

A Hybrid Path Planning Framework for Inspection Robots: Integrating Improved A-Star Algorithm with Dynamic Safety Constraints

Chengwei Li¹, Yuan Zhou², Shoubin Wang³, Xiaofeng Bai², Youbing Li⁴

¹School of Intelligent Equipment Engineering, Wuxi Taihu University, Wuxi 214064, Jiangsu, China

²School of Instrument Science and Engineering, Harbin Institute of Technology, Harbin 150001, Heilongjiang, China

³School of Control and Mechanical, Tianjin ChengJian University, Tianjin 300384, China

⁴STECOL Corporation, Power Construction Corporation of China, Tianjin 300384, China

Corresponding author: Chengwei Li (e-mail: chengweiligood@163.com)

ABSTRACT Path planning is a core technology for enabling inspection robots to operate safely and efficiently in complex industrial spaces, including smart manufacturing sites where sensor reliability, wireless communication continuity, and electromagnetic interference may affect autonomous navigation. Although the classical A-star algorithm is widely used in global path planning because of its completeness and optimality guarantees, it often suffers from computational inefficiency, excessive node expansion, and non-smooth trajectories that are unsuitable for real robot motion. To address these limitations, this work proposes an enhanced A-star algorithm for inspection robots. The heuristic function is redesigned with an adaptive weighting mechanism, balancing search speed and path optimality and reducing the number of expanded nodes. A safety-aware strategy is further introduced by adding virtual safety buffers around obstacles, allowing the robot to maintain a minimum safe distance during navigation. Finally, Bézier curve interpolation is used to improve path smoothness and kinematic feasibility by generating continuous and differentiable trajectories. Simulation experiments show that, compared with the standard A-star algorithm, the proposed method reduces searched nodes by nearly 50% and redundant turns by more than 60%. Overall inspection efficiency is improved by up to 90%, confirming the method's practicality for robot navigation in complex engineering environments.

INDEX TERMS inspection robots, path planning, a-star algorithm, dynamic safety constraints, wireless inspection

I. INTRODUCTION

PATH planning constitutes a fundamental cornerstone of autonomous robotic systems, directly determining their operational efficiency and safety within the intricate spatial distribution of manufacturing environments [1-3]. This precision is essential for navigating the complex mechanical layouts required to monitor fiber morphology and maintain continuous manufacturing excellence. In inspection and maintenance applications, effective path planning is not merely a matter of geometric connectivity but a critical enabler for ensuring complete area coverage, avoiding costly collisions with infrastructure, and executing missions within stringent time and energy budgets. Without robust and intelligent path planning, robotic autonomy remains severely constrained, limiting their deployment in complex, hazardous, or dynamic environments where manual intervention is impractical or unsafe. Consequently, advancing the state of the art in path planning methodologies is essential

for unlocking the full potential of robots in increasingly demanding real-world applications.

The earliest path planning for robotic systems was primarily performed through manual teleoperation or by following predefined physical tracks. The adoption of computational graph-search methods marked a significant shift towards autonomy, with classical algorithms such as Breadth-First Search (BFS) and Dijkstra's algorithm being widely applied [4-6]. While BFS guarantees shortest-path finding in unweighted grids, and Dijkstra's algorithm extends this guarantee to weighted graphs, both methods are inherently limited by their exhaustive, non-directional search nature, resulting in substantial computational inefficiency for large-scale or real-time applications. This underscored the critical need for more guided and efficient search strategies that could preserve optimality while drastically reducing search space.

The A* algorithm is subsequently developed to address

this fundamental trade-off. By incorporating a heuristic function that estimates the cost to the goal, the search process is intelligently guided, prioritizing the most promising nodes. This approach not only guarantees optimality when an admissible heuristic is used but also achieves significantly higher efficiency compared to its predecessors [7-9]. As a result, A* and its variants have been established as a foundational framework for global path planning in robotics. The A* search algorithm has been extensively utilized as a cornerstone for global path planning in robotics due to its algorithmic completeness and guarantee of optimality under an admissible heuristic. However, for deployment in real-world robotic applications—spanning industrial inspection, autonomous navigation, precision agriculture, and planetary exploration—the classical A* algorithm was frequently deemed insufficient. Its limitations, including inefficiency in complex environments, generation of non-smooth paths incompatible with kinematic constraints, and neglect of critical factors like energy consumption and dynamic obstacles, were widely documented. To bridge this gap between theoretical optimality and practical applicability, a significant volume of research focused on enhancing the A* algorithm.

A primary avenue for improvement was the refinement of the heuristic function and the underlying search logic to enhance efficiency and adaptability. Traditional heuristic functions were modified to incorporate environmental or task-specific knowledge. For instance, an adaptive observation mechanism was integrated to dynamically adjust the heuristic based on mountainous terrain features, drastically reducing search nodes and time [10]. Similarly, to optimize for energy efficiency—a critical concern for electrically driven robots—the heuristic and cost functions were explicitly redesigned to incorporate motor energy consumption models, as seen in path planning for a heavy-duty six-legged robot and a spherical robot [11,12]. Multi-objective optimization was directly embedded into the search process through a weight-incorporating A* with a multifactor cost function, simultaneously evaluating path length, smoothness, and safety [13]. Beyond the heuristic, the fundamental search process was also enhanced. The search neighborhood was intelligently pruned and the node expansion strategy was optimized to minimize redundant computations, leading to significant gains in planning speed for general mobile robot navigation [14-16].

The jagged, grid-aligned paths produced by the standard A* were commonly post-processed or integrated with other methods to ensure smoothness and kinematic feasibility. A widely adopted strategy was the fusion with curve-fitting techniques. For example, paths generated by an improved A* for space robots were smoothed using B-spline curves to ensure continuous and executable trajectories on satellite surfaces [1-3]. Similarly, the Hybrid A* algorithm, designed for nonholonomic vehicles, was further improved by integrating Lemming Optimization to refine the search space and output smoother paths for autonomous vehicles. For robots with complex kinematics, such as underground load-

haul-dump (LHD) vehicles, the Hybrid A* was guided by pre-defined reference states to ensure the planned paths respected the vehicle's turning constraints[17].

To address dynamic and uncertain environments, the improved A* was employed as the global planner within a hierarchical framework, coupled with a local reactive planner. The most prevalent fusion was with the Dynamic Window Approach (DWA). This architecture was implemented for intelligent substation inspection robots, underground robots, and cable trench inspection robots. In these systems, the improved A* provided an optimal global path, while the DWA handled real-time local obstacle avoidance and velocity control, ensuring both global efficiency and local safety [18].

Substantial efforts were made to extend A*'s applicability beyond 2D grids. For UAV navigation in cluttered 3D airspace, a hybrid 3D-MIHE-RRT-A* algorithm was developed, combining a multi-indicator heuristic evaluation A* with a Rapidly-exploring Random Tree (RRT) for comprehensive exploration and optimization. For ground robots operating on construction sites, the algorithms cost function was modified to account for terrain slope and roughness, enabling safe and stable global planning on uneven surfaces. Furthermore, the algorithm was adapted for multi-robot coordination. In agriculture, an improved A* formed the basis for a full coverage path planning (FCPP) system. For building construction, a collaborative A* variant was designed to manage task allocation and conflict-free path planning for multiple on-site robots [19].

Despite its widespread adoption, the A* algorithm exhibits several well-documented limitations in practical robotic deployment, particularly for inspection tasks. Its performance is highly sensitive to heuristic design, often leading to excessive node expansions in complex or cluttered environments, which compromises real-time responsiveness. Furthermore, the paths generated in discrete configuration spaces tend to be suboptimal, containing unnecessary turns and disjointed segments that violate the kinematic and dynamic constraints of physical robots. Most critically, the conventional formulation lacks an explicit mechanism for embedding safety margins or reactive obstacle avoidance, rendering it inadequate for safe navigation in confined and dynamic operational settings.

