electronics

MDPI

Article

Research on Path Planning Method for Autonomous Patrol Robots

Qiang Zou [1,2,*], Haipeng Wang [3], Tianle Zhang [1], Zhengqi Li [1] and Yaoming Zhuang [1]

[1] Faculty of Robot Science and Engineering, Northeastern University, Shenyang 110819, China; 20217887@stu.neu.edu.cn (T.Z.); 20217859@stu.neu.edu.cn (Z.L.); zhuangyaoming@mail.neu.edu.cn (Y.Z.)
[2] Foshan Graduate School of Innovation, Northeastern University, Foshan 528311, China
[3] Faculty of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China; 3123154008@stu.xjtu.edu.cn
* Correspondence: zouq@mail.neu.edu.cn

Abstract: For autonomous patrol robots, how to complete multi-point path planning efficiently is a crucial challenge. To address this issue, this work proposes a practical and efficient path planning method for patrol robots. Firstly, the evaluation function of the traditional A* method is improved to ensure that the planned path maintains a safe distance from the obstacles. Secondly, a Dubins curve is used to optimize the planned path to minimize the number of turning points while adhering to kinematic constraints. Thirdly, a trajectory-preserving strategy is proposed to preserve the continuous trajectory, linking multi-points for future inspection tasks. Finally, the proposed method is validated through both simulation and real-world experiments. Experimental results demonstrate that our proposed method performs exceptionally well in terms of safety, actual trajectory distance, and total execution efficiency.

Keywords: path planning; autonomous patrol robots; improved A* algorithm; Dubins curve; trajectory preserving strategy

Citation: Zou, Q.; Wang, H.; Zhang, T.; Li, Z.; Zhuang, Y. Research on Path Planning Method for Autonomous Patrol Robots. *Electronics* **2024**, *13*, 2865. https://doi.org/10.3390/electronics13142865

Academic Editor: Felipe Jiménez

Received: 23 June 2024
Revised: 11 July 2024
Accepted: 18 July 2024
Published: 20 July 2024

1. Introduction

Patrol robots are now widely used in hospitals, parks, communities, and various other scenarios, significantly enhancing inspection efficiency, reducing operational and maintenance costs, as well as realizing intelligent inspection. Path planning is a crucial component of an autonomous patrol robot system; it aims to guide the patrol robot from its current position to the target position along an optimal, collision-free path in open, dynamic environments [1,2].

For patrol robots, global path planning and local path planning are two essential stages of navigation, each serving distinct but complementary roles in ensuring efficient and safe movement. Global path planning refers to planning a path from an initial point to a goal point across the entire map or environment; it considers global environmental information and constraints, such as map structure, obstacle positions, and target location, in order to find an optimal or feasible path. Local path planning refers to planning a path in the local vicinity of the mobile robot's current position to avoid obstacles. It primarily considers the mobile robot's kinematic constraints and focuses on the robot's immediate perception and dynamic changes in the environment to generate a real-time path that is both feasible and safe [3]. The integration of both global and local path planning is vital for the overall navigation strategy of patrol robots. They enable patrol robots to perform their tasks efficiently, covering all necessary areas while adeptly handling dynamic environments.

The most representative and widely utilized path planning algorithms in recent years include neural network algorithms [4], artificial potential field algorithms [5], rapidly expanding random tree algorithms [6], ant colony algorithms [7], and A* algorithms [8,9]. Among these methods, A* is a heuristic search method that is known for its robust and

effective global search capabilities, high efficiency, and ability to find the shortest path. Due to these advantages, the A* algorithm is extensively used in the field of path planning. Based on the A* algorithm, some methods such as D* [10], Theta* [11] and hybrid A* [12] have been proposed. These methods can generate feasible paths; however, many of the generated paths have shortcomings such as low planning efficiency, frequent collisions with obstacles, or failure to meet smoothness requirements. Consequently, to enhance the efficiency of the traditional A* algorithm in practical applications, the improvement and optimization of the A* algorithm has been a significant area of research. Tang proposed a planning algorithm that incorporates the artificial potential field method to optimize the path generated by a hybrid A* algorithm. This approach can generate a smooth path that maintains a safe distance from obstacles. But the performance of its dynamic obstacles' avoidance is inadequate in complex and dynamic environments [13]. To enhance the path planning efficiency in complex environments, a hybrid mapping-based global path planning was proposed; however, it does not guarantee the planned path is optimal path [14]. To plan a feasible path for multiple objectives, a path planning method that combines the genetic algorithm and ant colony algorithm was proposed for static environments [15]. However, the proposed method is only suitable for static environments. For mobile robots in dynamic environments, Song proposed a dynamic path planning method to plan a smooth path; however, its efficiency still needs improvement [16]. Yonetani proposed an improved A* algorithm based on a neural network; this method uses the neural network to learn the environmental features for much more accurate heuristic function, and it can achieve dynamic path planning in dynamic environments [17]. Liu improved planning efficiency by selecting jumping points as well as compulsory neighbors as key points and eliminating redundant points [18]. Sun improved the planning efficiency of the A* algorithm by optimizing the heuristic function and improving the eight-neighbor search strategy [19]. Jiang proposed an improved bidirectional A* algorithm that enhances both the arch direction and heuristic function, reducing the average path distance and planning time. However, this method does not take safety distance into account [20]. The Dynamic Window Approach (DWA) [21] and Time Elastic Band (TEB) [22] methods have good obstacle avoidance ability in dynamic environments. Focusing on optimizing the efficiency of global path planning and the quality of the generated path, Niu proposed a global dynamic path planning method based on an improved A* algorithm and the dynamic window method [23]. Li used the adaptive adjustment step method and a three-time Bezier curve to minimize the number of turning points, and then integrated the improved A* algorithm with the DWA method to avoid dynamic obstacles and falling into local optimization [24].

For autonomous patrol robots, how to complete multi-point path planning efficiently is a crucial challenge. The most crucial metrics for a path planning method is its feasibility and capability of satisfying the robot's kinematic constraints, as well as prevent the patrol robot from colliding with other obstacles in the environment [25]. Although many research works have focused on finding a feasible global path for navigation, the global path is often optimized for overcoming the following disadvantages: (1) the path is too close to the obstacles; (2) the path consists of lots of turning points and redundant nodes; (3) the path does not satisfy the kinematic characteristics; and (4) low planning efficiency. Moreover, these above works rarely involve research on multi-point planning; they cannot be well deployed for real patrol robots. With the goal of achieving an optimal path and high planning efficiency, this work proposes an efficient and practical path planning method for autonomous patrol robots. Compared with the existing work, the main contributions are listed as follows:

(1) This work proposes an inspection system architecture for autonomous patrol robots, which can be deployed in a real robot to execute inspection tasks;
(2) To overcome the shortcomings of nearness to obstacles, this work introduces a distance value, and optimizes the traditional A* algorithm's evaluation function, which can keep the generated global path a safe distance from obstacles;

(3) To reduce the turning points and maintain kinematic constraints, this work optimizes the global path using Dubins curves, which can improve patrol efficiency while maintaining safety;
(4) This work proposes a trajectory-preserving strategy as well as a smoothing strategy near inspection points, which can preserve the continuous trajectory that satisfies the kinematic constraints, and the preserved trajectory can be reloaded for future inspection tasks.

The overall structure of this work is as follows: Section 2 introduces the system architecture of an autonomous patrol robot. Section 3 describes the map construction method. Section 4 describes the path planning methods. Section 5 reports the experiments and results. And, finally, the conclusion of the entire work is given in Section 6.

