

Article

A Full-Coverage Path-Planning Algorithm for a Glass-Curtain-Wall-Cleaning Robot Driven by Ropes

Dong Zhang ^{1,2}, Yuao Li ², Pei Jia ², Xin Jiao ², Yueshuo Zheng ², Guoliang Wang ², Zhihao Li ², Minglu Zhang ², Jingtian Wang ^{2,*} and Manhong Li ^{1,2,*}

¹ State Key Laboratory of Reliability and Intelligence of Electrical Equipment, Hebei University of Technology, Tianjin 300130, China

² School of Mechanical Engineering, Hebei University of Technology, Tianjin 300130, China

* Correspondence: 202011201011@stu.hebut.edu.cn (J.W.); 2015038@hebut.edu.cn (M.L.)

Abstract: Rope-driven robots are increasingly being applied for the efficiently cleaning of glass curtain walls. However, increasingly complex wall surfaces and the various shapes of obstacles may block the robot and reduce coverage. In this study, three-DOF rope-driven cleaning robots and a full-coverage path-planning algorithm were developed to achieve global operation. The robot adopts a five-rope parallel configuration, and four winches are mounted on the wall and one on the ground to produce 3D motion performance. We used a grid method to build the wall model to mark obstacles, and then we decomposed it according to the wall curvature to better access cleaning subareas. To further increase the cleaning coverage rate, a full-coverage path-planning algorithm based on an improved priority heuristic was designed, which does not ignore the inset area of U-shaped obstacles. By introducing two sets of priority criteria to judge the forward direction, the robot can switch directions to cover a whole area when encountering U-shaped obstacles. Furthermore, by planning a return route requiring the least amount of time when entering a dead zone, an escape strategy was developed to prevent the robot from being unable to choose a direction. The experimental results show that the robot, after applying the proposed path-planning algorithm, could complete the global cleaning of complex glass walls with various obstacles.

Keywords: glass wall cleaning; rope-driven robot; wall modeling; full-coverage path planning



Citation: Zhang, D.; Li, Y.; Jia, P.; Jiao, X.; Zheng, Y.; Wang, G.; Li, Z.; Zhang, M.; Wang, J.; Li, M. A Full-Coverage Path-Planning Algorithm for a Glass-Curtain-Wall-Cleaning Robot Driven by Ropes. *Appl. Sci.* **2023**, *13*, 5052. <https://doi.org/10.3390/app13085052>

Academic Editors: Dimitris Mourtzis and Marco Troncosi

Received: 27 January 2023

Revised: 6 April 2023

Accepted: 7 April 2023

Published: 18 April 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

At present, with the rapid development of the social economy and the improvement in people's living standards, the application of glass curtain walls in buildings is becoming increasingly popular [1–3]. These glass walls need to be regularly cleaned to prolong building service life [4–6]. While the traditional method for glass curtain wall cleaning relies on manual labor, working at heights poses risks to worker safety. Thus, façade-cleaning robots have been adopted as an assistant or substitute for human workers performing these tasks. In addition, the curved surfaces and irregularly shaped obstacles on buildings place stricter demands on robot performance. For robots working on glass curtain walls, most research has focused on prototype design and stability control; little attention has been paid to the obstacle-overcoming and path-planning abilities of these robots [7–9]. Therefore, studies on robot DOF extensions and global path-planning algorithms are required.

Currently, some robots use wheels or mechanical legs to crawl on façades for wall cleaning [10–12]. For example, Liu et al. developed a wheeled wall-climbing robot with the ability to climb walls using circular arrays of miniature spines located around the wheel [13]. Wang et al. proposed a flexible and adjustable wheel-leg composite cleaning robot composed of three leg frames on a rotating shaft to expand the movement space [14]. In addition, one type of wall-cleaning robot is driven by ropes. TITO, ROPE RIDE, Sky Scraper-I, etc., have been proposed as robots with a structure combining ropes and slide

rails to realize two-DOF motion, but their stability is insufficient on high buildings [15–17]. To safely apply robots at high altitude, Kite uses four independent winches to control four ropes from four different angles [18]. The winches and cleaning body form a planar two-DOF redundant parallel mechanism, which limits excess body movement, so movement stability can be effectively guaranteed. However, this robot cannot work on a wall with raised obstacles and sunken surfaces because of the lack of movement freedom perpendicular to the wall direction. For the obstacle-overcoming problem, Dai et al. added an electric push rod that can retract perpendicular to the wall for pushing ropes away from the surface [19]. So, the robot can overcome obstacles, but the performance is restricted by the rod length. As taller buildings require more water and cleaning products, this considerably increases the engineering requirements for robot load capacity. However, a robot using legs or wheels is driven by its own adsorption force to move the load, which limits the load capacity; therefore, a rope-driven robot with a high load capacity is more appropriate for window cleaning. From the analysis of complex curtain wall shapes, three-dimensional space motion ability is necessary for a robot to successfully overcome obstacles.

Wall modeling is the premise of robot path planning. A variety of environmental modeling methods have been proposed, among which the most common are the grid, topology, and viewable methods [20–22]. In the grid method, each grid corresponds to a region in the map and represents the environmental information of the region [23–25]. With the topological method, a region is divided into multiple subspaces according to the geometric features of the environment [26–28]. The main idea of the viewable method is to view the robot as a particle and the obstacle as a polygon [29–31]. However, when topological methods encounter regions with similar features, constructing the corresponding topological maps is challenging due to an inability to distinguish similar regions. The disadvantage of the viewable method is that as the number of obstacles increases, the number of lines between vertices increases, which leads to increases in the path-planning complexity and time consumption. Therefore, the topology and viewable methods are unsuitable for wall surfaces with complex-shaped obstacles. In contrast, the grid method is more appropriate because it can determine the complete obstacle shape by setting smaller grid sizes.

A path-planning algorithm is necessary for a robot to realize the task of global glass wall cleaning, which demands the planned path to be as short as possible with a low repetition rate. To solve the problem posed by the global coverage of operational areas, a series of path-planning algorithms have been proposed. The cattle tillage covering algorithm (BCD) proposed by Choset et al. divides a working area into several subareas, and then the robot uses reciprocating or spiral motion to cover the area in each subarea [32]. When encountering obstacles, the robot uses a back-off operation to complete the covering task. He et al. proposed a full-coverage path-planning algorithm for a sweeping robot [33], which used a priority inspiration function to determine the moving position of the robot and used a genetic algorithm to solve the problem of falling into dead zones. Tang et al. presented a path-planning robot based on a rolling window priority heuristic algorithm for a wall-climbing robot, using a rolling window combined with a mobile traversal priority rule implementation area, preventing the robots from being trapped in dead zones [34]. Ma et al. designed a new fusion algorithm based on the improved A* and Open_Planner algorithms (A-OP), improving the smoothness of the path using the minimum snap method and A-OP algorithm to implement the path planning of an unmanned sweeper. [35]. Chen et al. proposed a cognitive space–time environment modeling approach for autonomous vehicles, which models a multiscenario-adapted space–time environment model from a cognitive perspective and transforms the scenario-based trajectory planning problem into a unified spatial–temporal planning problem [36]. Hernandez and Carreras developed a method to generate and sort homotropy classes according to a lower bound heuristic estimator. The classes were used to constrain and guide path-planning algorithms [37]. Galceran and Carreras presented a review of the most successful coverage path-planning (CPP) methods, which determine a path that passes over all points of an area or volume of interest

while avoiding obstacles. Furthermore, they discussed the reported field applications of the described CPP methods [38]. Emile et al. investigated CPP for the robotic single-sided dimensional inspection of free-form surfaces. Others improved CPP by replacing the random viewpoint sampling strategy [39]. Wang et al. proposed a CPP method based on deep reinforcement learning for a kiwifruit-picking robot, using LiDAR to collect the environmental point cloud information of the kiwifruit orchard and construct a two-dimensional grid map; an improved deep reinforcement learning algorithm, the re-DQN algorithm, was proposed to determine the traversal order of each region [40].

In summary, existing rope-driven robots face the following problems in obstacle overcoming and path planning. 1. The robot can complete cleaning operations on simple wall surfaces, but moving from one piece of glass to another is difficult, as these panes of glass are separated by large barriers such as balconies or partitions. 2. The existing path-planning algorithms generally ignore the inset area of U-shaped obstacles, which limits the cleaning coverage rate. 3. The dead zone escape strategy has not been widely applied in façade cleaning path planning, so the robot can become stuck in dead zones and unable to choose a direction.