To address the aforementioned challenges, a hybrid path planning framework centered on a refined A* algorithm is proposed to optimize robotic navigation within the complex spatial distribution of manufacturing environments. This approach ensures the high-precision monitoring of fiber morphology and mechanical operations necessary to maintain continuous manufacturing excellence. The core innovation lies in the introduction of a context-aware heuristic function that dynamically integrates environmental structural features. This adaptive heuristic mechanism significantly prunes the search space, leading to a substantial reduction in expanded nodes and a marked improvement in computational efficiency compared to the conventional A* approach.

These zones serve as cost-modulation regions in the grid map, ensuring that the path planning algorithm maintains a minimum safe distance throughout the navigation process. This allows the robot to preemptively adjust its speed in confined or hazardous areas, thereby enhancing operational safety and reliability in complex underground environments without compromising path optimality. Finally, to ensure the generated path is kinematically feasible for smooth robotic traversal, a Bézier curve-based smoothing technique is applied. This post-processing step effectively eliminates unnecessary abrupt turns and generates continuous, differentiable trajectories. Comprehensive simulation results demonstrate that the proposed integrated framework significantly enhances the overall navigation performance of inspection robots in pipeline environments, validating its superior effectiveness and practical utility.

The paper is organized as follow: The problem formulation is given in Section 2. The new path planning algorithm is proposed in Section 3. In Section 4, simulation examples are given to demonstrate the effectiveness of the method mentioned above. In Section 5, we discuss the results in simulation examples.

II. THE KEY COMPONENTS OF A* PATH PLANNING ALGORITHM

A. ENVIRONMENTAL MODELING METHOD

In this paper, the environment is modeled by the raster method, which converts the real spatial information of the environment into an abstract spatial model recognizable by the robot. The working principle of the grid method is to divide the limited space in which the inspection robot operates into small squares of the same size (grid), which is equivalent to transforming the working environment into a binary substitution, with “1” representing the obstacle region and “0” representing the non-obstacle region, as shown in Figure 1. The conversion into matrix form is shown in Eq (1). Obstacles are represented by occupied cells; the total count of these cells corresponds to the physical footprint of the obstacle, while their coordinates define its spatial distribution within the map [20].

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (1)$$

In the grid method, the selection of grid size affects the performance of the path planning algorithm. If the selected grid is too small, small obstacles will clearly appear in the grid, and the environmental information is also clear, which will increase the amount of stored information and sharply increase the calculation load. This will eventually reduce the search efficiency during path planning. If the selected grid is too large, environmental details are lost

and only large obstacles are resolved. While this simplifies the search space by reducing the total number of nodes, it significantly increases the risk of generating kinematically infeasible paths that may intersect with unmodeled smaller obstacles [21].

B. TRADITIONAL A* ALGORITHM

The A* algorithm effectively combines the Dijkstra algorithm and the Best First Search (BFS) algorithm, relying on heuristic information for the search of paths. Dijkstra algorithm is a typical algorithm for finding the shortest path. The advantage is that the path is the shortest, but the disadvantage is that the search is time-consuming. The BFS algorithm is a heuristic search algorithm, with the advantage of fast search speed, but the disadvantage is that the search path is not necessarily optimal. The A* algorithm combines the advantages of these two algorithms, and can accurately find the optimal path while improving the efficiency.

The A* algorithm performs path planning by expanding nodes. When the inspection robot is at the current node, the next best node is determined by the heuristic function, which is demonstrated in Eq. (2).

$$f(n) = g(n) + h(n) \quad (2)$$

where $f(n)$ is the evaluation function from the starting node to the target node; $g(n)$ is the actual cost from the starting node to the intermediate node n ; $h(n)$ is the estimated distance of the optimal path from the intermediate node n to the target node.

The search of the A* algorithm targets at eight directions, namely east, south, west, north, southeast, southwest, northeast, and northwest, as shown in Figure 2. In Figure 2, assuming that the Red Pentagon represents the position of the robot, the collection of grids in the next step is shown in Eq. (3).

$$\begin{aligned} E(i, j) = & (i, j + 1), (i, j - 1), (i + 1, j + 1), \\ & (i + 1, j - 1) \mid (i, j) \in A, \\ & (i + 1, j), (i - 1, j), (i - 1, j + 1), \\ & (i - 1, j - 1) \mid (i, j) \in A \end{aligned} \quad (3)$$

The steps of A* algorithm are as follows. From the starting node, traverse the adjacent nodes of the starting node, and evaluate the cost of each adjacent node according to the heuristic function; select the node with the highest priority (child node), and set it as the parent node; search for the adjacent node of this parent node, and select the best child node; repeat this step until the target node is found. The process of A* algorithm search is shown in Figure 3.

In Figure 3, where black represents obstacles; the blue circle represents the starting node; and the green circle represents the target node. In each search process, the colored area is the search area; and the red circle is the node with the highest priority (child node) in this search.

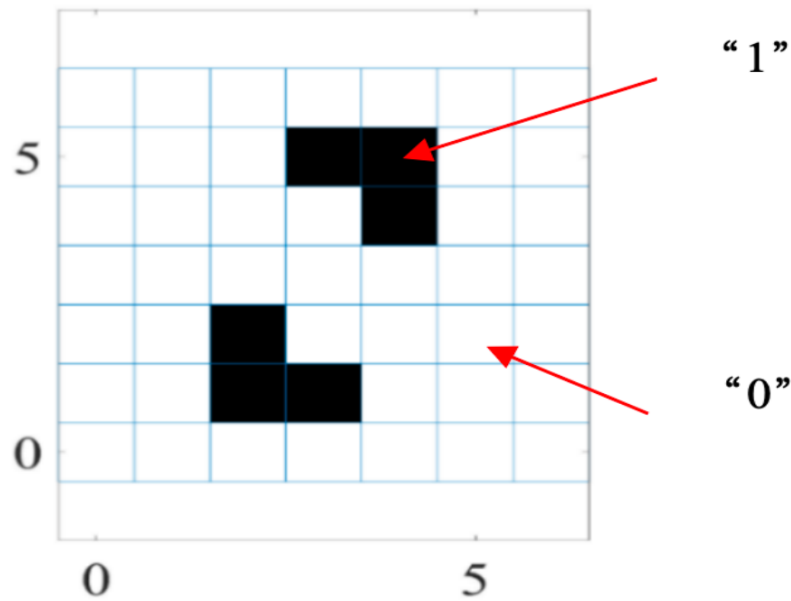


FIGURE 1. Grid method

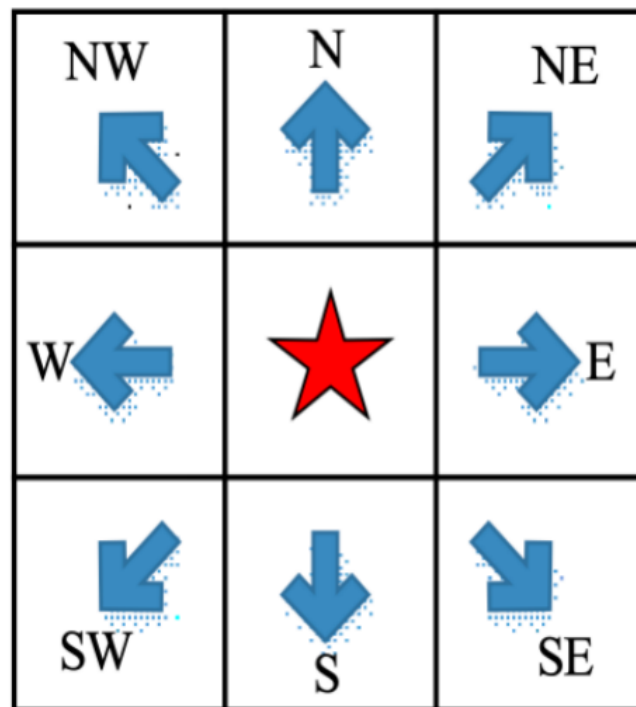


FIGURE 2. Eight search directions

The number in the upper left corner of the search area is $g(n)$; the number in the upper right corner is $h(n)$ and the middle number is $f(n)$. The cost graph of overall search is shown in Figure 4.