2. System Architecture of Autonomous Patrol Robots

2.1. Problem Statement

This work proposes an efficient and practical multi-point path planning method for autonomous robots; it aims to enhance the inspection capability and generalization ability of patrol robots in open, dynamic environments. The input of this method is Lidar point cloud and IMU sensor data, and these input data are used for mapping the environment. Then, the built map is used for self-localization and global path planning; its output is a continuous-curvature and collision-free global path with coordinates and orientations. Finally, the local path planning method works; its output is the motion command for controlling the patrol robot, including linear velocity and angular velocity.

2.2. System Architecture

In this work, an ROS (robot operating system) framework is utilized for the development of the autonomous patrol robots' system. The use of the ROS provides a modular and flexible architecture, allowing for the seamless integration of the key components required for the patrol robots' operation. Patrol robots often carry out inspection tasks in dynamic environments. The system architecture of patrol robots is shown in Figure 1; it has two key function modules: map construction and autonomous path planning.

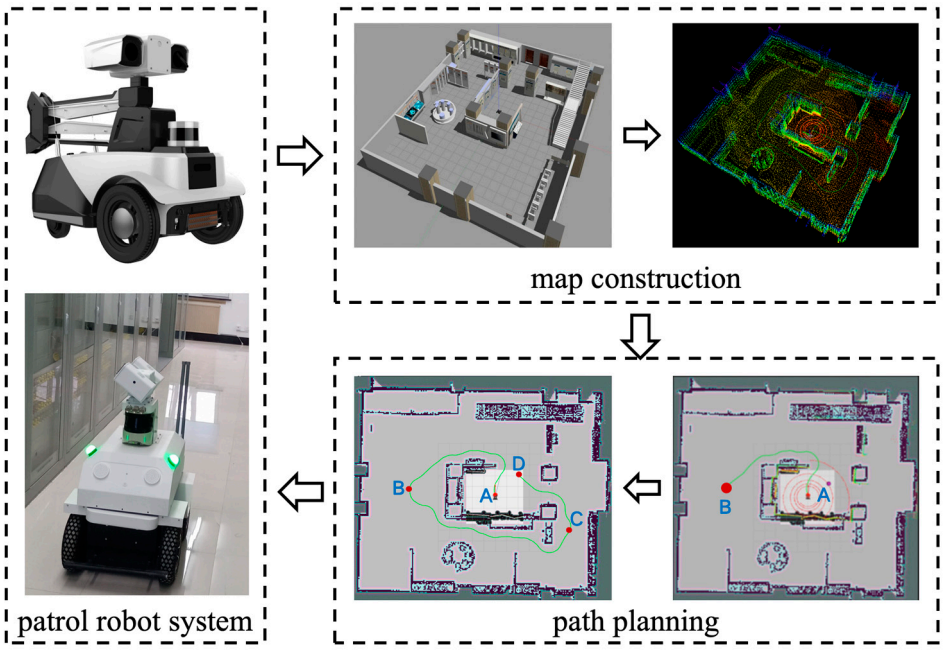


Figure 1. The main functional modules of patrol robots.

Map construction plays a crucial role in robotic self-localization and autonomous path planning. Point cloud maps provide detailed information about the surrounding

environment. By comparing the real-time sensor data with the pre-built point cloud map, the robot can estimate its position and orientation accurately, as well as identify obstacles and plan its paths accordingly, ensuring collision-free navigation. Real-time updates to the point cloud map can also help robots adapt to dynamic environments and avoid newly introduced obstacles. For the reason that multi-line Lidar has the advantages of high precision and strong robustness, in this work the patrol robot is equipped with 16-line Lidar and uses it to construct the point cloud map of the dynamic environment.

Path planning in this work is also divided into the planning of the global path and the local path. Global planning involves determining the optimal route from the robot's current position to its desired destination. It takes into account the overall environment, including obstacles, boundaries, and any relevant constraints. The goal of global path planning is to find a high-level path that minimizes distance, avoids obstacles, and satisfies any specified criteria or objectives. Once the global path is determined, local path planning comes into play. Local planning focuses on the robot's immediate surroundings and deals with fine-grained navigation within the global path. It takes into account the robot's current position, sensor readings, and dynamic obstacles to make real-time adjustments and avoid collisions. Local path planning ensures that the robot can navigate smoothly and safely along the global path, making necessary corrections and adaptations as needed. By dividing the path planning process into global and local steps, the patrol robot can make intelligent decisions efficiently and navigate autonomously in challenging and changing environments.

The above two key function modules empower the patrol robot with comprehensive capabilities. It can autonomously navigate in various scenarios, ranging from structured indoor environments to unstructured outdoor environments, efficiently patrol designated areas, adapt to dynamic changes, and enhance surveillance.

3. Map Construction Method

Due to the fact that research of autonomous path planning for patrol robots is mainly applied in two-dimensional environments, this work constructs point cloud maps of the patrol environments firstly, and then converts the point cloud maps into grid maps for patrol path planning. The whole map construction process is shown in Figure 2.

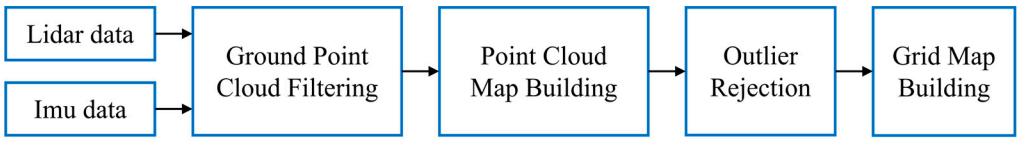
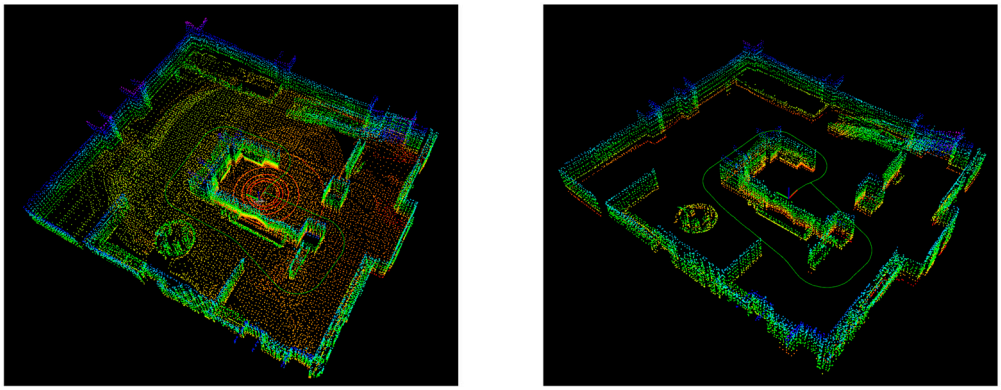


Figure 2. The whole map construction process.

Currently, there are many algorithms for building a point cloud map, such as A-LOAM [26], Lego-LOAM [27], LIO-SAM [28], FAST-LIO [29], Faster-LIO [30], etc. Because patrol robots often work in open, large-scale environments, ground point cloud data typically exhibit the characteristics of demanding a large data volume and relatively limited information content. Through filtering out the ground points, the computational load and memory requirements for processing point cloud data can be reduced, and the remaining point cloud data primarily represent objects, obstacles, and structures above the ground level, which can enable faster and more efficient analysis and improve the reliability and accuracy of the point cloud map. This work used the Ground Plane Fitting algorithm [31] to filter out the ground points. As shown in Figure 3, the Ground Plane Fitting algorithm can remove the ground points from the original point cloud data while retaining the feature information in the environment, which can reduce the pressure of point cloud map building as well as maintain the mapping quality.



