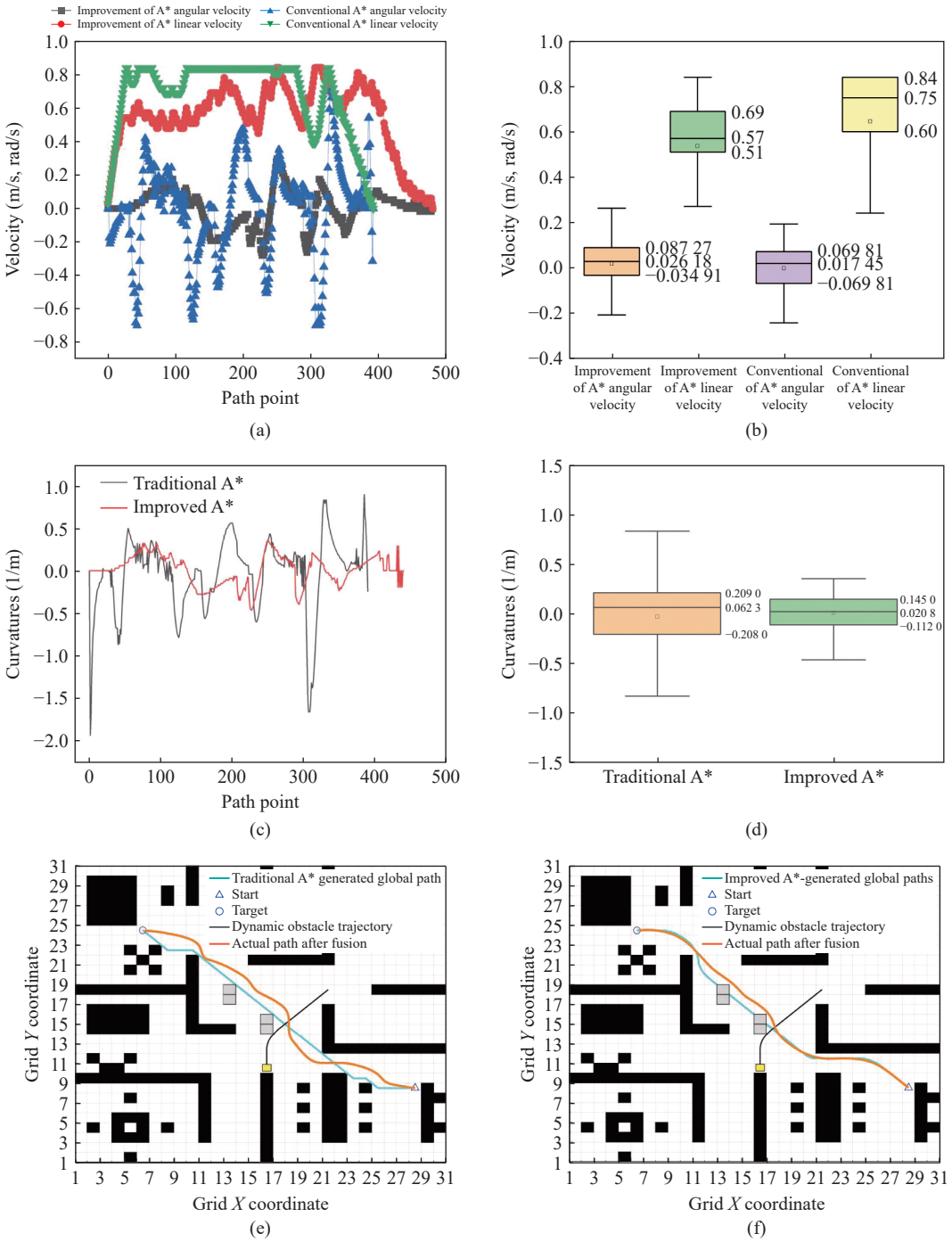


Fig. 7 Comparison of effects before and after fusion algorithm improvements: (a) angular and linear velocity profiles (linear velocity is measured in m/s, and angular velocity is measured in rad/s); (b) boxplots of velocity magnitudes; (c) curvature trace comparison; (d) curvature boxplots; (e) A*+DWA baseline dynamic avoidance; (f) improved A*+DWA dynamic avoidance

rises 52.3%;linear-velocity dispersion also contracts, with median speed falling 24%.The resulting trajectories are smoother and safer, satisfying the stringent accuracy demands of fire-inspection robotics.