III. THE KEY STRUCTURE AND WORKING PROCESS OF THE PROPOSED ALGORITHM

A. THE IMPROVED CALCULATION METHOD OF HEURISTIC FUNCTION

The size of the estimated distance determines to some extent the search efficiency of the A* algorithm. Estimating the magnitude of a distance is generally estimated using three distance formulas, Euclidean distance, Manhattan distance and Chebyshev distance, of which the Euclidean distance

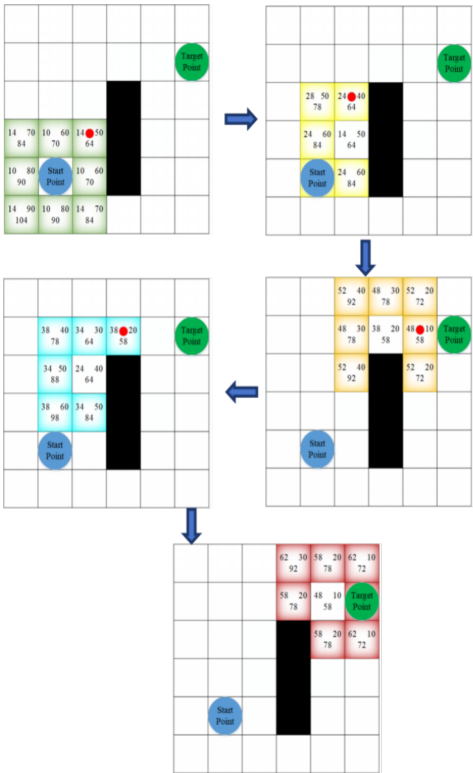


FIGURE 3. The process of A* algorithm search

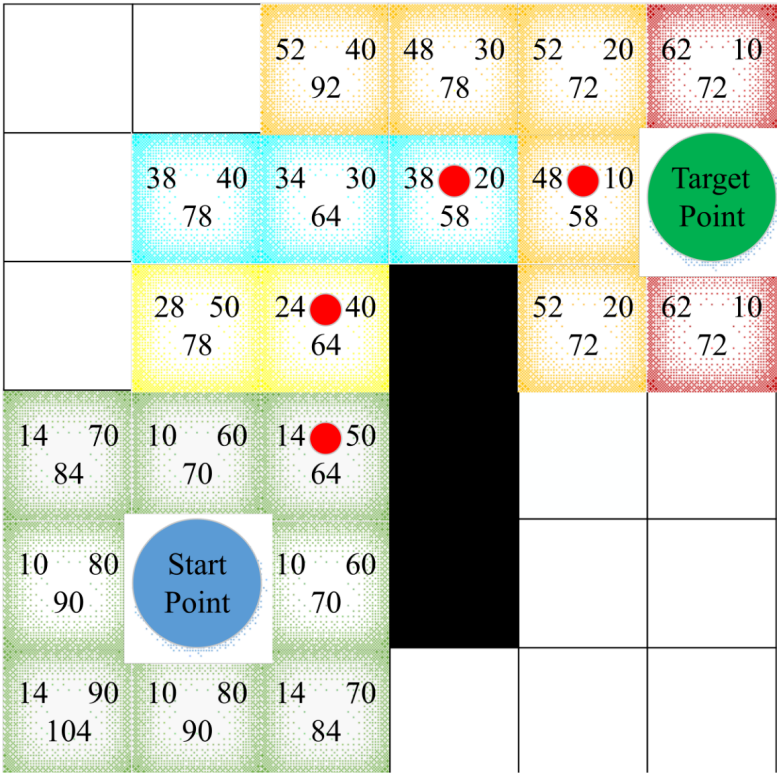


FIGURE 4. The cost graph of overall search

formula and the Manhattan distance formula are the most widely used in the A* path planning algorithm. The Eu-

clidean distance and the Manhattan distance are shown schematically in Figure 5. As can be seen from Figure 5,

the Euclidean distance is the straight-line distance between point A and point B. As can be seen from Figure 5, the Euclidean distance represents the straight-line distance between point A and point B, which is the shortest possible path in a 2D Euclidean space. The Manhattan distance, calculated as the sum of absolute differences of coordinates, is consistently greater than or equal to the Euclidean distance.

The formulas for Euclidean distance, Manhattan distance, and Chebyshev distance are shown in Eqs. (4)–(6).

$$D_{\text{Euclidean}} = |AC| = \sqrt{AB^2 + BC^2} \quad (4)$$

$$D_{\text{Manhattan}} = |AB| + |BC| \quad (5)$$

$$D_{\text{Chebyshev}} = \max(|AC|, |BC|) \quad (6)$$

This paper chooses the median of Euclidean distance and Manhattan distance as part of the heuristic function $h(n)$. The schematic diagram is shown in Figure 6, where the red dashed line is $h'(n)$. It is worth noting that $h'(n)$ is not necessarily this line, as long as it is from point A to point B and its distance is the median of the Euclidean distance and the Manhattan distance.

The formula for $h'(n)$ is shown in Eq. (7).

$$h'(n) = \frac{D_{\text{Euclidean}} + D_{\text{Manhattan}}}{2} \quad (7)$$

Substituting Eq. (4) and Eq. (5) into Eq. (7), we obtain Eq. (10).

$$h'(n) = \frac{|AC| + |AB| + |BC|}{2} \quad (8)$$

$h'(n)$ is weighted to get $h^*(n)$ in the improved heuristic function, as shown in Eq. (9), where q is the weight value. The purpose of weighting is to reduce the search range again and improve the search efficiency.

$$h^*(n) = h'(n) \times q \quad (9)$$

Substituting Eq. (7) into Eq. (9), we could obtain Eq. (10), which is the calculation formula of $h^*(n)$ in the improved heuristic function.

$$h^*(n) = \frac{|AC| + |AB| + |BC|}{2} \times q \quad (10)$$

Replacing $h^*(n)$ with $h'(n)$ in Eq. (4), the improved heuristic function $f^*(n)$ is shown in it. Substituting Eq. (10) into it, Eq. (11) is obtained, which is the calculation formula of the improved heuristic function $f^*(n)$.

$$f^*(n) = g(n) + \frac{|AC| + |AB| + |BC|}{2} \times q \quad (11)$$

The improved heuristic $f^*(n)$ enhances search efficiency by dynamically balancing the influence of Euclidean and Manhattan components. From a convergence perspective,

since the chosen median value $h'(n)$ remains underneath the actual cost to the goal (admissible) and the weighting factor q is applied consistently, the algorithm maintains its completeness and optimality. The adaptive weighting effectively prunes the search tree, ensuring that the search converges towards the goal direction more rapidly than the standard A* by reducing the expansion of nodes in the local minima of the cost surface.

B. THE DESIGN OF OBSTACLE AVOIDANCE DECISION

In the actual turning process, the A* algorithm will not take into account the inspection robot's own size and turning radius, resulting in the inspection robot due to the safety distance is too small and collision, as shown in Figure 7, the yellow circle illustrates the part of the inspection robot is the most likely to hit the obstacle location.

To solve the above problems, this paper optimizes the traditional A* algorithm to incorporate obstacle avoidance decisions, enabling the inspection robot to safely avoid obstacles. This optimization algorithm implements obstacle avoidance by assigning a virtual cost-modulation layer (inflation zone) around each obstacle. Within these zones, the algorithm adjusts the traversal cost in the evaluation function rather than direct kinematic velocity control. By increasing the cost value $g(n)$ for nodes adjacent to obstacles, the search process is mathematically biased to prioritize paths with higher safety clearances, effectively achieving 'modulation' through cost-driven path selection in the global planning phase. The obstacle avoidance decision map environment is shown in Figure 8, with black representing obstacles and blue representing safety buffers.