(a) Before ground point cloud filtering (b) After ground point cloud filtering

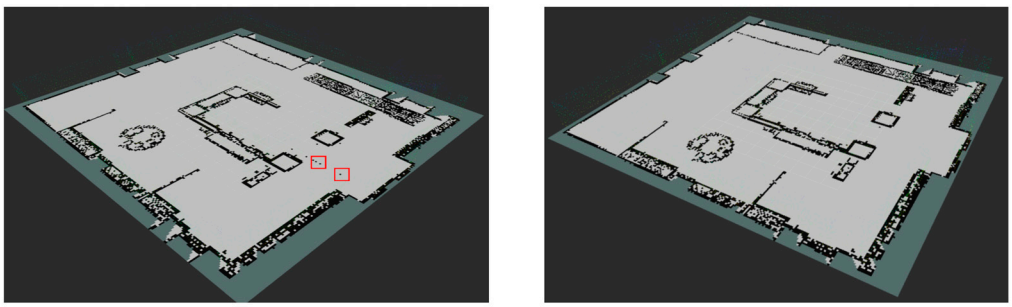
Figure 3. Schematic diagram of ground point cloud filtering.

After filtering the ground point cloud, this work uses the OctoMap tool [32] to convert three-dimensional point cloud map to two-dimensional grid map. Due to data acquisition errors, occlusions, sensor noise, and some other abnormal situations, some points exist that are significantly different or deviate from other points in the point cloud map, called outliers. Outliers would lead to some disruptions and interference in the map building process. If the point cloud map is converted to a grid map directly without outlier rejection, some outliers would be regarded as obstacles, which results in some grid cells that are actually unoccupied being marked as occupied in the grid map. As shown in Figure 4a, the grid cells that are marked with red rectangular boxes are outliers. To solve the problems that outliers bring, this work adopts statistical filtering methods to remove outliers. For the point p_i, the average distance between it and its k-nearest neighbor points can be computed as show in (1). The mean $\overline{u}$ and standard deviation σ for N points can be computed as shown in (2) and (3), respectively.

$$u_i = \sum_{j=1}^{k} \left\| p_i - p_{ij} \right\| / k \tag{1}$$

$$\overline{u} = \sum_{i=1}^{N} u_i / N \tag{2}$$

$$\sigma = \sqrt{\sum_{i=1}^{N} (u_i - \overline{u})^2 / N} \tag{3}$$



(a) Before outlier rejection (b) After outlier rejection

Figure 4. Schematic diagram of outlier rejection.

According to the mean and standard deviation, this work set one threshold $\vartheta = \overline{u} + \alpha\sigma$, where α is a tunable parameter. If the average distance u_i is larger than ϑ, the point p_i is regarded as an outlies, and it will be removed from the point cloud map. After outlier rejection, the grid map building result is shown in Figure 4b, and the generated grid map can be effectively utilized for future planning.

4. Path Planning Methods

Based on the above built grid map, this section primarily introduces the global path planning method for autonomous patrol robots.

As a classic algorithm, the traditional A* algorithm is widely used in field such as gaming due to its simplicity in implementation and its ability to find the shortest path. Figure 5 shows the result of global path generated by the traditional A* algorithm, which shows that the generated path maintains a very close distance from obstacles and has multiple instances of touching the obstacles' corners and edges. For the robot's inspection applications within large environments, the closer the distance to obstacles, the higher the probability of collision with obstacles. Consequently, the traditional A* algorithm is unable to satisfy the patrol robots' obstacle avoidance requirements in real-world scenarios. It has some deficiencies, such as large numbers of redundant nodes and turning points, as well as insufficient path smoothness.

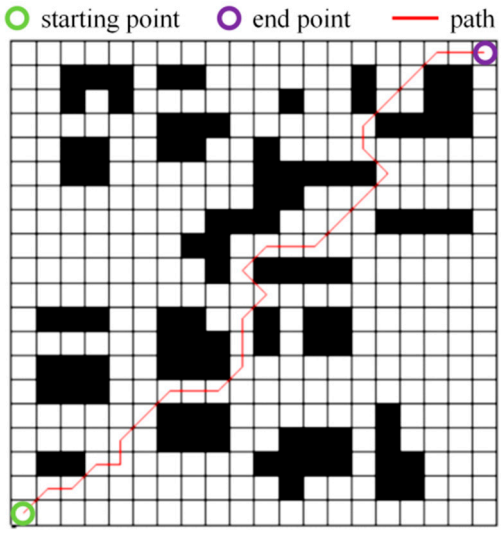


Figure 5. Example of path generated by traditional A* algorithm.

For the path planning problems described above, some improvements are made in the following subsections.

4.1. Global Path Planning

To maintain a safe distance between the generated path and obstacles, this work introduces an obstacle distance value, and optimizes the evaluation function of the traditional A* algorithm. The improved evaluation function can be described as (4).

$$f(n) = g(n) + h(n) + b(n) \tag{4}$$

where $f(n)$ is the estimated value of node n, $g(n)$ is the actual value from initial node to node n, $h(n)$ is the estimated value from node n to target node, and $b(n)$ is the distance value from node n to the nearest obstacles; $b(n)$ is represented as shown in (5).

$$b(n) = -k \cdot d \tag{5}$$

where d is the Hamming distance from node n to the nearest obstacles, and k is a coefficient that needs to adjust based on the actual situation. The larger the value of this coefficient, the greater the distance impact in the traditional A* algorithm's heuristic function, and the generated path will be farther away from the obstacles. However, if this coefficient is too large, it may also lead to the failure of path planning. With the improved evaluation function, the generated global path maintains a safe distance from the obstacles; therefore, the probability of collision with obstacles during robot navigation will be greatly reduced.

In addition, to minimize the number of redundant nodes and turning points, this work utilized a Dubins curve to smooth the global safe path generated by our proposed improved A* algorithm, which can allow the patrol robots to improve patrol efficiency while maintaining safety and maintaining kinematic constraints.

A Dubins curve consists of a sequence of circular arcs and straight-line segments; they are commonly used to model the shortest path between two points while considering the vehicle's kinematic constraints. The path optimization principle of the Dubins curve in this work is shown in Figure 6. This optimization method no longer searches for neighboring nodes around the current node, but instead, based on the position of the patrol robot at a particular node and considering the patrol robot's kinematic constraints, it searches for points around it. It is obvious that the optimized path satisfies the kinematic constraints of the patrol robot and ensures the patrol robot's motion capabilities.

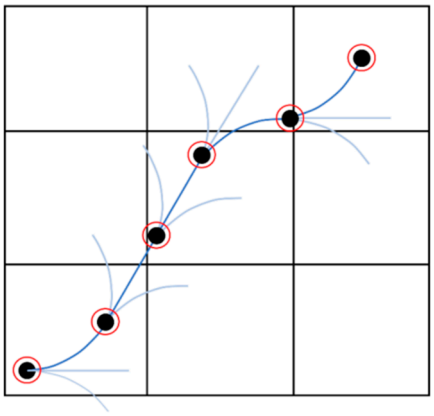


Figure 6. Path optimization principle of Dubins curve.