4 ROS-Based Mapping and Navigation

To validate the effectiveness of the proposed path planning algorithm for fire inspection robots, experiments were conducted on the ROS platform. SLAM mapping was implemented using a LiDAR, and the improved A* - DWA fused path planning algorithm was deployed to achieve indoor mapping and navigation, thus verifying the algorithm's feasibility and practicality.

4.1 Robot URDF Modeling

URDF (an XML-based description framework) is used to define the 3D model of the inspection robot in ROS, including geometric shapes, joints, and connection details. URDF files work in conjunction with Rviz and Gazebo for visualization: Rviz interfaces with sensor plugins to perceive the environment and render the robot model, while Gazebo simulates a realistic map environment to demonstrate the robot's movement. The robot's 3D model can be visualized in Gazebo by creating a .launch startup file (see Fig. 8).

4.2 Actual Map Environment Setup

The experimental environment is an indoor scene enclosed by boards, with six static obstacles placed internally to meet inspection requirements and validate the algorithm's robustness (Fig. 9(a)). Upon initiating the SLAM algorithm, the robot scans the environment via LiDAR and constructs a 2D grid map using the gmapping algorithm. This algorithm leverages particle swarm poses and LiDAR data to continuously update the robot's position, enabling incremental map construction (Fig. 9(b)). Once map construction is complete, the map saver node saves the map, while the map saver node loads and publishes the map information. strategies.

4.3 Deployment of the Path Planning Algorithm

The proposed improved A* and DWA fusion path planning algorithm is deployed via the ROS Move base framework. Here, the improved A* serves as the global planner, while DWA functions as the local planner; their collaboration enables fast path planning and dynamic obstacle avoidance. The deployment process is as follows:

1) Launch the Move base node, create a function package, and add dependencies. The improved A* algorithm, rewritten in C++, is placed in a folder, added as a library file, and

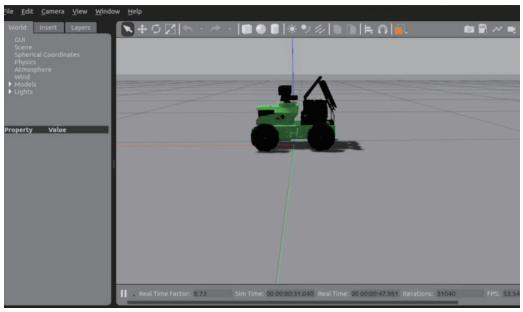
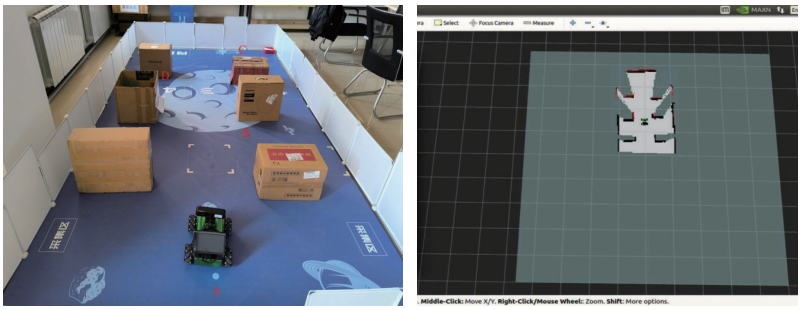


Fig. 8 Modeling and implementation of inspection robots



(a) (b)

Fig. 9 Mapping process: (a) indoor real map environment; (b) SLAM map construction

registered as a pluginlib plugin.

2) Compose the plugin description file, input the type and name into the XML file to ensure ROS can load the plugin.

3) Integrate the global and local path planning node plugins into the path.launch file, and visualize the path planning and navigation process using Rviz.