Figure 9 shows the result after the obstacle avoidance decision-making is added. It can be seen from Figure 9 that the newly planned path obviously has a safe distance from the obstacle, which can ensure that the inspection robot crosses the obstacle to the target node.

C. THE NEW PATH SMOOTHING METHOD

Since the path planned by the A* algorithm has more redundant nodes and turning points, the planned path needs to be smoothed to make the planned path more suitable for the inspection robot. In this paper, the Bessel curve method is utilized to smooth the path curve.

Bessel curve method mainly through the start point, termination point and two mutually non-overlapping intermediate points for graphing, the formula is shown in Eq. (12) for the quadratic Bessel curve method formula.

$$B(t) = (1-t)^2 P_0 + 2t(1-t)P_1 + t^2 P_2, \quad t \in [0, 1] \quad (12)$$

The fitted schematic is shown in Figure 10, where A, C, E, and F are the start point, the termination point, and two intermediate points that do not overlap each other, respectively.

In this paper, the n -order Bessel curve method is used for fitting and smoothing, and the formula is shown in Eq. (13).

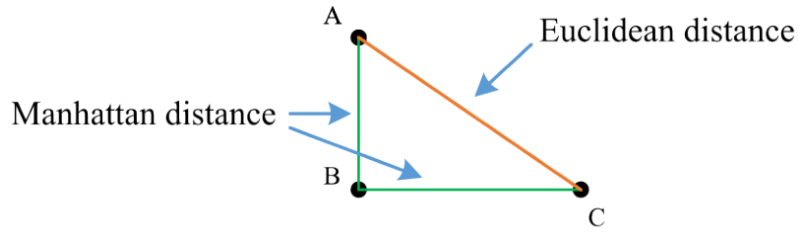


FIGURE 5. Distance diagram

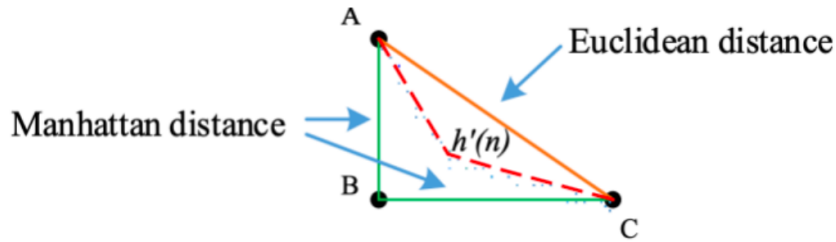
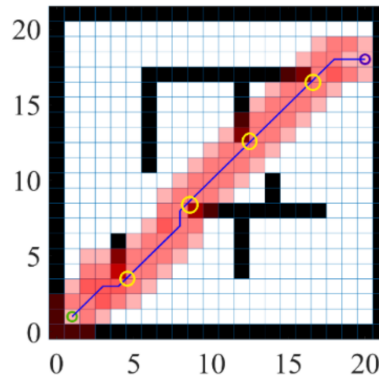
FIGURE 6. $h'(n)$ diagram

FIGURE 7. The location where traditional A* algorithm is prone to collision

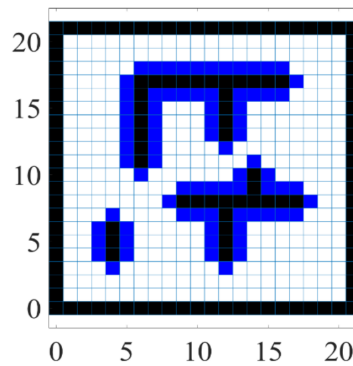


FIGURE 8. Add obstacle avoidance decision map environment

$$B(t) = C_0^n(1-t)^n p_0 + C_1^n t(1-t)^{n-1} p_1 \\ \dots + C_n^n t^n p_n = \sum_{i=0}^n C_i^n t^i (1-t)^{n-i} p_i \quad (13)$$

where $B(t)$ is the fitting point of the Bezier curve; t is the control parameter ($t=EO/EF=AE/AB=BF/BC$)
 P_i is the control point of the original path.

As shown in Figure 11, the paths are smoother after the smoothed fitting.

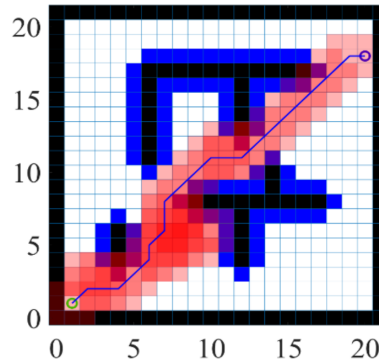


FIGURE 9. Operation results after adding obstacle avoidance decision

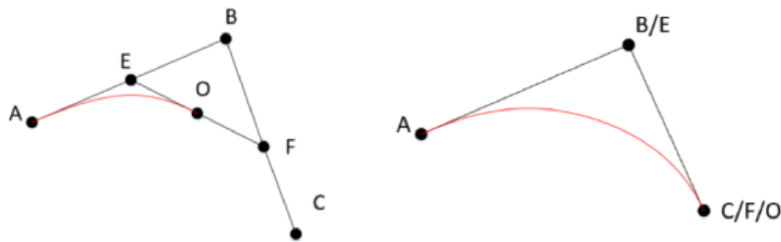


FIGURE 10. Bezier curve fitting

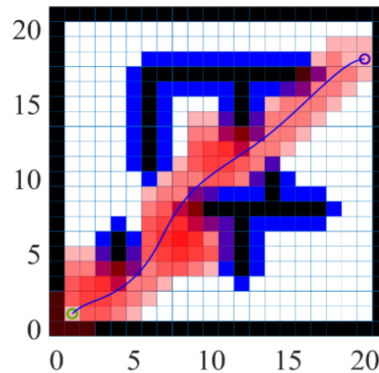


FIGURE 11. Operation results after Bezier curve fitting

The enhancement of the A* algorithm in this study is structured around three key components: environmental modeling and data preprocessing, the design of an adaptive heuristic function, and path smoothing via Bézier curve fitting. Initially, operational environment data, including the predefined locations of speed regulation zones, are processed to construct a detailed navigation map. Subsequently, a novel heuristic function is formulated to dynamically estimate path cost, effectively reducing node expansions and accelerating the search process. Finally, the raw path generated by the improved A* algorithm is optimized using Bézier curves to ensure smoothness and kinematic feasibility. This structured approach systematically addresses the limitations of conventional A* in complex settings while enhancing both efficiency and trajectory quality [22]. The

flow chart of the improved A* algorithm is shown in Figure 12.

IV. SIMULATION RESULTS

In order to verify the feasibility of the improved A* algorithm, this section constructs a raster map of 20×20 cells, and carries out simulation experiments on the four calculation methods of $h(n)$ in different raster environments and the same raster environment respectively.