To address the above problems, we used five winches to establish a spatial model to drive the robot body through ropes, so that it has three degrees of freedom of movement to overcome obstacles. Using the grid method to build the wall model to mark obstacles, we then decomposed it according to the wall curvature to more thoroughly clean subareas. To further improve the cleaning coverage rate, a full-coverage path-planning algorithm based on an improved priority heuristic was developed, which does not ignore the inset area of U-shaped obstacles. By introducing two sets of priority criteria with which to judge the forward direction, the robot can switch directions to cover the whole area when encountering U-shaped obstacles. Furthermore, by planning a return route taking the least amount of time possible when entering a dead zone, an escape strategy was designed to prevent the robot from being unable to choose a direction. Aiming at problems in terms of the structure design, wall modeling, and path planning of cleaning robots, we propose a five-rope 3D driving configuration, a grid modeling method, and a full-coverage path-planning algorithm including a dead zone escape strategy to achieve the efficient global cleaning of a curved wall with complex obstacles.

The rest of this paper is arranged as follows. Section 2 introduces the rope-driven cleaning robot structure. The wall model gridding and region decomposition are shown in Section 3. Section 4 introduces the path-planning algorithm based on the priority heuristic. Section 5 describes the experimental procedure and results of the analysis. Section 6 provides several conclusions from this study.

2. Mechanical Design of Cleaning Robot

When the robot encounters large obstacles such as balconies and partitions, using the jumping strategy to overcome obstacles is most effective. The main idea of the jumping strategy is to realize 3D movement by relying on the 5-rope parallel configuration.

To obtain continuous and flexible three-dimensional space motion on complex glass curtain walls, the proposed rope-driven parallel cleaning robot is mainly composed of a cleaning body with four propeller fans to provide wind pressure and five driving winches. The propeller fans in the four corners can form even and stable pressure, pushing the robot body to closely fit the surface, solving the problem of efficient cleaning for concave surfaces. In accordance with previous studies [29,33], the rope-driven robots connected to more than $n + 1$ ropes in a space with n degrees of freedom are called redundant rope-driven parallel robots. In this configuration, the ropes in different directions are always subjected to a high tension in the workspace, which substantially reduces vibrations in the ropes. So, four winches are mounted on the façade, similar to a normal rope-driven robot, but one particular winch is mounted on the ground and provides a pulling force perpendicular to the façade, enabling the robot to leave the wall surface.

As shown in Figure 1, the four winches on the façade are designed to maintain the movement stability on the two-dimensional plane of façade to prevent the robot body from shaking due to external factors and colliding with the wall or obstacles. The winch is the power unit that drives the cleaning body to move by controlling the winding/unwinding of the rope fastened to the body. A winch placed on the ground provides a third degree of freedom for robot movement, giving the robot the ability to “jump” on walls to easily cross obstacles. When the robot encounters an obstacle, a fifth rope lock is shortened; at the same time, with the change in the other four ropes’ lengths, it can be temporarily removed from the wall to avoid collisions with obstacles. After the robot travels over the obstacle, the fifth rope is extended and works with the other four ropes again. At this time, the robot can return to the wall surface to continue its cleaning work.

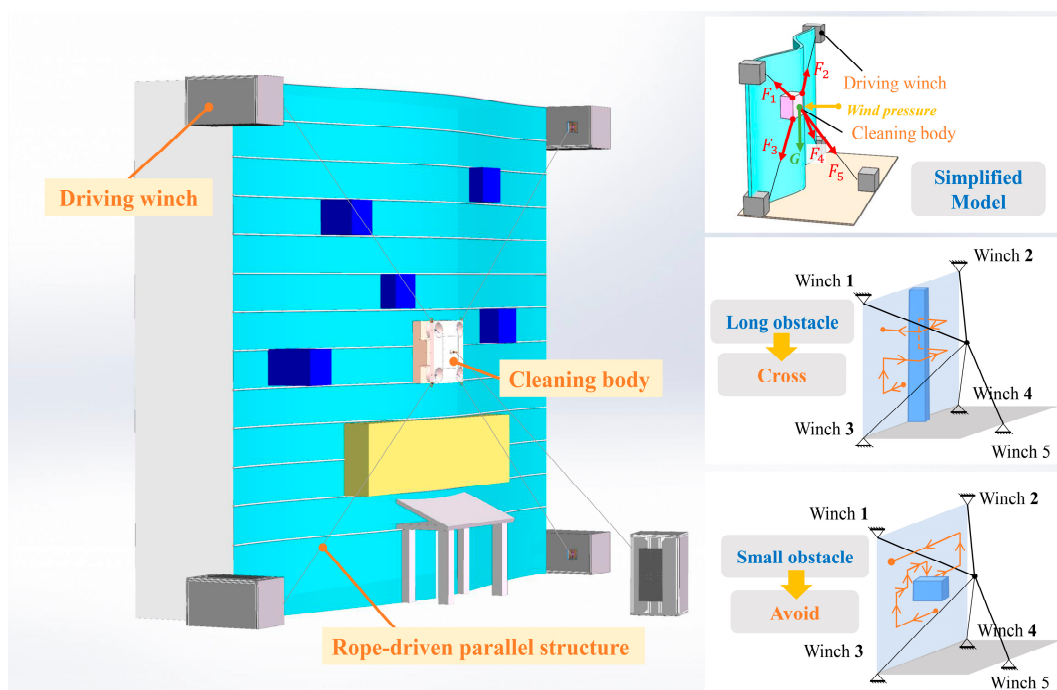


Figure 1. Robot structure.

Therefore, through the above structural design, the robot can overcome large obstacles and realize stable spatial motion. This is crucial for the robot to efficiently complete global cleaning.

3. Wall Modeling

3.1. Wall Gridding

To clean a large glass curtain wall, the body of the robot must be pulled by the rope of winch 5 to make it continuously jump to overcome obstacles. This is inefficient on walls with densely distributed obstacles. Therefore, a full-coverage path must be planned for the robot to avoid obstacles with as little jumping as possible to efficiently complete the cleaning work.

The aim of full-coverage path planning is to find the path that cleans the maximum area. To facilitate robot path planning, the edge learning strategy can be adopted, in which sensors are used to obtain the shape and location information of obstacles on the wall. Through data processing, a three-dimensional model of the wall is constructed to simplify the wall, which is then developed into a two-dimensional plane with length L and width H . This is convenient for the subsequent wall modeling, which is achieved using the grid method to represent the distributed obstacles. Then, the working environment can be

simplified to a two-dimensional rectangular plane model with length L and height H by approximate processing, using grids to represent obstacles, as shown in Figures 2 and 3.

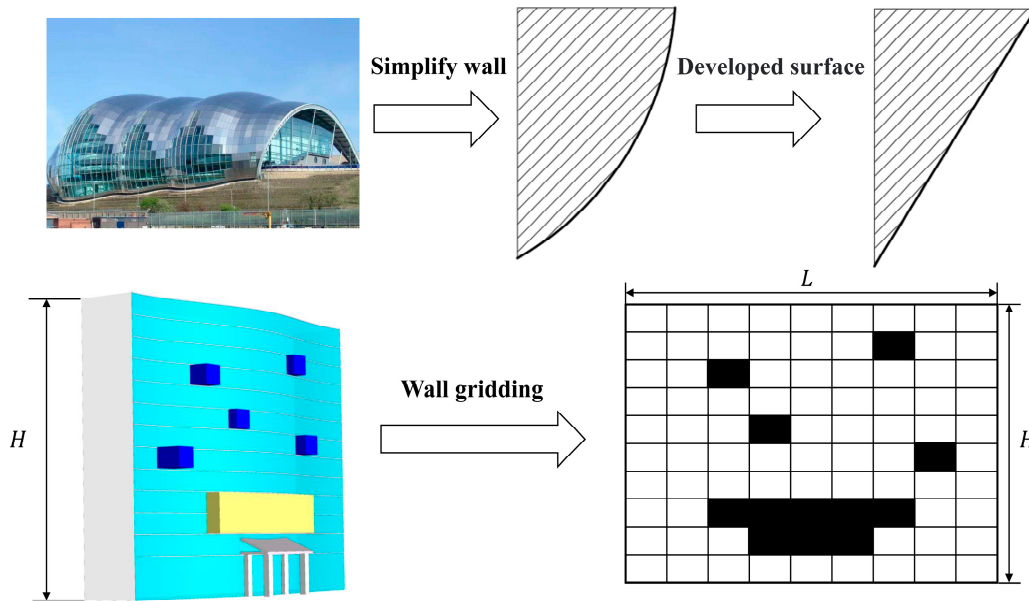


Figure 2. Wall modeling.