4.4 Implementation of Navigation Functionality for Fire Inspection Robots

Fire inspection robots operate in two modes: routine multi-waypoint patrol and emergency single-waypoint dash to the fire origin. In the emergency mode, SLAM updates the map, the improved A*–DWA planner calculates a collision-free path, and the robot executes it at maximum safe speed. The navigation workflow is as follows:

1) Set the start and end points.

2) The map_saver node invokes the global and local path planning plugins to generate feasible paths based on map and sensor data.

3) The controller drives the robot to avoid obstacles, dynamically adjusting the path according to real-time path information.

4) Real-time detection of target point arrival; the task terminates upon reaching the target.

In dynamic obstacle avoidance tests, the robot initially travels along the optimal path generated by the global planner. When encountering dynamic obstacles, it activates the local planner for avoidance, with the path adjusted in real-time based on obstacle positions to achieve dynamic obstacle avoidance (Fig. 10).

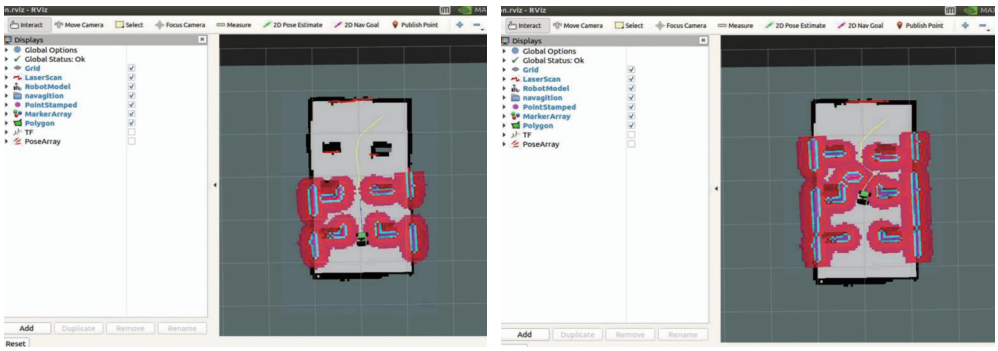
In the Fig. 11, dynamic obstacles move downward from the top. The path generated by the path planner is adjusted in real time according to the obstacle positions, enabling effective avoidance of dynamic obstacles.

5 Conclusion

Manual fire inspection and rescue operations are frequently hampered by inefficiency, low accu-

(a) (b)

Fig. 10 Changes in the movement direction of the small car before and after dynamic obstacle avoidance: (a) movement direction of the small car before obstacle avoidance; (b) movement direction of the small car after obstacle avoidance



(a) (b)

Fig. 11 Changes in the generated path before and after dynamic obstacle avoidance: (a) path generation for small vehicles before obstacle avoidance; (b) path generation for small vehicles after obstacle avoidance

racy, and the inherent risks of accessing hazardous zones. To mitigate these issues, intelligent inspection robots have become a critical asset. This study addresses the core navigational challenges of such systems —specifically path planning and target detection—by introducing a hybrid framework that synergizes an optimized A* algorithm with the Dynamic Window Approach (DWA).This integration facilitates adaptive navigation in complex environments, significantly curtailing search time while bolstering safety.

Our methodology refines the traditional A* algorithm through a sub-node search strategy utilizing an "8+5" neighborhood configuration, complemented by a specialized weight function for 16-neighborhood obstacle detection. Furthermore,we employ dynamic heuristic adjustments and segmented Bézier curves to smooth the trajectory. When fused with DWA, this enhanced approach ensures superior planning efficiency and motion fluidity. Validation via simulation and physical ROS deployment confirms the system's efficacy.

Future work acknowledging current experimental constraints, future research will target two primary areas:

Algorithmic diversification: Investigating data-driven planning paradigms, particularly those leveraging deep learning or reinforcement learning.

Active intervention: Addressing hardware limitations that currently preclude fire suppression. Future iterations will focus on structural upgrades to support and validate autonomous fire extinguishing capabilities.

References:

[1] L. Liu, X. Wang, X. Yang, H. Liu, J. Li, and P. Wang,"Path planning techniques for mobile robots: Review and prospect," *Expert Systems with Applications*, vol. 227, pp. 120254, 2023.