A. SIMULATION EXPERIMENT IN DIFFERENT ENVIRONMENT

This experiment constructs a raster map of 20×20 grids with weight value q taken as 10. Three different obstacle environments are created in the raster graph. The simulation ex-

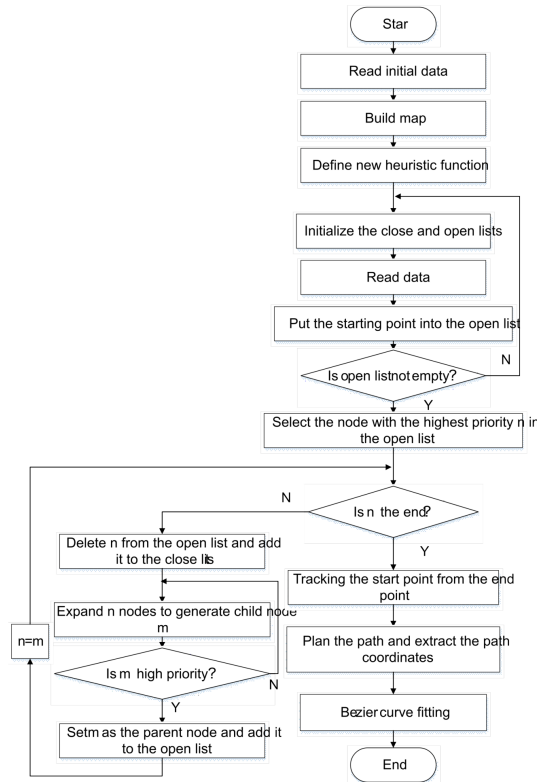


FIGURE 12. Flow chart of improved A* algorithm

periments are conducted in teams for the four computational methods of $h(n)$, with the green circle (1,1) as the starting node, the blue circle (20,18) as the target node, the black area representing the obstacle, the blue area representing the speed bump, and the red area representing the area of the search node. The results of the three experimental runs are shown in Figure 13-Figure 15.

The search path data is shown in Table 1. It can be seen that the search path of the improved A* algorithm shows a non-increasing trend compared to the other three algorithms.

TABLE 1. Searching path data in different environments

Number of experiments\algorithm	Experiment 1	Experiment 2	Experiment 3
Traditional A* algorithm	266.2	280.4	292.1
Weighted Manhattan A* algorithm	295.5	266.2	277.9
Weighted Euclidean A* algorithm	286.2	266.2	277.9
Improved A* algorithm	266.2	266.2	277.9

The search time is shown in Table 2 and the search time curve is shown in Figure 16.

TABLE 2. Search time in different environments (s)

Number of experiments\algorithm	Experiment 1	Experiment 2	Experiment 3
Traditional A* algorithm	5.252	5.031	4.620
Weighted Manhattan A* algorithm	1.313	1.043	1.671
Weighted Euclidean A* algorithm	1.387	1.275	2.085
Improved A* algorithm	1.169	1.020	1.450

From Figure 16, it can be seen that the search time of the improved A* algorithm is lower than the other three algorithms. The search node data is shown in Table 3 and the search node curve is shown in Figure 17.

TABLE 3. Searching node data in different environments (number)

Number of experiments\algorithm	Experiment 1	Experiment 2	Experiment 3
Traditional A* algorithm	2240	2232	2056
Weighted Manhattan A* algorithm	248	184	272
Weighted Euclidean A* algorithm	344	304	648
Improved A* algorithm	204	176	264

As can be seen from Figure 17, the improved A* algorithm searches fewer nodes than the other three algorithms.

B. SIMULATION EXPERIMENT IN THE SAME ENVIRONMENT

In this experiment, a raster graph of 20×20 grids is constructed, and the weight value q is taken as 10. An environment of obstacles is created in the raster graph, and three simulation experiments are conducted, Experiment 1 with (1,7) as the starting node and (20,18) as the target node; Experiment 2 with (6,8) as the starting node and (18,18) as the target node; Experiment 3 with (2,12) as the starting node and (9,15) as the target node.

The results of the three experimental runs are shown in Figure 18-Figure 20.

The search path data is shown in Table 4. As can be seen from Table 4, the improved A* algorithm search path

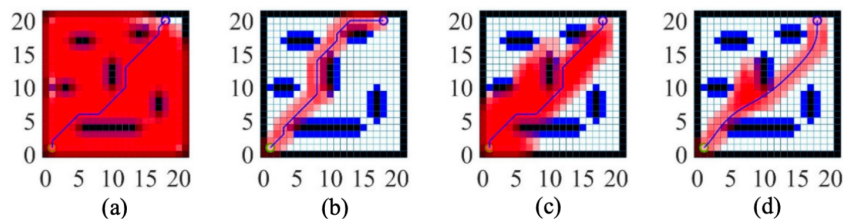


FIGURE 13. Comparison diagram of simulation results of experiment 1 in different environments: (a) shows the operation results of traditional A* algorithm; (b) shows the operation results of weighted Manhattan A* algorithm; (c) shows the operation results of weighted Euclidean A* algorithm; (d) shows the operation results of improved A* algorithm

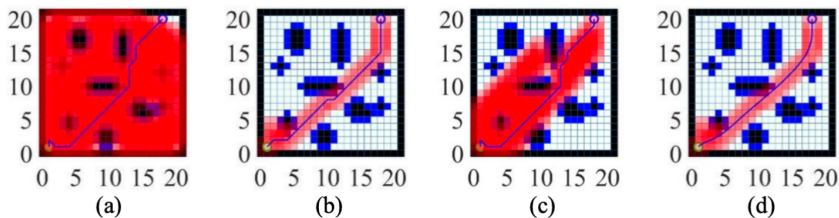


FIGURE 14. Comparison diagram of simulation results of experiment 2 in different environments: (a) shows the operation results of traditional A* algorithm; (b) shows the operation results of weighted Manhattan A* algorithm; (c) shows the operation results of weighted Euclidean A* algorithm; (d) shows the operation results of improved A* algorithm

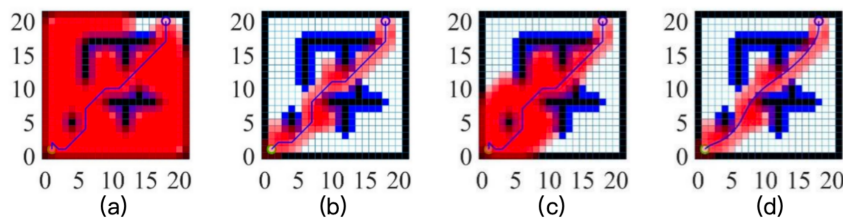


FIGURE 15. Comparison diagram of simulation results of experiment 3 in different environments: (a) shows the operation results of traditional A* algorithm; (b) shows the operation results of weighted Manhattan A* algorithm; (c) shows the operation results of weighted Euclidean A* algorithm; (d) shows the operation results of improved A* algorithm

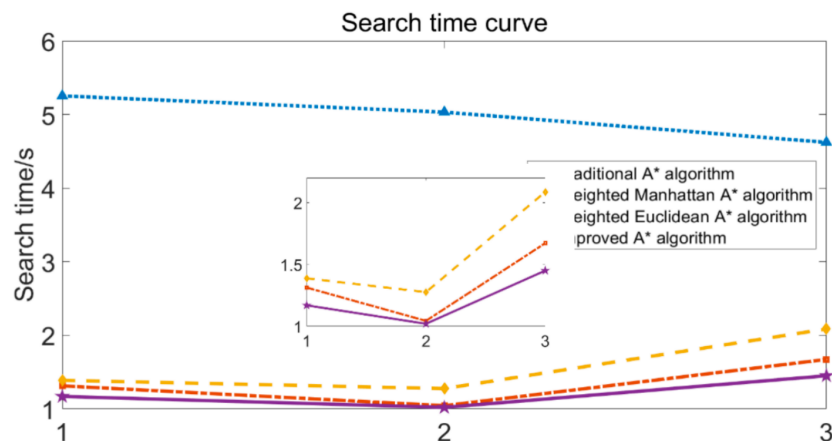


FIGURE 16. Searching time in different environments

shows a non-increasing trend compared to the other three algorithms.