As shown in Figure 7, assuming that the position and orientation of the robot in the world coordinate system is w_{p_0} and w_{q_0}, p_1, p_2 and p_3 are the points that are obtained through a heuristic search using a Dubins curve at the current position of the patrol robot, and their position R_{p_i} $(i = 1, 2, 3)$ and orientation R_{q_i} in the robot coordinate system can be represented as shown in (6) and (7).

$$\left[R_{p_1} R_{p_2} R_{p_3}\right] = \begin{bmatrix} \|v\| \cdot t & r \cdot \sin wt & r \cdot \sin wt \\ 0 & r \cdot (1 - \cos wt) & -r \cdot (1 - \cos wt) \\ 0 & 0 & 0 \end{bmatrix} \tag{6}$$

$$\left[R_{q_1} R_{q_2} R_{q_3}\right] = \begin{bmatrix} 1 & \cos\frac{\theta}{2} & \cos\frac{\theta}{2} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \sin\frac{\theta}{2} & -\sin\frac{\theta}{2} \end{bmatrix} \tag{7}$$

In this way, the position of these three points in the world coordinate system can be represented as shown in (8).

$$w_{p_i} = {}^{w}_{R}\boldsymbol{T} R_{p_i} \tag{8}$$

where ${}^{w}_{R}\boldsymbol{T}$ can be computed by w_{p_0} and w_{q_0}. After obtaining w_{p_i}, it can be inferred whether obstacles exist near the position of point p_i and, if they exist, the point is neglected; otherwise, the orientation of this point is calculated as shown in (9), and then this point is regarded as the start point, and the next point is continuously selected until the target point is reached.

$$w_{q_i} = w_{q_0} \times R_{q_i} \tag{9}$$

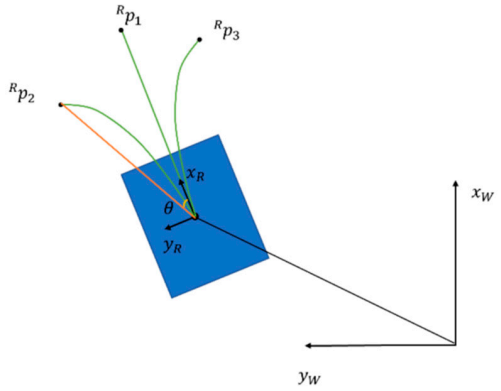


Figure 7. Schematic diagram of robotic position.

4.2. Local Path Planning

Time Elastic Band (TEB) is a local path planning method. It combines time optimization and an elastic band concept to generate smooth and efficient trajectories for mobile robots, especially in dynamic environments. Therefore, this work chooses TEB as the basic local planning algorithm. TEB represents the mobile robot's motion trajectory as a time elastic band, and the elastic band consists of two boundary lines, representing the feasible and safe motion region respectively. By adjusting the position and shape of these boundary lines, different trajectories can be generated.

Assuming that the elastic band is $(x_{min}(t),\ x_{max}(t))$, the motion trajectory of the robot must lay in the elastic band. So, the motion trajectory $(x_k, y_k,\ \theta_k)$ can be represented as shown in (10). To ensure safety, this work represents the patrol robot's safe region as a circular area, and the radius of the circle is r_{safe}; thus, the safe region of patrol robot can be described as shown in (11).

$$\begin{cases} x_k \in [x_{min}(t + k\Delta t),\ x_{max}(t + k\Delta t)] \\ y_k \in [y_{min}(t + k\Delta t),\ y_{max}(t + k\Delta t)] \\ \theta_k \in [\theta_{min}(t + k\Delta t),\ \theta_{max}(t + k\Delta t)] \end{cases} \tag{10}$$

$$(x_k - x_i)^2 + (y_k - y_i)^2 \geq r_{safe}^2 \tag{11}$$

where k is the time step and $(x_i,\ y_i)$ is the position of the ith obstacle. In this way, the motion trajectory can be optimized and planned.

4.3. Trajectory Preservation

For a patrol robot, its operating mode is wandering along one pre-planned trajectory that covers all inspection points, rather than different trajectories every time. Considering this factor, this work proposes to preserve the patrol trajectory, which can then be loaded for further inspection tasks. If some places in the patrol trajectory are inevitably impassable, such as road maintenance and occupation, the patrol trajectory will be planned again.

The preserved trajectory can be regarded as a set of planned trajectories among these adjacent inspection points. And the above planning method can plan one smooth and safe trajectory between two inspection points. However, a bias exists between the actual position that the patrol robot arrived inand the planned inspection point, which results in the trajectory near to the inspection point not being smooth. As shown in Figure 8, p_l is the last point belonging to path l_1 that was generated by our proposed method, p_t is the ending of path l_1, and p_r is the actual position that the patrol robot arrived in, corresponding to point p_t.

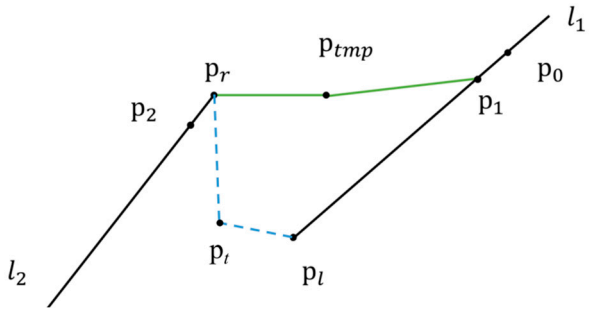


Figure 8. Trajectory smoothing nearing to inspection point.

To smooth the trajectory that nears to these inspection points, this work inserts one point p_{tmp} between p_r and l_1, and selects one point p_1 in l_1, ensuring the trajectory that links p_1, p_{tmp}, and p_r satisfies the kinematic constraints. In Figure 8, p_2 is the next point of p_r, and p_0 is the previous point of p_1. Point p_1 can be selected as shown in (12).

$$F(p_1) = \min_{p_{tmp} \in R^2,\ p_1 \in l_1} (|r_1 - r_{radius}| + |r_2 - r_{radius}|) \quad r_1 \geq r_{radius} \quad r_2 \geq r_{radius} \tag{12}$$

where r_{radius} is the rotation radius of the patrol robot and r_1 is the radius of the circle determined by points p_2, p_r, and p_{tmp}; similarly, r_2 is the radius of the circle determined by points p_0, p_1, and p_{tmp}. After selecting point p_1, the trajectory after point p_1 in l_1 should be deleted, and the trajectory linking p_1, p_{tmp}, and p_r is the optimized trajectory that nears to point p_l. In this way, the preserved patrol trajectory maintains smoothness and satisfies the kinematic constraints of the patrol robot, and can be used for further inspection tasks.

5. Experiments and Results

To evaluate the performance and practicality of our proposed methods, some simulation and real-world experiments were conducted.

5.1. Simulation Experiments

The simulation experiments were implemented in Gazebo. Gazebo is a simulation environment that allows developers to accurately and effectively simulate robotic systems, sensors, objects, and environments, providing a platform for developing and testing algorithms in a realistic setting without the need for physical hardware. In the simulation environments, a museum test environment was set up with a configuration of a 22.5 m $\times$ 22.5 m grid with a 0.1 m $\times$ 0.1 m resolution. An Ackermann mobile robot equipped with a 16-line Lidar and IMU sensors was constructed. For the simulation experiment, A-LOAM algorithm was chosen to build the point cloud map. Figure 3b shows the built point cloud map after filtering the ground point cloud, and Figure 4b shows the result of grid map building, which was used for the patrol trajectory planning later.

To evaluate the effectiveness of our proposed path planning method, planning experiments with the traditional A* method, RRT method, as well as our proposed method were carried out. The comparison results are shown in Figure 9. This work set three inspection points (B, C, and D) in the simulation environment; point A is the initial point of the patrol robot, and point D is the ending point. Every method has planned three paths; they are A-B, B-C, and C-D, respectively. The green solid lines are the planned paths. From the comparison results, it can be seen that the paths planned by our method maintain a moderate distance from the obstacles, and the optimization of the Dubins curve can make the patrol paths much smoother, which can satisfy the kinematic constraints of the mobile robot.