[2] H. Gao , D. Liu, and J. Hu, "A survey on path planning for mobile robot systems, "In *Proceedings of the 2023 IEEE 12th Data Driven Control and Learning Systems Conference (DDCLS)*, pp. 1176-1181, 2023.

[3] C. Miao, G. Chen, C. Yan, and Y. Wu,"Path planning optimization of indoor mobile robot based on adaptive ant colony algorithm," *Computers & Industrial Engineering*, vol. 156, pp. 107230, 2021.

[4] J. Zhao, C. Deng, H. Yu, H. Fei, and D. Li,"Path planning of unmanned vehicles based on adaptive particle swarm optimization algorithm," *Computer Communications*, vol. 216, pp. 112-129, 2024.

[5] Y. Dai, J. Yu, C. Zhang, B. Zhan, and X. Zheng,"A novel whale optimization algorithm of path planning strategy for mobile robots," *Applied Intelligence*, vol. 53, no. 9, pp. 10843-10857, 2023.

[6] Y. Cui, W. Hu, and A. Rahmani,"Multi-robot path planning using learning-based artificial bee colony algorithm," *Engineering Applications of Artificial Intelligence*, vol. 129, pp. 107579, 2024.

[7] N. Abu, W. Bukhari, M. Adli, and A. Ma'arif, "Optimization of an autonomous mobile robot path planning based on improved genetic algorithms," *Journal of Robotics and Control (JRC)*, vol. 4, no. 4, pp. 557-571, 2023.

[8] X. Liu, G. Li, H. Yang, N. Zhang, L. Wang, and P. Shao, "Agricultural UAV trajectory planning by incorporating multi-mechanism improved grey wolf optimization algorithm," *Expert Systems with Applications*, vol. 233, pp. 120946, 2023.

[9] T. V. Dang, and N. T. Bui,"Multi-scale fully convolutional network-based semantic segmentation for mobile robot navigation," *Electronics*, vol. 12, no. 3, pp. 533, 2023.

[10] D. H. Lee and J. L. Liu,"End-to-end deep learning of lane detection and path prediction for real-time autonomous driving," *Signal*, no. 1, pp. 199-205, 2023.

[11] H. Bae, G. Kim, J. Kim, D. Qian, and S. Lee, "Multi-robot path planning method using reinforcement learning, " *Applied Sciences*, vol. 9, no. 15, pp. 3057, 2019.

[12] Y. Wang, Z. He, D. Cao, L. Ma, K. Li, L. Jia, and Y. Cui,"Coverage path planning for kiwifruit picking robots based on deep reinforcement learning," *Computers and Electronics in Agriculture*, vol. 205, pp. 107593, 2023.

[13] Y. Qin, Z. Zhang, X. Li, W. Huangfu, and H. Zhang,"Deep reinforcement learning based resource allocation and trajectory planning in integrated sensing and communications UAV network," *IEEE Transactions on Wireless Communications*, vol. 22, no. 11, pp. 8158-8169, 2023.

[14] G. G. D. Castro, G. S. Berger, A. Cantieri, M. Teixeira, J. Lima, A. I. Pereira, and M. F. Pinto, "Adaptive path planning for fusing rapidly exploring random trees and deep reinforcement learning

in an agriculture dynamic environment UAVS," *Agriculture*, vol. 13, no. 2, pp. 354, 2023.

[15] X. Liu, D. Zhang, T. Zhang, J. Zhang, and J. Wang,"A new path plan method based on hybrid algorithm of reinforcement learning and particle swarm optimization," *Engineering Computations*, vol. 39, no. 3, pp. 993 -1019, 2022.

[16] L. Zhang, Y. Zhang, and Y. Li,"Path planning for indoor mobile robot based on deep learning," *Optik*, vol. 219, pp. 165096, 2020.

[17] Y. Liu, Z. Zheng, F. Qin, X. Zhang, and H. Yao,"A residual convolutional neural network based approach for real-time path planning," *Knowledge-Based Systems*, vol. 242, pp. 108400, 2022.