TABLE 4. Searching path data in the same environment

Number of experiments\algorithm	Experiment 1	Experiment 2	Experiment 3
Traditional A* algorithm	249.7	181.4	138.9
Weighted Manhattan A* algorithm	249.7	167.2	138.9
Weighted Euclidean A* algorithm	235.5	167.2	124.8
Improved A* algorithm	235.5	167.2	124.8

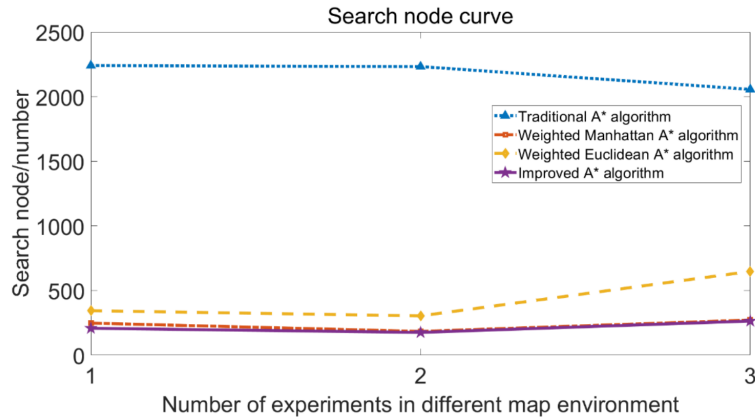


FIGURE 17. Searching node in different environments

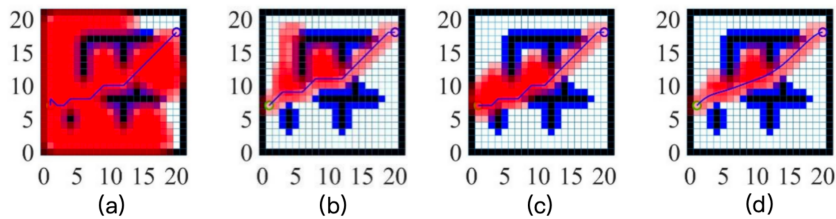


FIGURE 18. Comparison diagram of simulation results of experiment 1 in the same environments: (a) shows the operation results of traditional A* algorithm; (b) shows the operation results of weighted Manhattan A* algorithm; (c) shows the operation results of weighted Euclidean A* algorithm; (d) shows the operation results of improved A* algorithm

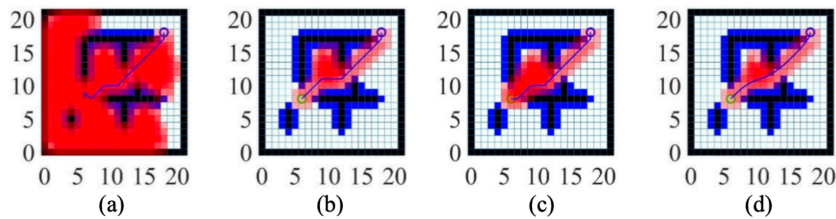


FIGURE 19. Comparison diagram of simulation results of experiment 2 in the same environments: (a) shows the operation results of traditional A* algorithm; (b) shows the operation results of weighted Manhattan A* algorithm; (c) shows the operation results of weighted Euclidean A* algorithm; (d) shows the operation results of improved A* algorithm

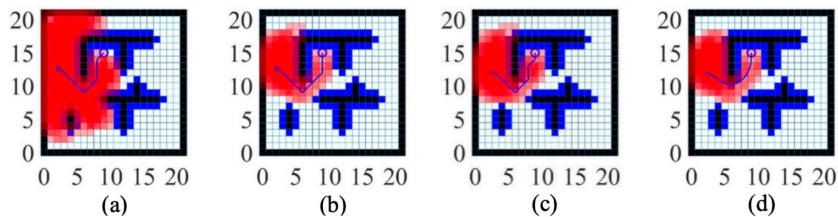


FIGURE 20. Comparison diagram of simulation results of experiment 3 in the same environments: (a) shows the operation results of traditional A* algorithm; (b) shows the operation results of weighted Manhattan A* algorithm; (c) shows the operation results of weighted Euclidean A* algorithm; (d) shows the operation results of improved A* algorithm

The search time is shown in Table 5 and the search time curve is shown in Figure 21.

TABLE 5. Search time in the same environment (s)

Number of experiments\algorithm	Experiment 1	Experiment 2	Experiment 3
Traditional A* algorithm	4.637	4.266	2.464
Weighted Manhattan A* algorithm	1.447	1.322	1.310
Weighted Euclidean A* algorithm	1.634	1.372	1.416
Improved A* algorithm	1.337	1.111	1.300

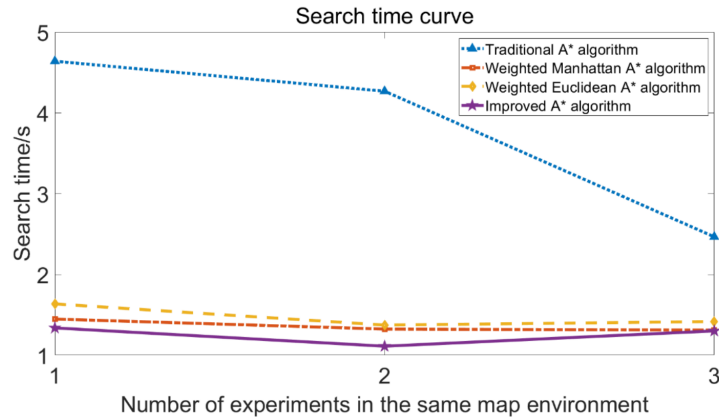


FIGURE 21. Searching time in the same environment

As can be seen in Figure 21, the search time of the improved A* algorithm is lower than the other three algorithms. The search node data is shown in Table 6 and the search node curve is shown in Figure 22.

TABLE 6. Searching node data in the same environment (number)

Number of experiments\algorithm	Experiment 1	Experiment 2	Experiment 3
Traditional A* algorithm	1912	1688	848
Weighted Manhattan A* algorithm	368	168	312
Weighted Euclidean A* algorithm	400	224	352
Improved A* algorithm	256	144	296

As can be seen from Figure 22, the proposed algorithm in this paper searches fewer nodes than the other three algorithms.

V. DISCUSSION

A. DISCUSSION ON DIFFERENT ENVIRONMENTAL EXPERIMENTS

It is important to note the trade-off between the safety margin (inflation radius) and map passability. While increasing the width of the 'safety buffers' (formerly referred to as speed bumps) enhances collision avoidance, an excessively large margin may block narrow passages, leading to the 'no feasible path' problem in cluttered environments. In this study, the buffer width is set to 1.5 times the robots radius, which provides a balanced constraint that ensures safety while preserving the topological connectivity of the grid map for inspection tasks.

In experiments in different environments, the improved A* heuristic algorithm is compared with the other three algorithms in terms of search time efficiency and search node efficiency, as shown in Table 7 and Table 8. The efficiency improvement graph for search time and the search node are shown in Figure 23 and Figure 24.

TABLE 7. Searching time improves efficiency data in different environments

Number of experiments\algorithm	Experiment 1	Experiment 2	Experiment 3
Improved A* algorithm compared to traditional A* algorithm	77.74%	79.73%	68.62%
Improved A* algorithm compared to weighted Manhattan A* algorithm	10.94%	2.27%	13.27%
Improved A* algorithm compared to weighted Euclidean A* algorithm	15.75%	20.06%	30.48%

TABLE 8. Searching node improves efficiency data in different environments

Number of experiments\algorithm	Experiment 1	Experiment 2	Experiment 3
Improved A* algorithm compared to traditional A* algorithm	90.71%	92.11%	87.16%
Improved A* algorithm compared to weighted Manhattan A* algorithm	16.13%	4.35%	3.07%
Improved A* algorithm compared to weighted Euclidean A* algorithm	39.53%	42.11%	59.26%

It could be seen from Figure 23 and Figure 24 that in terms of search time, the improved A* heuristic algorithm increases the efficiency by 75% on average compared with the traditional A* algorithm. Comparing with the weighted Euclidean A* algorithm, the efficiency is increased by an average of 22%/. Comparing with the weighted Manhattan, the efficiency is increased by an average of 8%. In terms of search nodes, the improved A* heuristic algorithm increases the efficiency by 89% on average compared with the traditional A* algorithm. An average increase of 46% compared with the weighted Euclidean A* algorithm. An average increase of 7% compared with the weighted Manhattan A* algorithm. In summary, in different experimental environments, the improved A* heuristic algorithm performs better than the other three algorithms.