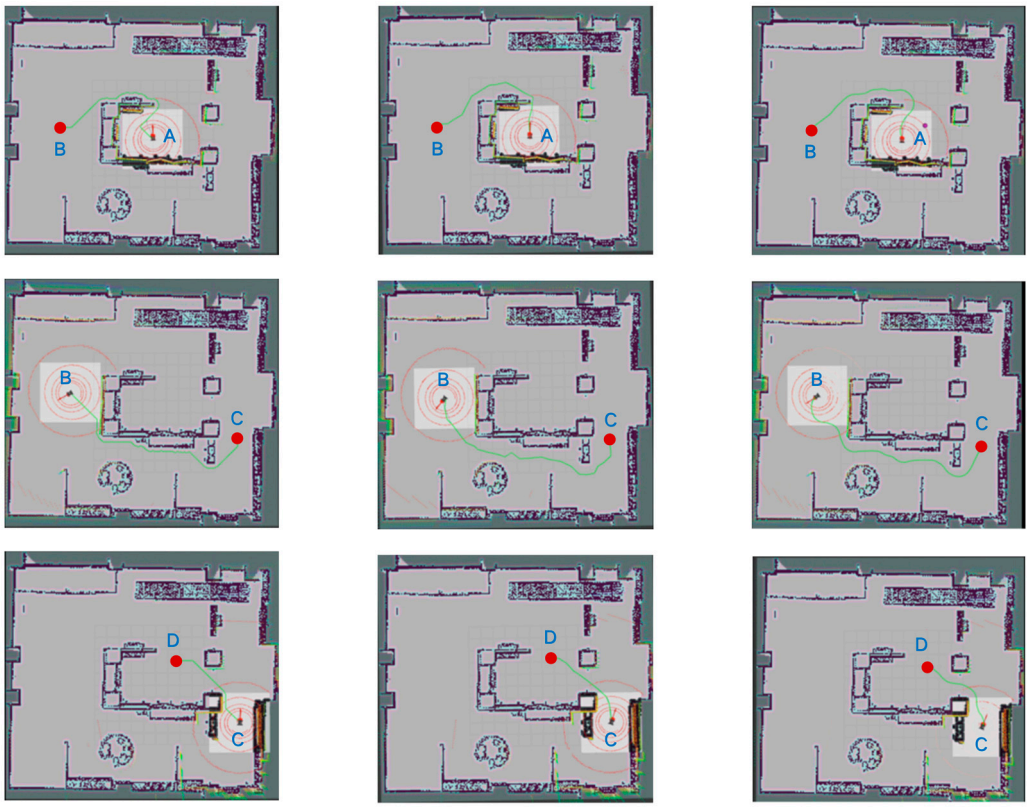


Figure 9. Results of global path planning in simulation environment.

After obtaining these trajectories among these inspection points, the trajectory preserve module links these different trajectories sequentially and smooths the trajectories nearing the inspection points, and finally preserves the whole patrol trajectories. Based on the role of the proposed trajectory preserve module, the whole patrol trajectories from A-B-C-D are generated, as shown in Figure 10. According to the whole patrol trajectories, the patrol robot can execute the inspection task using the local path planning algorithm, and Figure 11 describes the entire patrol process.

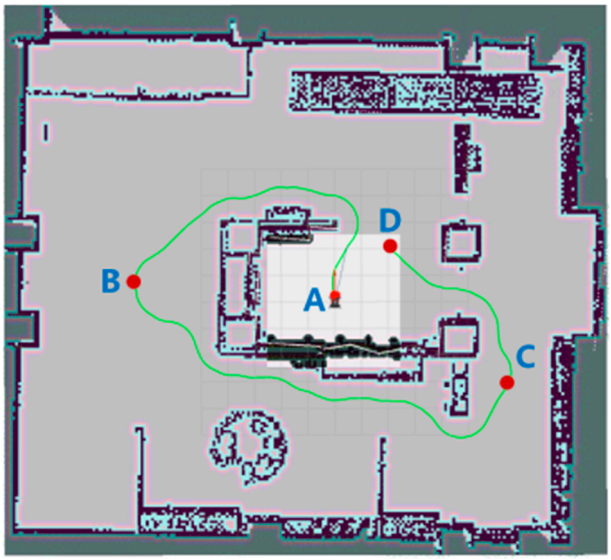


Figure 10. The preserved whole trajectories from A to D.

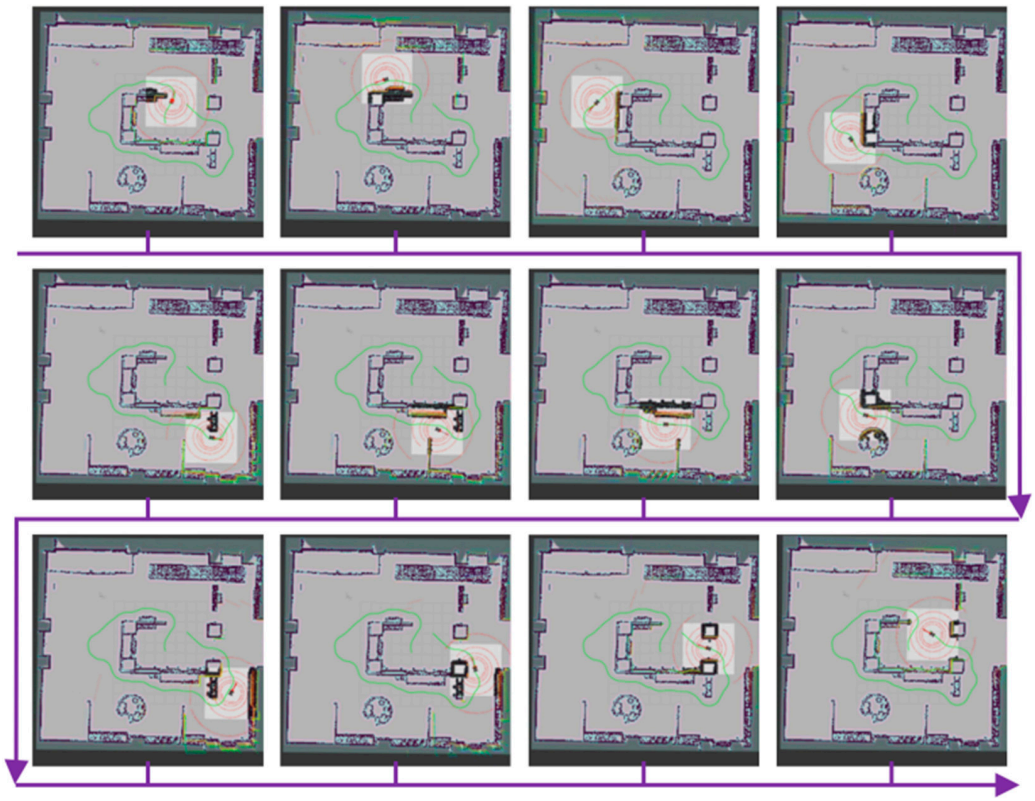


Figure 11. The entire patrol process in simulation environment using our method.

To evaluate the performance of our proposed method, we used different global path planning methods (traditional A*, RRT, and our method) but the same local path planning method to execute the same inspection task. The comparative experiment had the same maximum velocity set, and the linear and angular velocity were recorded by subscribing the topic of "/ackerman_steering_controller/cmd_vel/linear/x" and "ackerman_steering_controller/cmd_vel/angular/z" respectively during the whole patrol process. The results of linear and angular velocity changes along the planned path are shown in Figure 12. It is obvious that when the mobile robot moves along the global path generated by our method, the linear velocity does not exhibits serious vibration; however, in the paths planned by the traditional A* and RRT methods, the linear velocity exhibits significant vibration, which is marked using black rectangle boxes in Figure 12a,b. In addition, the angular acceleration in the path planned by the traditional A* method is occasionally too large, as shown in Figure 12a, which is marked using green rectangle boxes. These significant vibrations and the large angular acceleration would cause serious shaking of the patrol robot, which are not favorable for the long-term use of the patrol robot. Therefore, compared with the traditional A* and RRT methods, using the improved A* method and Dubins curve to plan the global path can alleviate the shaking of the patrol robot during navigation, and even improve the usage duration of the robot.

According to the planned path, the mobile robot was allowed to execute inspection tasks in this simulation environment five times. The results of patrol distance and patrol time for each time are shown in Figure 13. The average patrol distance and time are shown in Table 1. The comparison results show that the path planned by our method can guide the mobile robot to execute the inspection task with a shorter distance and in less time. Furthermore, under the global path generated by our method, the mobile robot's navigation performance is very stable, without a large change in the patrol distance or patrol time.