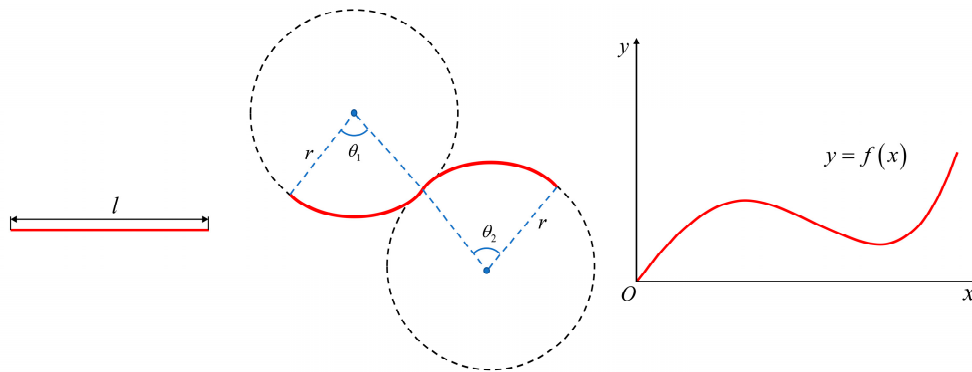


Figure 3. Wall length solution.

To simplify the calculation, the curve length of the surface is straightened to L of the rectangular model, and the height of the building is taken as H . Therefore, if the wall shape is a straight line, L is the straight-line length l ; if the wall shape is an arc, L can be calculated according to the arc length calculation formula; if the wall shape is a curve with variable curvatures, a curve function equation is needed, and L can thus be calculated according to the curve length calculation formula, as shown in Equation (1):

$$L = \begin{cases} l & \text{Straight line} \\ \frac{\theta_1 \pi r + \theta_2 \pi r}{180} & \text{Arc line} \\ \int_a^b \sqrt{[f'(x)]^2 + 1} dx & \text{Variable curvature curve} \end{cases} \quad (1)$$

Based on the two-dimensional rectangular plane model, the operating environment can be developed with the grid method to construct the map, which is relatively simple and convenient in terms of path-planning algorithm design.

To facilitate understanding, each grid represents a point on the map. Considering the robot size, computational efficiency, accuracy, and other factors, the grid size was set as

10 cm. (x_i, y_i) is defined as the position coordinates of grid k , where x_i and y_i denote its horizontal and vertical coordinates, respectively. To facilitate the calculation and processing, each grid attribute is assigned a value, as shown in Equation (2). The three values of 0, 1, and 2 are used to represent the three states of the grid: obstacle, uncovered, and covered grids, respectively, so the position attribute of grid k can be represented as (x_i, y_i, f_k) .

$$f_k = \begin{cases} 0 & \text{obstacle raster} \\ 1 & \text{uncovered raster} \\ 2 & \text{covered raster} \end{cases} \quad (2)$$

The environmental state of a robot can be judged by assigning grid attributes. As the robot moves, the numbers of covered and uncovered grids change and update. In addition, in the robot’s operational environment, some irregular obstacles may exist in the established grid map that do not completely coincide with the grid boundary. To simplify the representation of the grid map model as much as possible, these obstacles must be expanded into the regular grid model during grid map modeling. Based on the acquired wall map, the heuristic algorithm is both simple and operable. However, many wall-climbing robots are limited by insufficient movement and turning abilities and low intelligence during façade maintenance working. Therefore, considering the actual motion of a robot, we developed an improved priority heuristic full-coverage path-planning algorithm based on the reciprocating coverage algorithm. The algorithm sets the bottom length of the subregion grid map as L and the height as H . From the grid map perspective, the robot has eight moving directions: up, down, left, right, up left, down left, up right, and down right. The number of uncovered grids from the current grid to the obstacle grids or covered grids in the upper, lower, left, and right directions is defined as $N_n, N_s, N_w,$ and $N_e,$ respectively.

Figure 4a shows a grid map. By assigning grid attributes, the corresponding digital matrix of the grid map can be obtained, as shown in Figure 4b. In the uncovered case, no uncovered grids exist in the environment. When full coverage of the workspace is achieved, the uncovered grids become covered grids. The digital matrix of the grid map at this time is shown in Figure 4c.

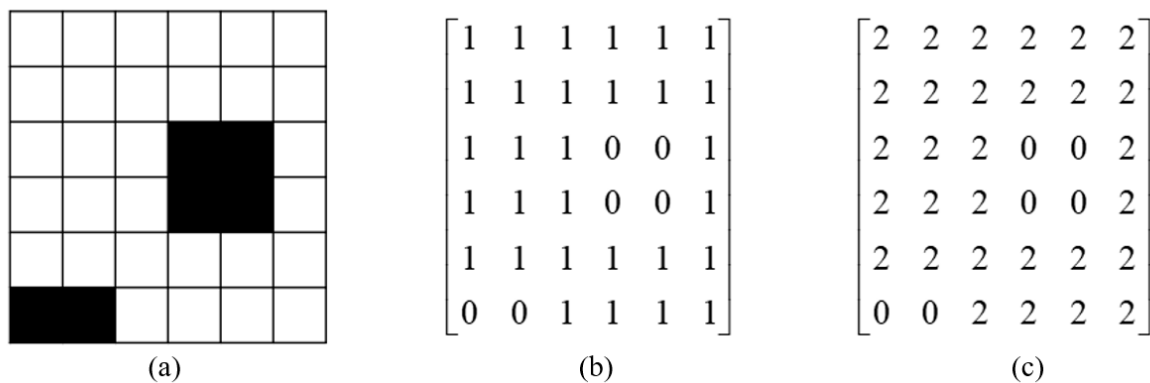


Figure 4. Grid map matrix: (a) grid map; (b) uncovered map digital matrix; (c) overlaid map digital matrix.

If all the cells of a row or column of the matrix are 0, as shown in Figure 5, the obstacle cannot be avoided; in this case, the obstacle must be crossed by jumping.

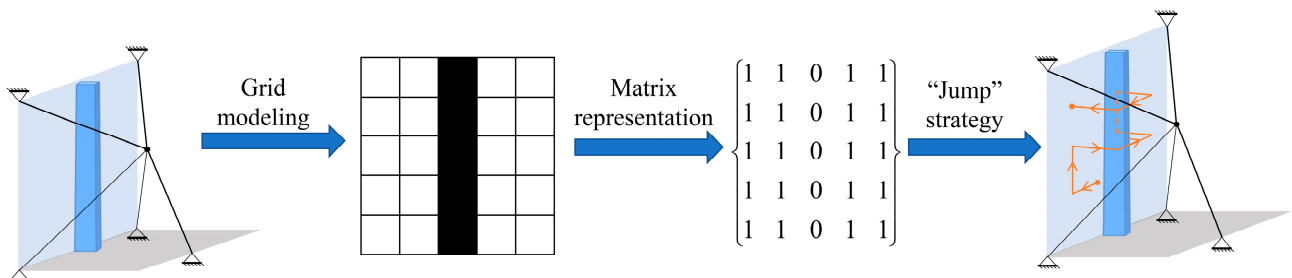


Figure 5. Application scenario of the jump obstacle-crossing mode.

3.2. Region Decomposition

Because a glass curtain wall is usually larger than the cleaning body, when the wall curvature is too large or some raised obstacles are present, the wall and rope controlling the cleaning body inevitably collide, causing friction, which we call interference, as shown in Figure 6. This considerably reduces the safety of the robot. So, reasonably decomposing the working area is an important issue in the path planning of a full-coverage robot. To avoid interference, according to the curvature of the wall, the whole façade working area is divided into several subareas by changing the winch horizontal position, so that the robot can separately operate in each subarea. Furthermore, to obtain more detailed information about the obstacle shape, the subareas should be as small as possible. Therefore, we suggest that concave regions should also be decomposed.