B. DISCUSSION ON THE SAME ENVIRONMENTAL EXPERIMENT

In experiments in the same environment, the improved A* heuristic algorithm is compared with the other three algorithms in terms of search time efficiency and search node efficiency, as shown in Table 9 and Table 10. The efficiency improvement graph for search time and the search node are shown in Figure 25 and Figure 26 respectively.

TABLE 9. Searching time improves efficiency data in the same environment

Number of experiments\algorithm	Experiment 1	Experiment 2	Experiment 3
Improved A* algorithm compared to traditional A* algorithm	71.16%	73.96%	47.24%
Improved A* algorithm compared to weighted Manhattan A* algorithm	7.6%	16.01%	0.8%
Improved A* algorithm compared to weighted Euclidean A* algorithm	18.18%	19.05%	8.3%

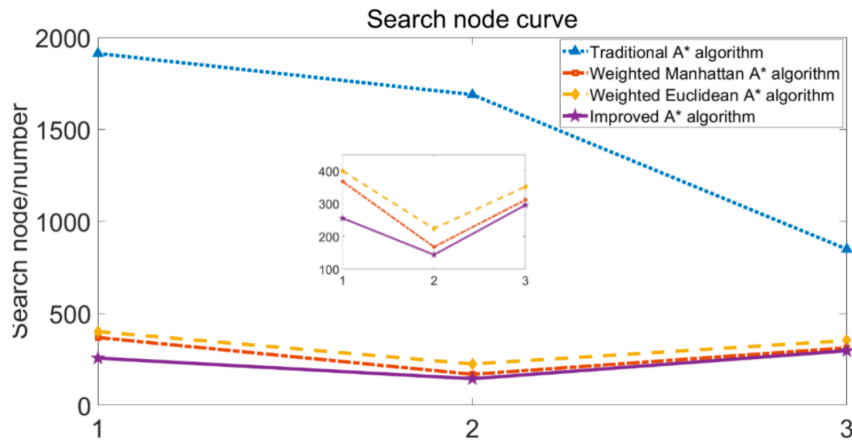


FIGURE 22. Searching node data in the same environment

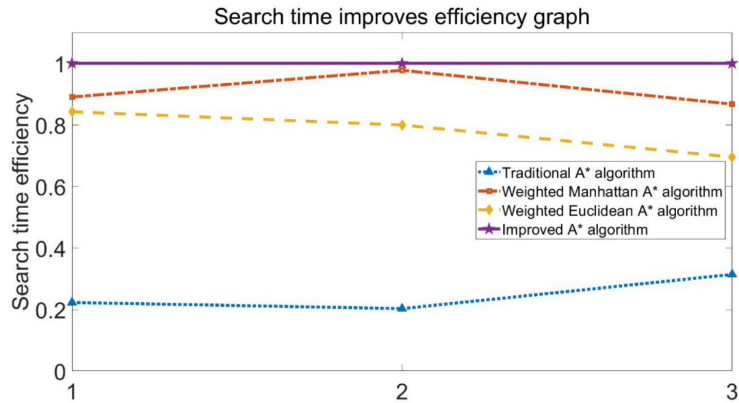


FIGURE 23. Search time improves efficiency in different environments

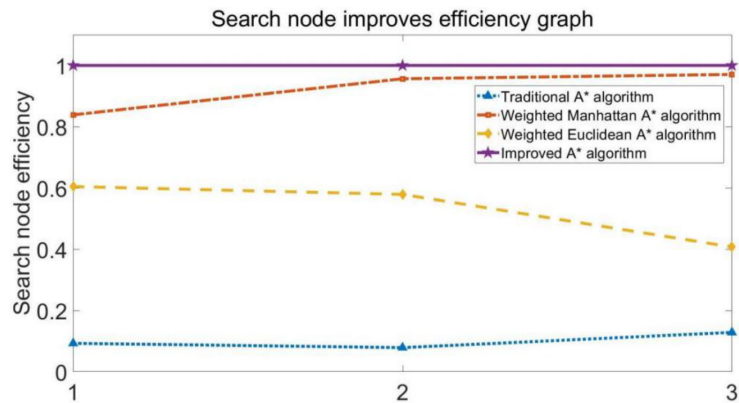


FIGURE 24. Search node improves efficiency in different environments

TABLE 10. Searching node improves efficiency data in the same environment

Number of experiments/algorithm	Experiment 1	Experiment 2	Experiment 3
Improved A* algorithm compared to traditional A* algorithm	86.61%	91.47%	65.09%
Improved A* algorithm compared to weighted Manhattan A* algorithm	30.43%	14.29%	5.23%
Improved A* algorithm compared to weighted Euclidean A* algorithm	36%	35.71%	15.91%

It could be seen from Figure 25 and Figure 26 that in terms of search time, the improved A* heuristic algorithm increases the efficiency by 64% on average compared with

the traditional A* algorithm. Compared with the weighted Euclidean A* algorithm, the efficiency is increased by an average of 15%. Compared with the weighted Manhattan, the efficiency is increased by an average of 8%. In terms of search nodes, the improved A* heuristic algorithm increases the efficiency by 81% on average compared with the traditional A* algorithm. An average increase of 29% compared with the weighted Euclidean A* algorithm. An average

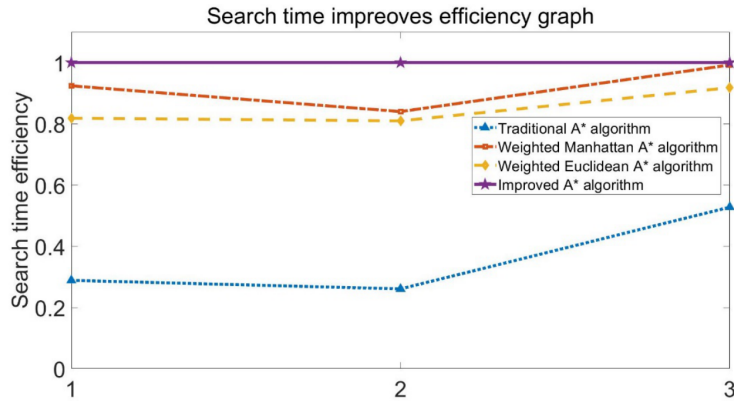


FIGURE 25. Search time improves efficiency in the same environments

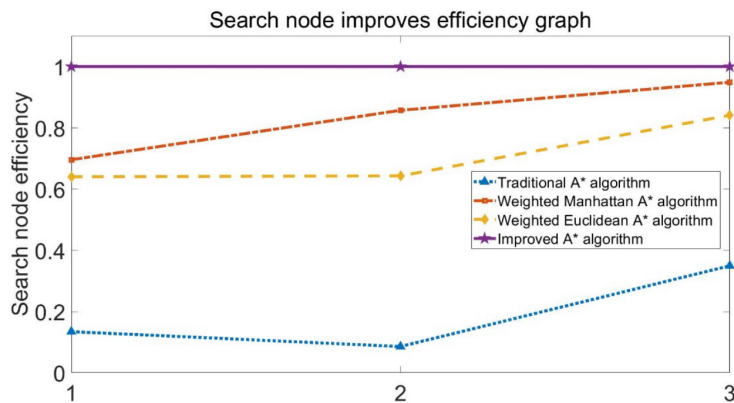


FIGURE 26. Search node improves efficiency in the same environments

increase of 16% compared with the weighted Manhattan A* algorithm. In summary, the improved A* heuristic algorithm performs better than the other three algorithms in the same experimental environment.