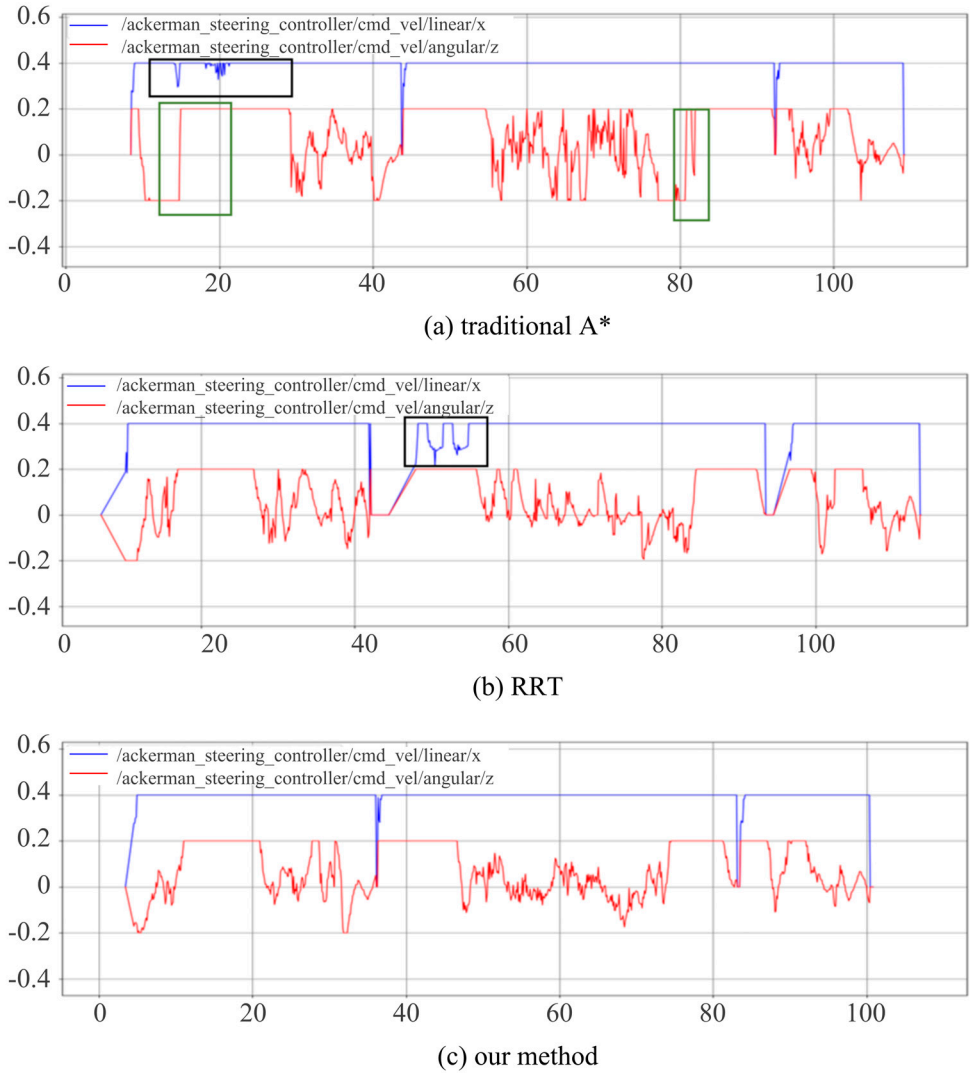


Figure 12. Changes in linear and angular velocity in simulation environment.

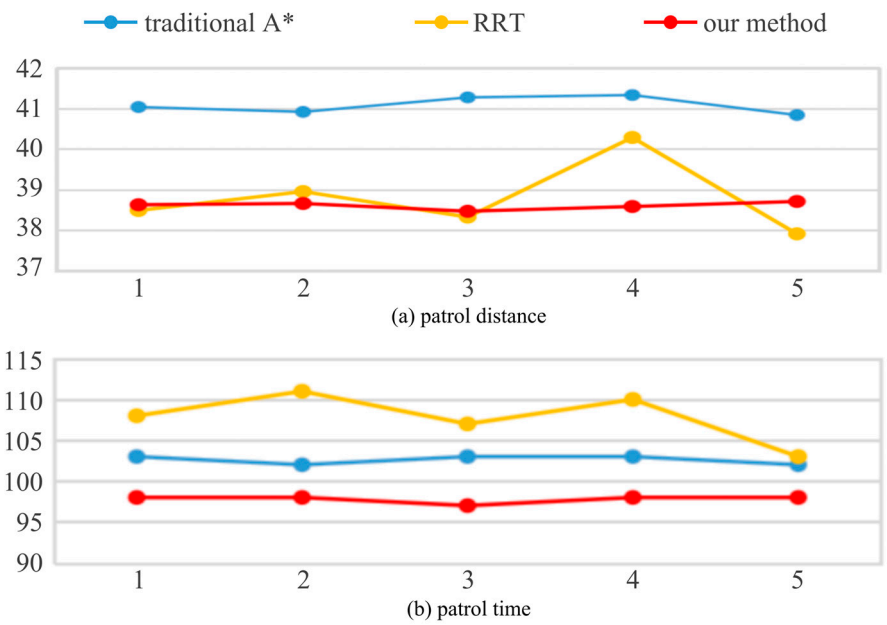


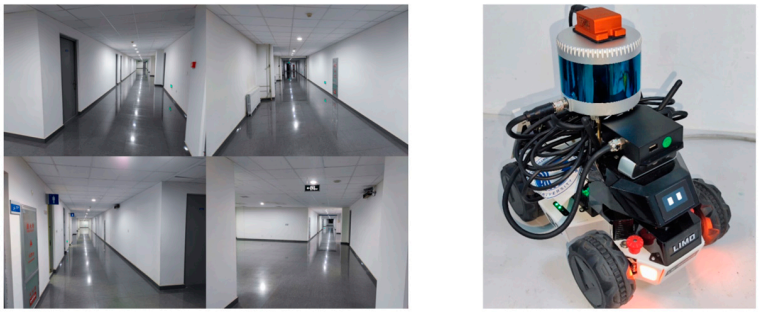
Figure 13. Comparison of patrol distance and patrol time.

Table 1. Comparison of average patrol distance and time.

Methods	Average Patrol Distance (/Meter)	Average Patrol Time (/Second)
traditional A*	41.082	103
RRT	38.789	108
our method	38.606	98

5.2. Real-World Experiments

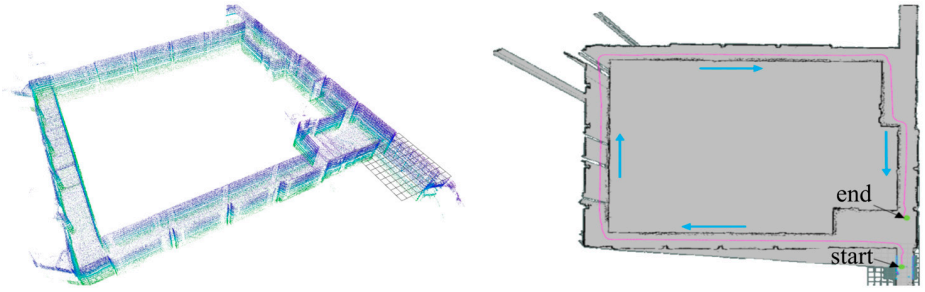
To validate the feasibility and robustness of the proposed methods on a real robot, indoor patrol experiments were conducted. The environment is a rectangular corridor, its perimeter is around 200 m, and the feasible region's width is around 2.5 m. The experimental scenes are shown in Figure 14a,b showing the real indoor patrol robot system. The patrol robot system includes a LIMO mobile robot that made by Agilex Robotics in Shenzhen China, a 16-line Lidar that made by LSLIDAR in Shenzhen China, an Xsens MTI-300 IMU sensor that made by Movella Inc. in Henderson United States, and a Jetson Nano (4G) controller that made by NVIDIA, as well as some accessories. The Jetson Nano is the mobile robot's computing core; it deploys the ROS operating system and interfaces with the mobile robot. The Jetson Nano collects data from the Lidar and IMU sensors for constructing the point cloud map, as well as performs the proposed algorithm for planning the global path and controlling the mobile robot.