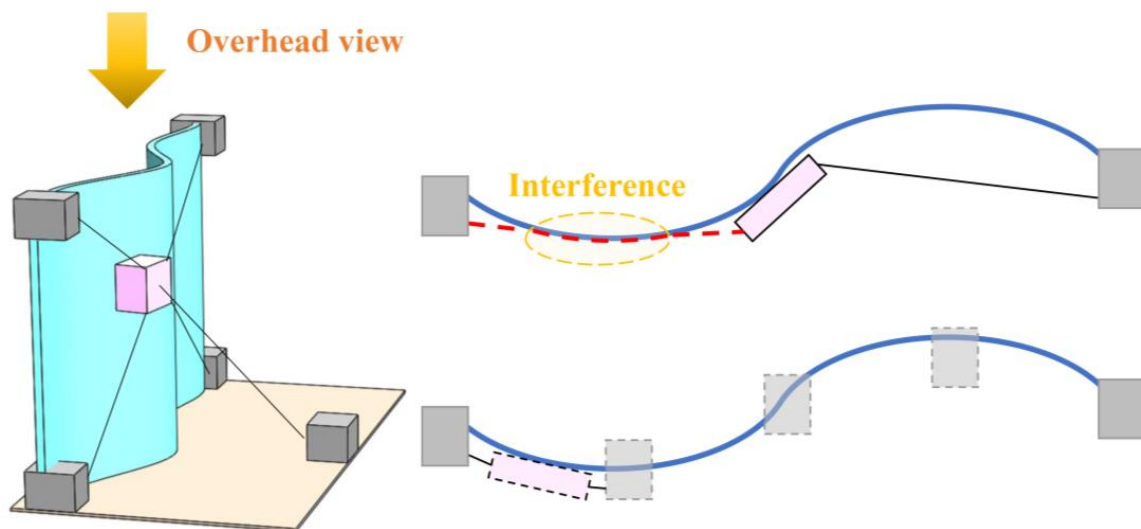


Figure 6. Interference problem between rope and wall surface.

Depending on the height of the winch relative to a wall bulge, we can divide the area by setting an empirical value so that interference does not occur in the subarea.

First, the overlooking shape of the glass curtain wall is abstracted into a curve to facilitate slope calculation. The upper left vertex of the curtain wall is selected as the origin of the coordinates to establish a coordinate system, and $0, \pm\sqrt{3}/2$ is taken as the ultimate slope (empirical value). From the beginning to the absolute value of the ultimate slope, the range can be viewed as a partition. Then, the end point of this partition is selected as the starting point of the next partition, and the starting point is used as the origin of the coordinates to establish the coordinate system. Then, the slope of the curve is calculated. The previous process is repeated until the end point is reached. The wall shown in the image can eventually be broken down into four subregions, as shown in Figure 7.

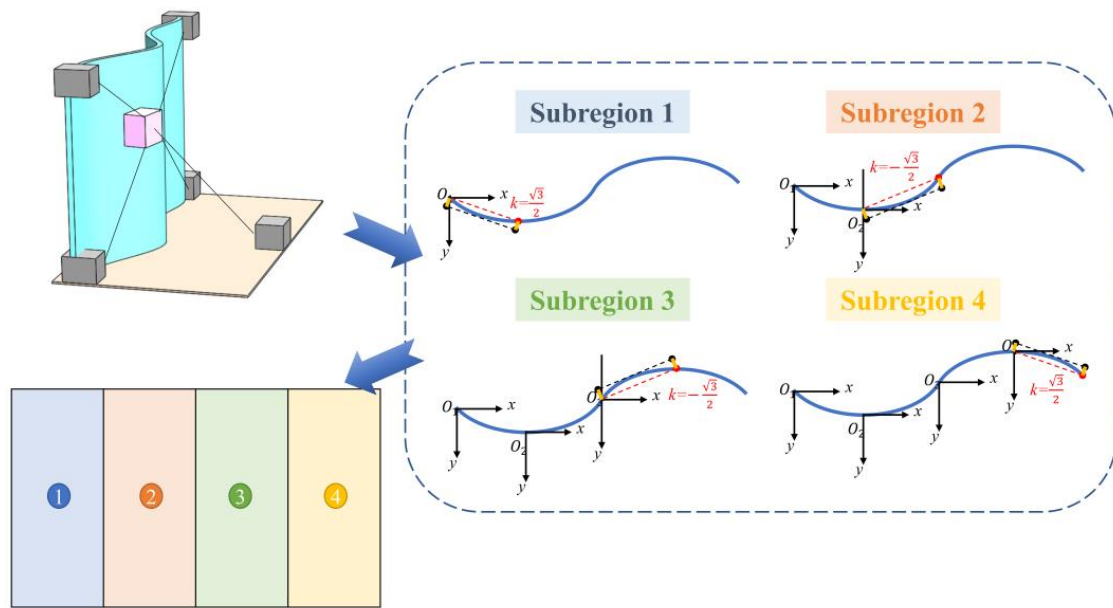


Figure 7. Region decomposition.

4. Full-Coverage Path Planning

To achieve autonomous robot full-coverage path planning, through the analysis of the previous study results related to CPPS, we selected a series of path-planning principles based on the improved priority heuristic algorithm, which can considerably improve the efficiency of the robot. A large exterior glass curtain wall surface is often complex; in addition to conventionally shaped obstacles, U-shaped obstacles may be present, which pose challenges to the robot body moving strategy. The dead zone position problem also arises during the working process. Therefore, effectively avoiding obstacles to achieve full coverage and escaping dead zones as quickly as possible have become key to improving robot performance.

To solve these problems, an improved priority heuristic algorithm was designed, which prioritizes the moving direction according to the priority order designed based on the center place of the robot in the established wall model. Thereby, a reasonable ergodic path can be planned. To further illustrate the improved priority heuristic algorithm, we selected three common wall models and proposed three path planning principles as examples.

4.1. Conventional Obstacle Avoidance Strategy

On a wall surface with conventional obstacles, when $L \leq H$, the robot's overall motion scheme is longitudinal reciprocating motion. When $L > H$, the robot's overall motion scheme is transverse reciprocating motion. The specific rules are as follows:

(1) When $L \leq H$, the first rule is that the horizontal and vertical robot motions have higher priority than oblique motions. The second rule is that if the robot starts to perform a coverage task from the subregion left (right) side, the priority of left (right) motion is set as the highest, the priority of up and down motion is second, and the priority of right (left) motion is lower than those of up and down motion. The third rule is calculating the number of uncovered grids in the upper and lower directions from the robot's current grid to the obstacle or covered grid, which are set as N_n and N_s , respectively. If $N_n < N_s$, the robot's upward motion has higher priority than the downward motion. Otherwise, the downward motion has higher priority than the upward motion. This rule ensures that the robot preferentially covers the side with a low number of uncovered grids to reduce the path repetition rate of the robot.

(2) When $L > H$, based on the transverse reciprocating motion, three priority rules were formulated for the robot to select the next grid position. The first rule follows that in (1). The second rule is that if the robot starts to perform the coverage task from the subarea's upper (lower) side, the upward (downward) motion priority is set as the highest, the left and right lower motion priority as the second-highest, and the downward (upward) motion priority is lower than that of left and right motion. The third rule is to calculate the number of uncovered grids in the left and right directions from the robot's current grid to the obstacle or covered grid, which are N_w and N_e , respectively. If $N_w < N_e$, the robot's left movement has higher priority than right movement. Otherwise, the right movement has higher priority than left movement.

This overall obstacle avoidance strategy is shown in Figure 8.

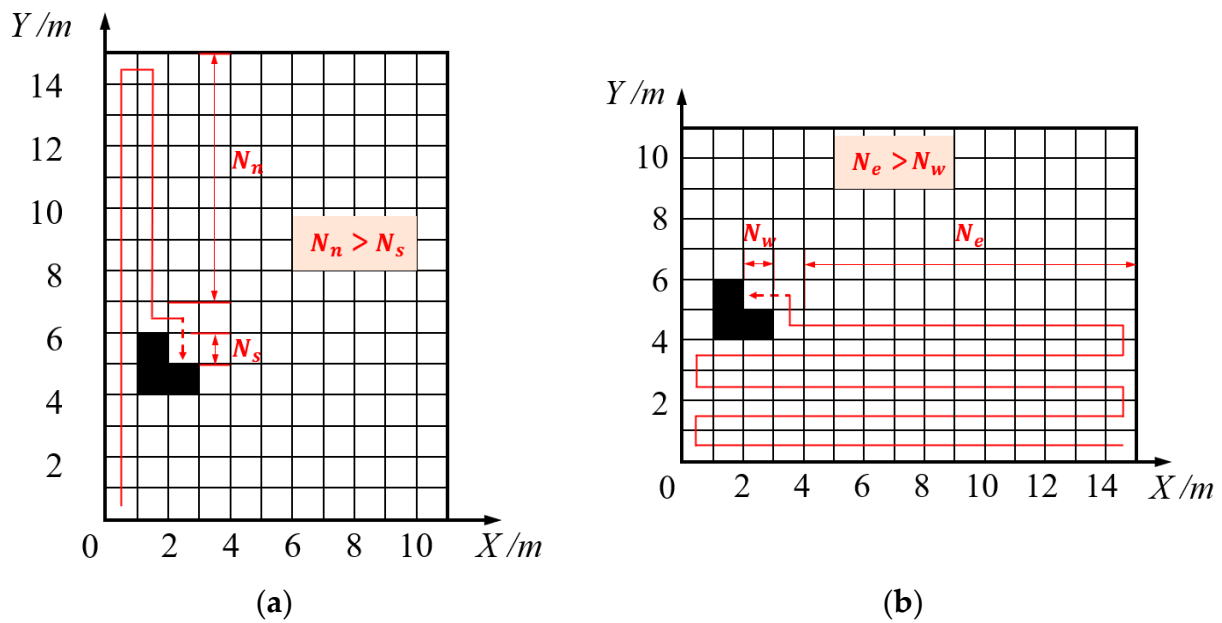


Figure 8. Conventional obstacle avoidance strategy: (a) longitudinal and (b) transverse reciprocating motion.