VI. CONCLUSION

This study proposed an enhanced path planning framework to address the limitations of the traditional A* algorithm for inspection robots navigating the complex spatial distribution of manufacturing environments. By optimizing navigation within these intricate mechanical layouts, the framework ensures the high-precision monitoring of fiber morphology and continuous manufacturing excellence. A context-aware heuristic function was introduced to dynamically guide the search process, significantly reducing the number of expanded nodes and computation time. Furthermore, a proactive safety mechanism utilizing virtual speed modulation zones was implemented around obstacles, ensuring reliable navigation by maintaining a safe distance. Finally, the discrete global path was optimized using Bézier curve interpolation, generating a smooth and kinematically feasible trajectory. Simulation results demonstrated that the proposed method markedly outperformed the classical approach, reducing search nodes and time while improving path quality, which collectively contributed to an average increase of

approximately 70% in operational efficiency, validating its effectiveness and practicality for real-world deployment.

AUTHOR CONTRIBUTIONS

Conceptualization – Chengwei Li, Yuan Zhou, Shoubin Wang, Xiaofeng Bai and Youbing Li; methodology – Chengwei Li, Shoubin Wang, Xiaofeng Bai and Youbing Li; investigation – Chengwei Li, Yuan Zhou, Shoubin Wang, Xiaofeng Bai and Youbing Li; writing-original draft preparation – Chengwei Li, Yuan Zhou, Shoubin Wang, Xiaofeng Bai and Youbing Li. All authors have read and agreed to the published version of the manuscript.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

FUNDING

This research received no external funding.

ACKNOWLEDGEMENTS

Not applicable.

REFERENCES

- [1] S. Alshammrei, S. Boubaker, and L. Kolsi, "Improved Dijkstra algorithm for mobile robot path planning and obstacle avoidance," *Comput-*

- ers *Materials & Continua*, vol. 72, no. 3, pp. 5939-5954, 2022, DOI: 10.32604/CMC.2022.028165.
- [2] S. Baik, H. J. Bong, and S. Jeong, "Weight-incorporating A* algorithm with multi-factor cost function for enhanced mobile robot path planning," *Actuators*, vol. 14, no. 8, p. 369, 2025, DOI: 10.3390/ACT14080369.
 - [3] X. Bu, G. Li, and B. Tong, "A robot navigation system based on improved A-star algorithm," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 38, no. 16, 2024, DOI: 10.1142/S0218001424560123.
 - [4] Y. Chen, Y. Liu, and W. Xu, "Improved hybrid A* algorithm based on Lemming optimization for path planning of autonomous vehicles," *Applied Sciences*, vol. 15, no. 14, p. 7734, 2025, DOI: 10.3390/APP15147734.
 - [5] Z. Chi, Z. Yu, and Q. Wei, "High-efficiency navigation of nonholonomic mobile robots based on improved hybrid A* algorithm," *Applied Sciences*, vol. 13, no. 10, 2023, DOI: 10.3390/APP13106141.
 - [6] J. Dai, Y. Chai, and P. Xiong, "Path planning of underground robots via improved A* and dynamic window approach," *Applied Sciences*, vol. 15, no. 13, p. 6953, 2025, DOI: 10.3390/APP15136953.
 - [7] Y. Fang, J. He, X. Wang, and B. Ma, "A* algorithm for on-site collaborative path planning in building construction robots," *Buildings*, vol. 15, no. 21, p. 3876, 2025, DOI: 10.3390/BUILDINGS15213876.
 - [8] H. Ge, Z. Ying, Z. Chen, and B. Zhang, "Improved A* algorithm for path planning of spherical robot considering energy consumption," *Sensors*, vol. 23, no. 16, 2023, DOI: 10.3390/S23167115.
 - [9] J. Guo, J. Yu, C. Feng, and B. Li, "Global path planning on uneven terrain at construction sites for robots based on an improved A* algorithm," *Journal of Ambient Intelligence and Smart Environments*, vol. 17, no. 3, pp. 302-325, 2025, DOI: 10.1177/18761364241296814.
 - [10] S. Huang, X. Wu, and Y. Wang, "Adaptive observation A* (AOA): Efficient deep heuristic method for global path planning in mountainous terrain," *Information Sciences*, vol. 731, Art. no. 122916, 2026, DOI: 10.1016/J.INS.2025.122916.
 - [11] R. Kabir, Y. Watanobe, and R. M. Islam, "Enhanced robot motion block of A-star algorithm for robotic path planning," *Sensors*, vol. 24, no. 5, 2024, DOI: 10.3390/S24051422.
 - [12] A. Liu and H. Liu, "Path planning of intelligent robots based on improved A* algorithm," *Journal of Physics: Conference Series*, vol. 3055, no. 1, Art. no. 012001, 2025, DOI: 10.1088/1742-6596/3055/1/012001.
 - [13] X. Liu, W. Zhou, C. Zhai, and S. Li, "Path planning and optimization of space robots on satellite surfaces based on an improved A* algorithm and B-spline curves," *Aerospace*, vol. 12, no. 10, pp. 943, 2025, DOI: 10.3390/AEROSPACE12100943.
 - [14] H. Ryu and K. W. Chung, "Local map-based exploration using a breadth-first search algorithm for mobile robots," *International Journal of Precision Engineering and Manufacturing*, vol. 16, no. 10, pp. 2073-2080, 2015, DOI: 10.1007/s12541-015-0269-9.
 - [15] W. Wu, C. Kong, Z. Xiao, and J. Xu, "3D-MIHE-RRT-A*: multi-indicator heuristic evaluation hybrid path planning algorithm for UAV navigation in complex environments," *Measurement*, vol. 260, Art. no. 119838, 2026, DOI: 10.1016/J.MEASUREMENT.2025.119838.
 - [16] X. J. Wu, P. Peng, L. Wang, and L. Feng, "A hybrid A* path planning algorithm for underground load-haul-dump vehicles guided by reference states," *Measurement Science and Technology*, vol. 36, no. 10, Art. no. 106215, 2025, DOI: 10.1088/1361-6501/AE10CE.
 - [17] X. Xing, Z. Jiazhu, Z. Yun, and S. Wei, "Research on global path planning algorithm for mobile robots based on improved A*," *Expert Systems with Applications*, vol. 243, Art. no. 122922, 2024, DOI: 10.1016/J.ESWA.2023.122922.
 - [18] Y. Yang, Z. Meng, W. Guo, and W. Li, "Full coverage path planning for multiple farming machines based on an improved A* algorithm," *Journal of Physics: Conference Series*, vol. 2991, no. 1, Art. no. 012001, 2025, DOI: 10.1088/1742-6596/2991/1/012001.
 - [19] W. Zhang, W. Li, X. Zheng, and H. Li, "Improved A* and DWA fusion algorithm based path planning for intelligent substation inspection robot," *Measurement and Control*, vol. 59, no. 1, pp. 113-119, 2026, DOI: 10.1177/00202940251316687.
 - [20] X. Zhou, J. Yan, M. Yan, and G. Wang, "Path planning of rail-mounted logistics robots based on the improved Dijkstra algorithm," *Applied Sciences*, vol. 13, no. 17, pp. 9955, 2023, DOI: 10.3390/APP13179955.
 - [21] H. Zhuang, S. Wang, N. Wang, and W. Liu, "Low-energy path planning method of electrically driven heavy-duty six-legged robot based on improved A* algorithm," *Applied Sciences*, vol. 15, no. 24, Art. no. 13113, 2025, DOI: 10.3390/APP152413113.
 - [22] D. Linjie, Z. Renfei, W. Junfei, and D. Zongcheng, "Research on the path planning algorithm and obstacle-crossing motion planning strategy for a cable trench inspection robot," *Robotica*, vol. 43, no. 2, pp. 433-448, 2024, DOI: 10.1017/S0263574724001930.