[18] J. Wang, J. Liu, W. Chen, W. Chi, and M. Q. H Meng,"Robot path planning via neural-network-driven prediction," *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 3, pp. 451- 460, 2021.

[19] S. Xu, Y. Gu, X. Li, C. Chen, Y. Hu, Y. Sang, and W. Jiang,"Indoor emergency path planning based on the q-learning optimization algorithm," *ISPRS International Journal of Geo-Information*, vol. 11, no. 1, pp. 66, 2022.

[20] H. Qin, S. Shao, T. Wang, X. Yu, Y. Jiang, and Z. Cao,"Review of autonomous path planning algorithms for mobile robots," *Drones*, vol. 7, no. 3, pp. 211, 2023.

[21] D. Zhang, C. Chen, and G. Zhang, "Agv path planning based on improved a-star algorithm, " In *Proceedings of the 2024 IEEE 7th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, vol. 7, pp. 1590-1595, 2024.

[22] C. Han, and B. Li,"Mobile robot path planning based on improved a* algorithm," In *Proceedings of the 2023 IEEE 11th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, vol. 11, pp. 672 - 676, 2023.

[23] W. Guan, and K. Wang,"Autonomous collision avoidance of unmanned surface vehicles based on improved a-star and dynamic window approach algorithms," *IEEE Intelligent Transportation Systems Magazine*, vol. 15, no. 3, pp. 36-50, 2023.

[24] Z. Mi, H. Xiao, and C. Huang, "Path planning of indoor mobile robot based on improved a* algorithm incorporating RRT and JPS, " *AIP Advances*, vol. 13, no. 4, pp. 1-10, 2023.

[25] Y. Li, R. Jin, X. Xu, Y. Qian, H. Wang, S. Xu, and Z. Wang,"A mobile robot path planning algorithm based on improved a* algorithm and dynamic window approach," *IEEE Access*, vol. 10, pp. 57736-57747, 2022.

[26] G. Ma, Y. Duan, M. Li, Z. Xie, and J. Zhu,"A probability smoothing bi-RRT path planning algorithm for indoor robot," *Future Generation Computer Systems*, vol. 143, pp. 349-360, 2023.

[27] M. Cao, B. Li, and M. Shi, "The dynamic path planning of indoor robot fusing b-spline and improved anytime repairing a* algorithm, " *IEEE Access*, vo. 11, pp. 92416-92423, 2023.

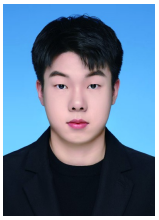
[28] R. Lai, Z. Wu, and N. Zeng,"Fusion algorithm of the improved A* algorithm and segmented bezier curves for the path planning of mobile robots," *Sustainability*, vol. 15, no. 3, pp.2483, 2023.

[29] H. Yang, X. Xu, and J. Hong,"Automatic parking path planning of tracked vehicle based on improved A* and DWA algorithms," *IEEE Transactions on Transportation Electrification*, vol. 9, no. 1, pp. 283-292, 2022.

Xiangyang Lu is currently a Doctor's degree professor. He completed his doctoral studies at Beijing Institute of Technology. He currently serves as a Master's supervisor at Zhongyuan University of Technology. Research areas: artificial intelligence, detection technology, information security countermeasures.

Yandan Wang is currently a Master's degree student. She received her Bachelor's degree in electronic information engineering major Engineering from Changchun University of Architecture, Changchun, Jilin, China, in 2024. She is now pursuing a Master's degree in New Generation Electronic Information Technology (including Quantum Technology) major. Her research interests mainly include robot path planning.

Guangyi Zang Guangyi Zhang is currently a Master's degree student. He received his Bachelor's degree in unmanned aerial vehicle system engineering from Zhongyuan University of Technology, Zhengzhou, Henan, China, in 2022. He is now pursuing a Master's degree in control science and engineering. His research interests mainly include deep learning and embedded systems.



Junqi Wang is currently a Master's degree student. He is now pursuing a Master's degree in School of Automation and Electrical Engineering, Zhongyuan University of Technology. His research interests mainly include infrared target detection and small target detection.