4.2. U-Shaped Obstacle Avoidance Strategy

First, the robot marks U-shaped obstacles in the grid map through edge learning and then determines whether a U-shaped obstacle is present in the grid map of the subregion whose opening direction is consistent with the initial coverage direction of the robot. If a U-shaped obstacle is present, the switching command operation is used to cover the U-shaped obstacle area. The priorities for avoiding U-shaped obstacles are as follows:

When $L \leq H$, if the robot starts to perform the coverage task from the left subregion (right) side, the right (left) motion priority is set as the highest, the up and down motion priority as the second-highest, and the right (left) motion priority is lower than that of the up and down motion. When $L > H$, if the robot starts to perform the coverage task from the upper (lower) side subregion, the downward (upward) motion priority is set as the highest, the left and right lower movements' priority is second-highest, and the upward (downward) motion priority is lower than the left and right motion priorities. The other priority rules remain the same.

The overall U-shaped obstacle avoidance strategy is shown in Figure 9.

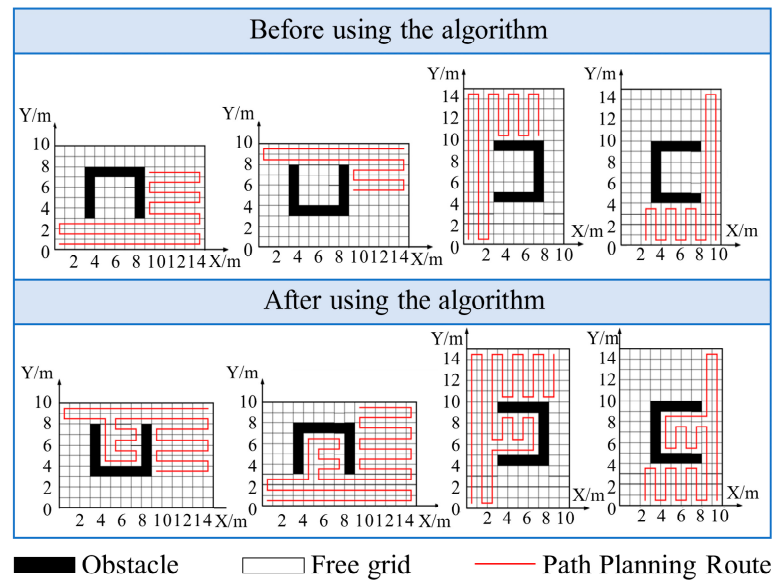


Figure 9. U-shaped obstacle avoidance strategy.

4.3. Dead Zone Escape Strategy

When a robot moves to a certain grid position, if the neighboring grids are obstacle or covered grids and some uncovered grids remain in the map, the robot may fall into a dead zone position. At this point, the robot needs to find the shortest possible path to escape the dead zone.

To address this situation, we developed a strategy to escape from dead zones. First, based on the wildfire algorithm, the properties of the eight adjacent grids are searched based on the grid where the robot is located. If an uncovered grid is found, it is marked. The grid around range is continually expanded; in other words, the attributes of 16 grid are searched, and any uncovered grid is marked. Then, the search scope is gradually expanded until the boundary of the grid map is searched, and the distance between the marked grids and dead zone grid is calculated. The grid with the shortest distance is the initial target point.

After finding the initial target point, a path to move from the dead zone position to the initial target point must be planned. We used the direction function to plan the path, which can effectively solve this problem. The direction function is defined as follows.

$$F = \min\{f_i = h_i + k_i, i = 1, 2, 3, \dots, 8\} \tag{3}$$

$$h_i = \frac{d_n}{d_i} \tag{4}$$

$$d_n = \sqrt{(x_n - x_m)^2 + (y_n - y_m)^2} \tag{5}$$

$$d_i = \sqrt{(x_i - x_m)^2 + (y_i - y_m)^2} \tag{6}$$

$$k_i = a \frac{\theta_k}{\pi} \tag{7}$$

where h_i represents the distance function; k_i is the steering function, d_n and d_i represent the distance from the next-move grid to target point grid and the distance from the current grid to target point grid, respectively; (x_n, y_n) , (x_m, y_m) , and (x_i, y_i) denote the next-move grid coordinates, the target grid coordinates, and the current grid coordinates, respectively; i represents the number of robot-adjacent grids. Because the next-move grid is the domain

grid of the current grid, the steering angles θ_k are $0, \pi/4, \pi/2, 3\pi/4,$ and π . The value of a is 0.2, which represents the influence of the steering angle on the robot's selection of grid position for the next move.

As shown in Figure 10, if the robot moves to position ①, it falls into a dead zone. At this time, the nearest uncovered grid position searched by the wildfire algorithm is ③, which is set as the initial target point. The path from the dead zone position ① to the initial target point position ③ calculated by the direction function is indicated by the blue line. The first uncovered grid position in this path is ②, which is the expected target point. As the robot moves from ① to the expected target point ②, it escapes the dead zone.

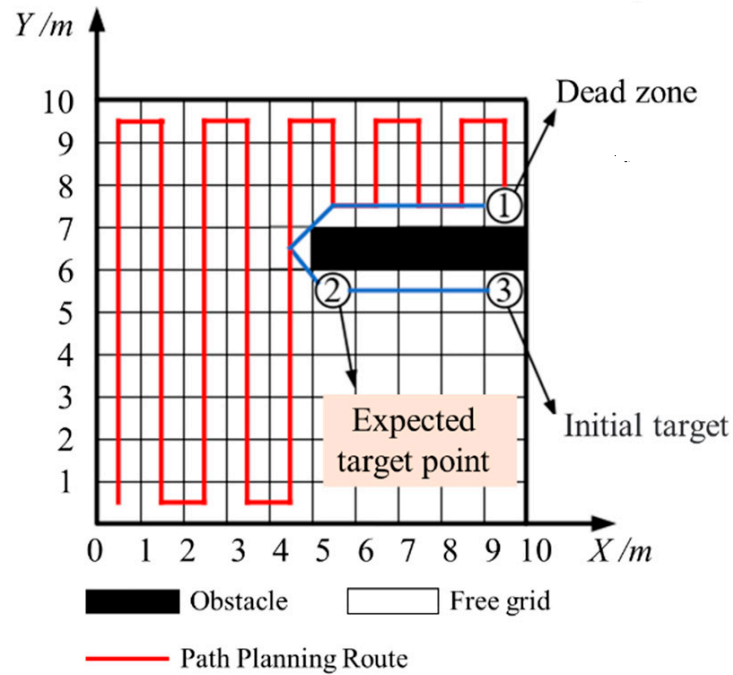


Figure 10. Dead zone escape strategy.

4.4. Full-Coverage Path-Planning Algorithm

Based on the proposed priority-setting rule, the glass-cleaning work can be performed on most walls according to the following steps:

Step 1. Set the robot's initial position; mark U-shaped obstacles whose opening direction is consistent with the initial coverage direction using the edge learning strategy.

Step 2. The robot determines the next-move direction according to the improved priority heuristic full-coverage path-planning algorithm, and the robot covers the moving direction following this plan.

Step 3. If the robot moves to a U-shaped obstacle whose opening direction is consistent with the initial coverage direction, execute the switching command operation to cover its area, and then proceed to Step 5 after the coverage is completed.

Step 4. If robot enters a dead zone, execute the escape from the dead zone strategy, search for the nearest uncovered grid to the dead zone position, and set it as the initial target point. Use the direction function to plan the path to move to this grid and set the first uncovered grid in this path as the intended target point, so the robot can move to the target point for escaping the dead zone. Proceed to Step 5.