(a) experimental scenes (b) indoor patrol robot

Figure 14. Corridor scenes and indoor patrol robot system.

Firstly, the patrol robot builds the point cloud map of the corridor environment using the Faster-LIO algorithm—the result of point cloud map building is shown in Figure 15a—and then the point cloud map is converted into the grid map, as shown in Figure 15b. Due to the existence of glass windows, the grid map building result is not entirely accurate; however, the grid map for the inspection area is correct. Therefore, this experiment did not try to optimize the grid map.



(a) Point cloud map (b) Grid map and patrol path

Figure 15. Results of map building and patrol path planning for corridor environment.

Secondly, some inspection points are set in the grid map, and then the proposed path planning method generates the global patrol path, which is represented by the magenta solid line in Figure 15b. The results also indicate that the improved A* algorithm and Dubins curve optimization can keep the patrol path a safe distance from the corridor walls, as well as ensure it meets the kinematic constraints of the indoor patrol robot.

Finally, according to the generated global patrol path, the local planning method guides the patrol robot to execute the inspection task, and the patrol process is shown in Figure 16. The patrol robot can move along the planned path and arrive at the end inspection point successfully; in total, it takes around 276 s for one inspection task.

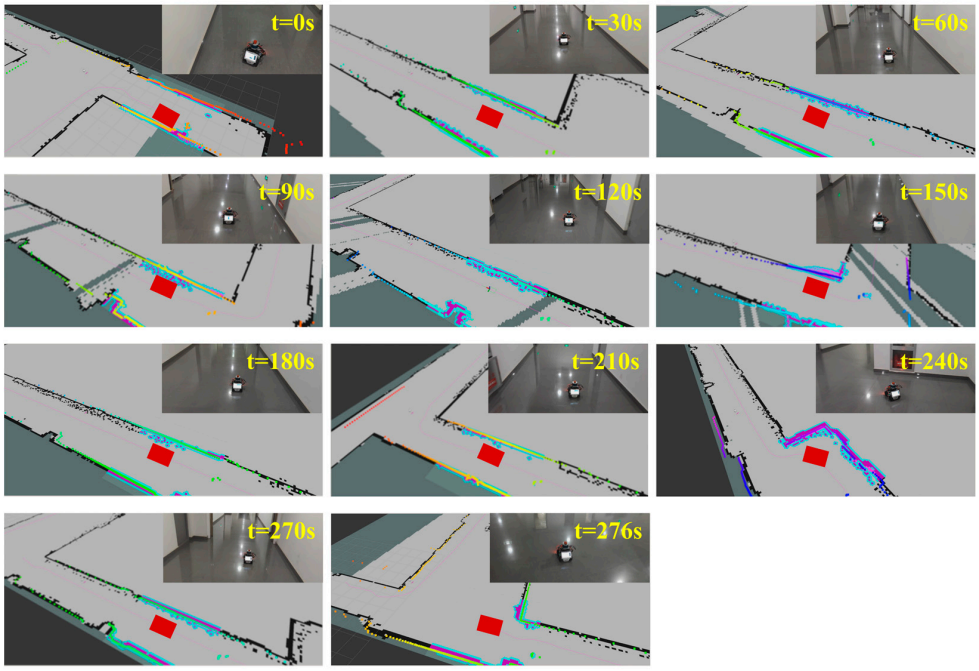


Figure 16. Patrol process in the corridor environment.

In addition, to evaluate the obstacle avoidance capability of the indoor patrol robot, an obstacle was placed on the patrol path. Figure 17 shows the process of obstacle avoidance. It is obvious that when the patrol robot detected an obstacle on its original planned trajectory, the robot would adjust its movement velocity in real-time to avoid the obstacle, and after avoiding the obstacles, it returned to its original motion trajectory again for the inspection task.

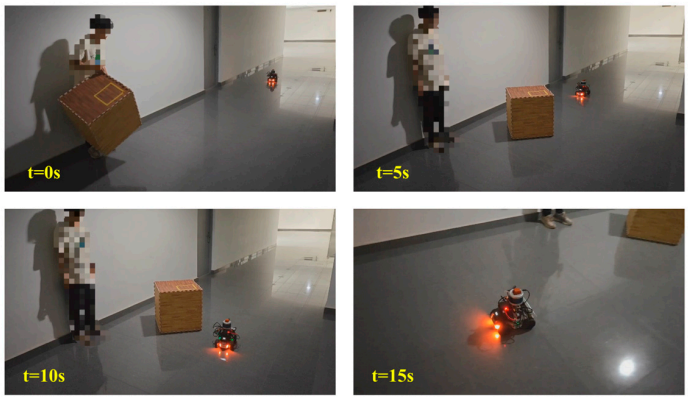


Figure 17. Process of obstacle avoidance.

Both the simulation and real-world experiments reveal the feasibility and performance of our proposed methods, such as the planned patrol paths maintaining a moderate distance

from obstacles and satisfying the kinematic constraints of the patrol robot, inspection task execution occurring with a shorter distance and in less time, as well as the possibility of deployment in real robots.

6. Conclusions

For autonomous patrol robots, efficiently completing multi-point path planning is a significant challenge. To address this issue, we propose an inspection system architecture for patrol robots that can be deployed in real robots to efficiently plan a feasible global path and execute inspection tasks. The architecture first introduces an obstacle distance value and optimizes the evaluation function of the traditional A* algorithm to ensure the planned path maintains a safe distance from obstacles. The global path is then optimized using a Dubins curve to reduce turning points, maintain kinematic constraints, and enhance patrol efficiency. Finally, we propose a trajectory-preserving strategy and a smoothing strategy near inspection points. These strategies maintain continuous trajectories that satisfy kinematic constraints and can be reloaded for future inspection tasks.

The proposed system has been validated in both simulation and real-world environments. Experimental results show that the global path planned by our method maintains a safe distance from obstacles and the kinematic constraints of patrol robots. Our method enables patrol robots to execute inspection tasks more efficiently, with shorter distances and in less time. The results demonstrate the superior performance of our inspection system in terms of safety, trajectory distance, and overall execution efficiency, making it suitable for deployment in real patrol robots.

The research findings of this work have significant implications for the long-term application of patrol robots. In future work, the proposed method will be tested and enhanced in campus environments, and empower the patrol robots with the capability for long-term autonomous execution of inspection tasks in large-scale open environments.

Author Contributions: Conceptualization, Q.Z. and H.W.; methodology, Q.Z. and H.W.; software, H.W., T.Z. and Z.L.; validation, H.W., T.Z. and Z.L.; investigation, Q.Z., H.W. and Y.Z.; writing—original draft preparation, Q.Z. and H.W.; writing—review and editing, Q.Z. and Y.Z.; supervision, Q.Z.; project administration, Q.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the China Postdoctoral Science Foundation (2023M740541), Guangdong Basic and Applied Basic Research Foundation (2023A1515110544), Liaoning Natural Science Foundation (2023-BSBA-120), and the Fundamental Research Funds for the Central Universities (N2326005).