Step 5. Judge whether all the grids in the grid map have been covered. If not, return to Step 2; otherwise, end the algorithm.

The flow of the above improved priority heuristic algorithm is shown in Figure 11.

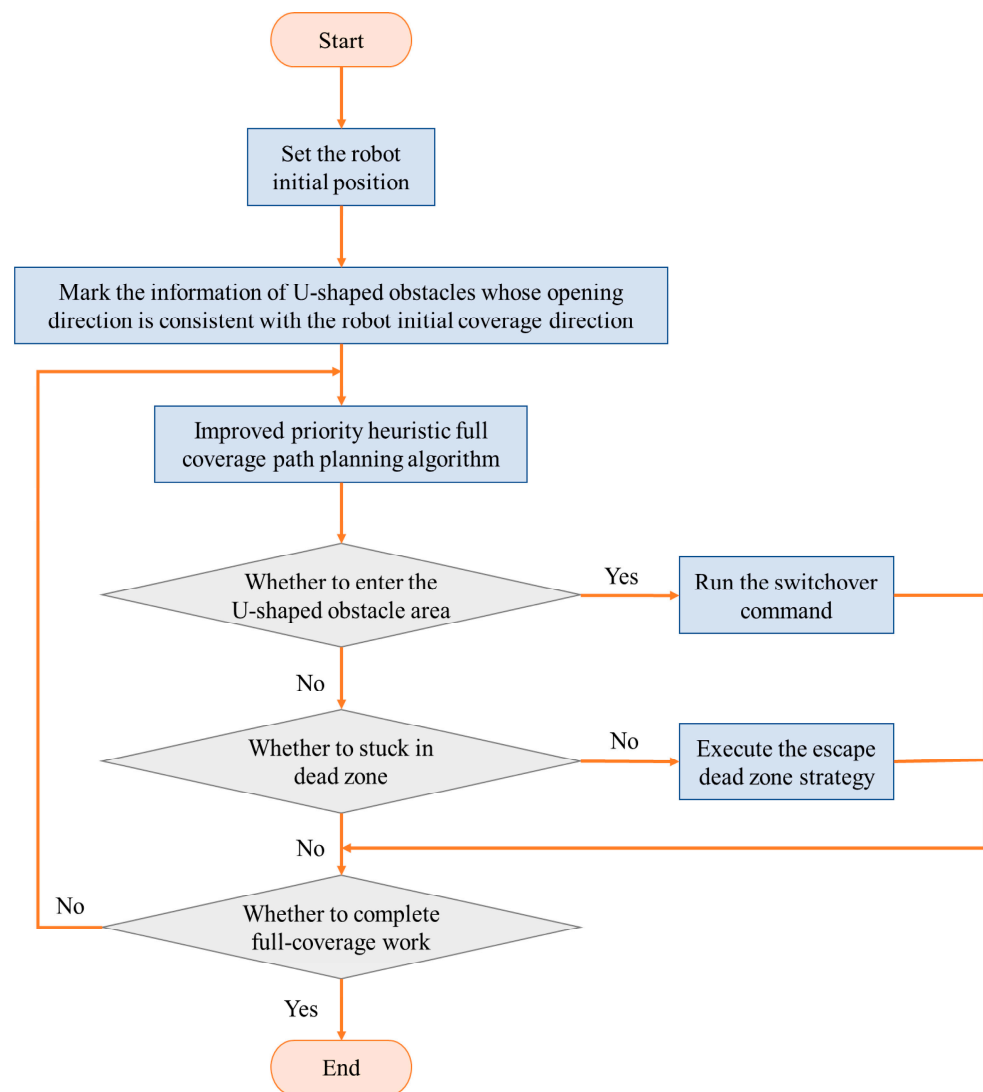


Figure 11. Improved priority heuristic algorithm.

To fully demonstrate the improved priority heuristic full-coverage path-planning algorithm, a complex wall with multiple obstacles was simulated. On this wall, $L = H = 20$, so the overall robot motion scheme was longitudinal reciprocating motion, and the starting point was the lower-left corner of the grid map. The improved priority heuristic full-coverage path-planning algorithm was used to plan the coverage path, as shown in Figure 12a.

When the robot moved to the position shown in Figure 12b, the robot obtained the obstacle information in the working environment through edge learning and determined the A positions based on the information. If the robot moved to point A, the robot would be judged as entering the obstacle.

As the robot moved to the position shown in Figure 12c, obstacles or covered grids were all around the robot, so the robot fell into a dead zone. At this time, the robot attempted to escape the dead zone, and the wildfire algorithm was used to find the initial target point.

Finally, the robot completed the global cleaning work according to the planned path, as shown in Figure 12d.

Therefore, full coverage of the working area on a complex wall was achieved by switching the rules of motion direction priority while avoiding special calculation, which verified the feasibility of the proposed full-coverage path-planning method, and effectively

solves the problem of efficient global cleaning of a wall with densely distributed obstacles by a robot.

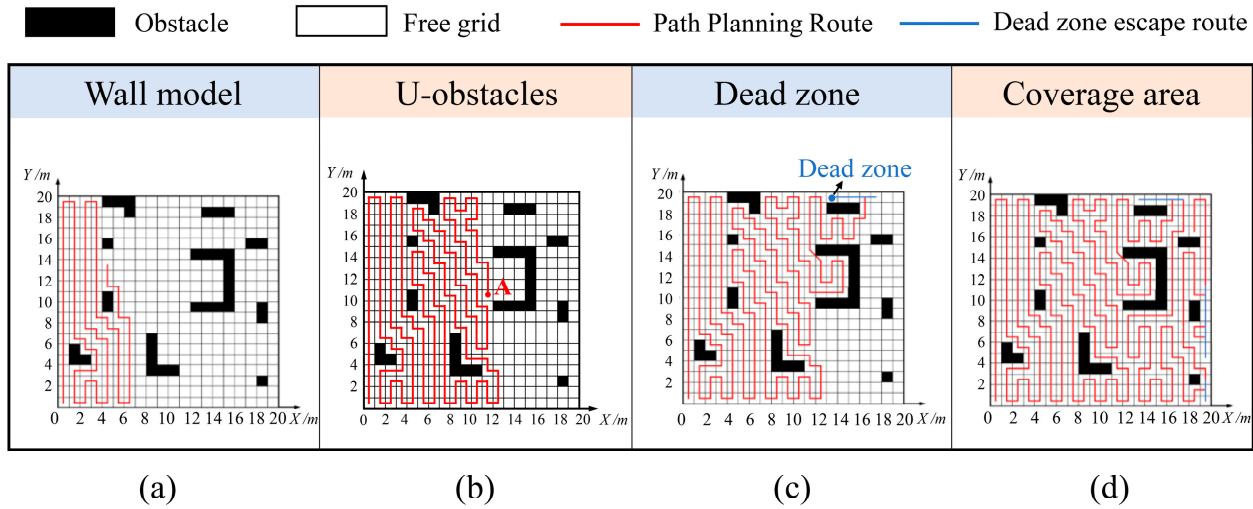


Figure 12. Full-coverage path planning: (a) wall model; (b) U obstacles; (c) dead zone; (d) coverage area.

5. Experiment and Results

To verify the feasibility of the algorithm, we built a simulation experimental platform. The outdoor glass curtain wall was replaced with an office glass partition, and the obstacles were simulated using raised cubes randomly placed on the partition. The cleaning robot was a robot prototype with a control system, as shown in Figure 13. The robot prototype contained a cleaning body and five winches equipped with encoders for feedback rope length data. The winch size was $10 \times 10 \times 30$ cm, the rope length was 9 m, the body size was $25 \times 25 \times 10$ cm, and the body weight was 3.62 kg. Winches 1 and 2 were mounted on the wall 2.5 m above the ground, with a distance of 3 m between them. Winches 3 and 4 and 1 and 2 formed a rectangle. Winch 5 was mounted on the ground 3 m from the wall. For the convenience of analysis, a working coordinate system with winch 3 as the origin was established, where the x axis was horizontally forward from winch 3 to 4, the y axis was opposite to the direction of gravity, and the z axis followed the right-hand spiral rule. In addition, a control system was used to adjust the winding/releasing of the ropes for each winch and obtain real-time rope length data. An observation camera (MV-CE060-10UM) fixed 4 m in front of the wall was used to record robot movement, and the actual position of the robot on the wall could be obtained via combining the size-proportional relationship between the image and experimental platform.



Figure 13. Experimental platform.

To verify the effectiveness of the improved priority heuristic full-coverage path-planning algorithm, the environment shown in Figure 13 was used for the experiment, wherein some different-shaped obstacles were distributed on the glass wall with an existing dead zone.

Before the experiment, the grid was modeled on the operating environment, the positions of the obstacles were marked, and the cleaning body was set to start from the lower-left corner of the map. The full-coverage algorithm proposed in this paper was implemented. The results show the escape route of the body when it encountered a dead zone. As shown in Figure 14, the cleaning robot applied the improved priority heuristic full-coverage path-planning algorithm for the area coverage task. In the process of covering the area, the cleaning body could avoid obstacles. When the body became trapped in the dead zone, the dead zone escape strategy was executed, and the robot successfully escaped from the dead zone.

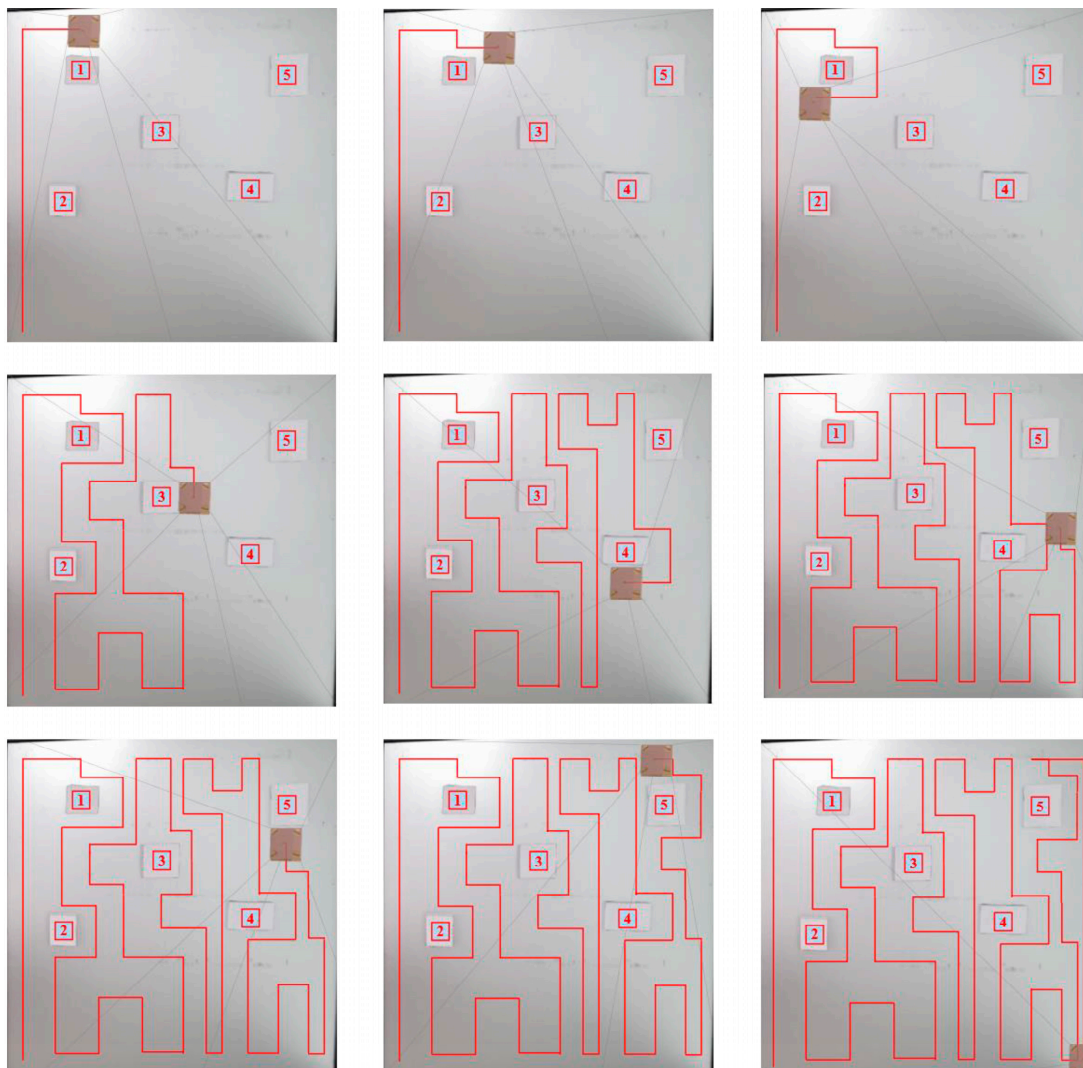


Figure 14. Full-coverage path-planning experiment.

To further verify the accuracy of the full-coverage path-planning algorithm, we set 20 points in this scene to calculate the theoretical robot position at these points, and then we measured the actual position. Finally, the position errors were obtained by comparing the two values. The experimental data are shown in Figure 15.

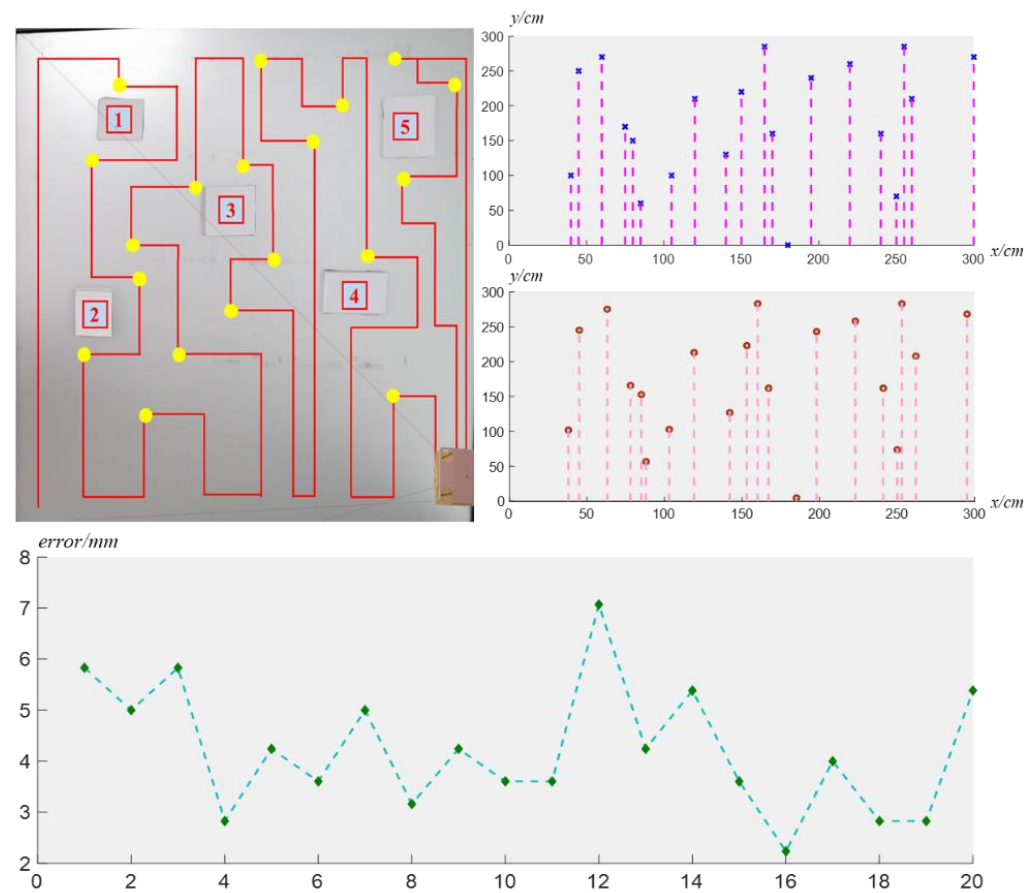


Figure 15. Body position errors.

The experimental results show that the proposed improved priority heuristic full-coverage path-planning algorithm could be applied to accurately complete the whole-area coverage task and is an efficient full-coverage path-planning algorithm.

6. Conclusions

This paper presents a 3D rope-driven robot and a full-coverage path-planning algorithm to realize the global cleaning of complex glass curtain walls. By designing a five-rope parallel configuration, the robot was made able to overcome obstacles to complete three-DOF motion. The wall model was built based on the grid method to mark obstacles, which was then decomposed according to the wall curvature to more thoroughly clean the subareas. To overcome the smooth obstacle avoidance problem for robots on complex walls, a full-coverage path-planning algorithm based on heuristic principles was proposed. By specifying the direction priority, the path of the robot body can be autonomously planned according to the distribution of wall obstacles. Using the proposed dead zone escape strategy can ensure the shortest escape path and maximize cleaning efficiency.