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Mohanan, M.G.; Salgoankar, A. A survey of robotic motion planning in dynamic environments. *Robot. Auton. Syst.* **2018**, *100*, 171–185. [CrossRef]
2. Jian, Z.; Zhang, S.; Chen, S.; Nan, Z.; Zheng, N. A global-local coupling two-stage path planning method for mobile robots. *IEEE Robot. Automat. Lett.* **2021**, *6*, 5349–5356. [CrossRef]
3. Gomez, C.; Hernandez, A.C.; Crespo, J.; Barber, R. A topological navigation system for indoor environments based on perception events. *Int. J. Adv. Robot. Syst.* **2016**, *14*, 1–12. [CrossRef]
4. Chen, M.; Zhu, D. Optimal time-consuming path planning for autonomous underwater vehicles based on a dynamic neural network model in ocean current environments. *IEEE Trans. Veh. Technol.* **2020**, *69*, 14401–14412. [CrossRef]
5. Park, S.O.; Lee, M.C.; Kim, J. Trajectory planning with collision avoidance for redundant robots using Jacobian and artificial potential field-based real-time inverse kinematics. *Int. J. Control Autom.* **2020**, *18*, 2095–2107. [CrossRef]
6. Hu, B.; Cao, Z.; Zhou, M. An efficient RRT-based framework for planning short and smooth wheeled robot motion under kinodynamic constraints. *IEEE Trans. Ind. Electron.* **2021**, *68*, 3292–3302. [CrossRef]
7. Wu, C.; Zhou, S.; Xiao, L. Dynamic path planning based on improved ant colony algorithm in traffic congestion. *IEEE Access* **2020**, *8*, 180773–180783. [CrossRef]

8. Liu, C.; Mao, Q.; Chu, X.; Xie, S. An improved A-star algorithm considering water current, traffic separation and berthing for vessel path planning. *Appl. Sci.* **2019**, *9*, 1057. [CrossRef]
9. Shang, E.; Dai, B.; Nie, Y.; Zhu, Q.; Xiao, L.; Zhao, D. An improved A-star based path planning algorithm for autonomous land vehicles. *Int. J. Adv. Robot. Syst.* **2020**, *17*, 1–13. [CrossRef]
10. Stentz, A. Optimal and efficient path planning for partially-known environments. In Proceedings of the 1994 IEEE International Conference on Robotics and Automation, San Diego, CA, USA, 8–13 May 1994; pp. 3310–3317. [CrossRef]
11. Daniel, K.; Nash, A.; Koenig, S.; Felner, A. Theta*: Any-angle path planning on grids. *J. Artif. Intell. Res.* **2010**, *39*, 533–579. [CrossRef]
12. Dolgov, D.; Thrun, S.; Montemerlo, M.; Diebel, J. Path planning for autonomous vehicles in unknown semi-structured environments. *Int. J. Robot. Res.* **2010**, *29*, 485–501. [CrossRef]
13. Tang, B.; Hirota, K.; Wu, X.; Dai, Y.; Jia, Z. Path planning based on improved hybrid A* algorithm. *J. Adv. Comput. Intell.* **2021**, *25*, 64–72. [CrossRef]
14. Wang, J.; Chen, Z. A novel hybrid map based global path planning method. In Proceedings of the 2018 3rd Asia-Pacific Conference on Intelligent Robot Systems (ACIRS), Singapore, 21–23 July 2018; pp. 66–70. [CrossRef]
15. Oleiwi, B.K.; Roth, H.; Kazem, B.I. A hybrid approach based on ACO and GA for multi objective mobile robot path planning. *Appl. Mech. Mater.* **2014**, *527*, 203–212. [CrossRef]
16. Song, X.; Gao, S.; Chen, C.; Cao, K.; Huang, J. A new hybrid method in global dynamic path planning of mobile robot. *Int. J. Comput. Commun.* **2018**, *13*, 1032–1046. [CrossRef]
17. Yonetani, R.; Taniai, T.; Barekatain, M.; Nishimura, M.; Kanezaki, A. Path planning using neural a* search. In Proceedings of the 38th International Conference on Machine Learning (ICML), Virtual, 18–24 July 2021; pp. 12029–12039.
18. Liu, F.; Qiu, S. Path planning of indoor mobile robot based on improved A* algorithm. In Proceedings of the 2nd International Conference on Artificial Intelligence and Information Systems, Chongqing, China, 28–30 May 2021; pp. 1–4. [CrossRef]
19. Sun, Y.; Zhao, X.; Yu, Y. Research on robot random obstacle avoidance method based on fusion of improved A* algorithm and dynamic window method. *Electronics* **2021**, *11*, 2683. [CrossRef]
20. Jiang, H.; Sun, Y. Research on global path planning of electric disinfection vehicle based on improved A* algorithm. In Proceedings of the International Conference on Energy Engineering and Power Systems, Hangzhou, China, 20–22 August 2021; pp. 1270–1279. [CrossRef]
21. Chen, T.; Sun, Y.; Dai, W.; Tao, W.; Liu, S. On the shortest and conflict-free path planning of multi-AGV system based on Dijkstra algorithm and the dynamic time-window method. *Adv. Mater. Res.* **2013**, *645*, 267–271. [CrossRef]
22. Rosmann, C.; Feiten, W.; Woesch, T.; Hoffmann, F.; Bertram, T. Trajectory modification considering dynamic constraints of autonomous robots. In Proceedings of the 7th German Conference on Robotics, Munich, Germany, 21–22 May 2012; pp. 74–79.
23. Niu, C.; Li, A.; Huang, X.; Li, W.; Xu, C. Research on global dynamic path planning method based on improved A* algorithm. *Math. Probl. Eng.* **2021**, *1*, 4977041. [CrossRef]
24. Li, Y.; Jin, R.; Xu, X.; Qian, Y.; Wang, H.; Xu, S.; Wang, Z. A mobile robot path planning algorithm based on improved A* algorithm and dynamic window approach. *IEEE Access* **2022**, *10*, 57736–57747. [CrossRef]
25. Heiden, E.; Palmieri, L.; Bruns, L.; Arras, K.O.; Sukhatme, G.S.; Koenig, S. Bench-MR: A motion planning benchmark for wheeled mobile robots. *IEEE Robot. Automat. Lett.* **2021**, *6*, 4536–4543. [CrossRef]
26. Zhang, J.; Singh, S. LOAM: Lidar odometry and mapping in real-time. In Proceedings of the Robotics: Science and Systems Conference, Berkeley, CA, USA, 14–16 July 2014. [CrossRef]
27. Shan, T.; Englot, B. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 4758–4765. [CrossRef]
28. Shan, T.; Englot, B.; Meyers, D.; Wang, W.; Ratti, C.; Rus, D. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 25–29 October 2020; pp. 5135–5142. [CrossRef]
29. Xu, W.; Zhang, F. FAST-LIO: A fast, robust LiDAR-inertial odometry package by tightly-coupled iterated Kalman filter. *IEEE Robot. Automat. Lett.* **2021**, *6*, 3317–3324. [CrossRef]
30. Bai, C.; Xiao, T.; Chen, Y.; Wang, H.; Zhang, F.; Gao, X. Faster-LIO: Lightweight tightly coupled LiDAR-inertial odometry using parallel sparse incremental voxels. *IEEE Robot. Automat. Lett.* **2021**, *7*, 4861–4868. [CrossRef]
31. Miądlicki, K.; Pajor, M.; Saków, M. Ground plane estimation from sparse LIDAR data for loader crane sensor fusion system. In Proceedings of the 22nd International Conference on Methods and Models in Automation and Robotics (MMAR), Miedzyzdroje, Poland, 28–31 August 2017; pp. 717–722. [CrossRef]
32. Hornung, A.; Wurm, K.; Bennewita, M.; Stachniss, C.; Burgard, W. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Auton. Robot.* **2013**, *34*, 189–206. [CrossRef]