A series of experiments were tailored to verify the performance of the robot and the effectiveness of the algorithm. The results show that the robot can avoid obstacles and achieve full-coverage movement on the complex façades of the experimental platform. The proposed algorithm planned an efficient global cleaning path for a rope-driven robot with a maximum position error of 7.3 mm. According to these results, the proposed priority heuristic algorithm increases the robot operation coverage rate on complex walls by independently planning the work path according to the distribution of obstacles.

This study provides a new idea for robot structure design and path planning for complex façade cleaning, which can be widely used in other scenes operated by rope-driven or crawling robots. Different from the existing methods that deal with planar CPPS,

we expanded the surface to a plane and then solved the global cleaning path planning problem considering obstacles, so no comparison with traditional CPPS was possible. In future work, we will compare our method with existing CPP techniques for flat surfaces and further improve the obstacle avoidance performance of the robot.

Author Contributions: Conceptualization, D.Z.; formal analysis, D.Z. and Y.L.; resources, M.L. and M.Z.; funding acquisition, M.L. and M.Z.; methodology, Y.Z. and Z.L.; investigation, P.J. and J.W.; validation, X.J. and G.W.; writing—original draft, D.Z. and Y.L.; writing—review and editing, P.J. and J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by the National Natural Science Foundation of China (grant numbers: U1913211, 52275016, and 52275017), the Science and Technology Development Fund Project on Central Government Guiding Local Government (grant numbers: 226Z1801G and 226Z1811G), the Natural Science Foundation of Hebei Province (grant numbers: F2021202016, F2021202062, and E2022202130), the Interdisciplinary Postgraduate Training Funding Project of Hebei University of Technology (grant number: HEBUT-Y-XKJC-2021108), and the State Key Laboratory of Reliability and Intelligence of Electrical Equipment (grant number: EERI_OY2021004).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data from in this study are available in the article.

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

References

- Feng, C.; Ma, F.; Wang, R.; Zhang, L.; Wu, G. The operation characteristics analysis of a novel glass curtain wall system by using simulation and test. *J. Build. Eng.* **2022**, *51*, 104311. [[CrossRef](#)]
- Li, H.; Huang, X.; Jin, S.; Jiang, Z.; Wang, B. Reliability and sensitivity analysis of cold-bent curtain wall glass. *J. Build. Eng.* **2022**, *49*, 104116. [[CrossRef](#)]
- Li, S.; Chen, S. Field monitoring and prediction on temperature distribution of glass curtain walls of a super high-rise building. *Eng. Struct.* **2022**, *250*, 113405. [[CrossRef](#)]
- Boafo, F.E.; Kim, J.-H.; Ahn, J.-G.; Kim, S.-M.; Kim, J.-T. Slim curtain wall spandrel integrated with vacuum insulation panel: A state-of-the-art review and future opportunities. *J. Build. Eng.* **2021**, *42*, 102445. [[CrossRef](#)]
- Lee, A.D.; Alimanza, J.A.; Shepherd, P.; Evernden, M.C. Axial Rotation and Lateral Torsional Buckling of Extruded Aluminium Mullions in Curtain Wall Facades. *Structures* **2019**, *20*, 658–675. [[CrossRef](#)]
- Schmidt, D.; Berns, K. Climbing robots for maintenance and inspections of vertical structures—A survey of design aspects and technologies. *Robot. Auton. Syst.* **2013**, *61*, 1288–1305. [[CrossRef](#)]
- Kim, S.-H.; Lee, K.-O.; Chae, Y.-J.; Lee, H.-W.; Park, Y.-Y. A Topology Map Generation Algorithm for Optimal Path Finding for Image-Based Maps. *Appl. Sci.* **2022**, *12*, 12436. [[CrossRef](#)]
- Van Dang, C.; Ahn, H.; Lee, D.S.; Lee, S.C. Improved Analytic Expansions in Hybrid A-Star Path Planning for Non-Holonomic Robots. *Appl. Sci.* **2022**, *12*, 5999. [[CrossRef](#)]
- Kim, H.; Choi, Y. Autonomous Driving Robot That Drives and Returns along a Planned Route in Underground Mines by Recognizing Road Signs. *Appl. Sci.* **2021**, *11*, 10235. [[CrossRef](#)]
- Lee, Y.-S.; Kim, S.-H.; Gil, M.-S.; Lee, S.-H.; Kang, M.-S.; Jang, S.-H.; Yu, B.-H.; Ryu, B.-G.; Hong, D.; Han, C.-S. The study on the integrated control system for curtain wall building façade cleaning robot. *Autom. Constr.* **2018**, *94*, 39–46. [[CrossRef](#)]
- Shi, X.; Xu, L.; Xu, H.; Jiang, C.; Zhao, Z.; Guo, Y.; Chen, X. A 6-DOF humanoid wall-climbing robot with flexible adsorption feet based on negative pressure suction. *Mechatronics* **2022**, *87*, 102889. [[CrossRef](#)]
- Hu, J.; Han, X.; Tao, Y.; Feng, S. A magnetic crawler wall-climbing robot with capacity of high payload on the convex surface. *Robot. Auton. Syst.* **2022**, *148*, 103907. [[CrossRef](#)]
- Liu, Y.; Sun, S.; Wu, X.; Mei, T. A Wheeled Wall-Climbing Robot with Bio-Inspired Spine Mechanisms. *J. Bionic Eng.* **2015**, *12*, 17–28. [[CrossRef](#)]
- Wang, B.; Ni, Z.; Shen, Y.; Zhang, S.; Shen, Q.; Niu, X.W. Design and analysis of a wheel-leg compound variable curvature ship hull cleaning robot. *Ocean Eng.* **2022**, *266*, 112755. [[CrossRef](#)]
- Akinfiev, T.; Armada, M.; Nabulsi, S. Climbing cleaning robot for vertical surfaces. *Ind. Robot. Int. J.* **2009**, *36*, 352–357. [[CrossRef](#)]
- Seo, K.; Cho, S.; Kim, T.; Kim, H.S.; Kim, J. Design and stability analysis of a novel wall-climbing robotic platform (ROPE RIDE). *Mech. Mach. Theory* **2013**, *70*, 189–208. [[CrossRef](#)]
- Imaoka, N.; Roh, S.G.; Yusuke, N.; Hirose, S. SkyScraper-I: Tethered whole windows cleaning robot. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 5460–5465. [[CrossRef](#)]

18. Sûr Efficace Et Durable Avec Des Résultats Transparents, The Netherlands. 2019. Available online: <https://www.kiterobotics.com> (accessed on 15 December 2022).
19. Yoo, S.; Kim, T.; Seo, M.; Oh, J.; Kim, H.S.; Seo, T. Position-tracking control of dual-rope winch robot with rope slip compensation. *IEEE/ASME Trans. Mechatron.* **2021**, *26*, 1754–1762. [[CrossRef](#)]
20. Song, Y.S.; Arshad, M.R. Coverage path planning for underwater pole inspection using an autonomous underwater vehicle. In Proceedings of the IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS), Selangor, Malaysia, 22 October 2016; pp. 230–235. [[CrossRef](#)]
21. Zhang, L.; Zhang, L.; Liu, S.; Zhou, J.; Papavassiliou, C. Three-Dimensional Underwater Path Planning Based on Modified Wolf Pack Algorithm. *IEEE Access* **2017**, *5*, 22783–22795. [[CrossRef](#)]
22. Li, J.; Zhang, J.; Zhang, H.; Yan, Z. A Predictive Guidance Obstacle Avoidance Algorithm for AUV in Unknown Environments. *Sensors* **2019**, *19*, 2862. [[CrossRef](#)]
23. Sun, B.; Zhu, D.; Tian, C.; Luo, C. Complete Coverage Autonomous Underwater Vehicles Path Planning Based on Glasius Bio-Inspired Neural Network Algorithm for Discrete and Centralized Programming. *IEEE Trans. Cogn. Dev. Syst.* **2019**, *11*, 73–84. [[CrossRef](#)]
24. Xia, W.; Yang, X. Research on Ant Colony Strategy with Multi-constraint and Multiresponse for Three-dimensional Route Planning of Underwater Platforms. *Acta Armamentarii* **2018**, *39*, 1795–1803. [[CrossRef](#)]
25. Yao, X.; Wang, F.; Wang, J.; Wang, X. Bilevel Optimization-Based Time-Optimal Path Planning for AUVs. *Sensors* **2018**, *18*, 4167. [[CrossRef](#)]
26. Chen, Z. State-Space Modeling and Target Detection and Tracking in Distributed Underwater Acoustic Networks. Ph.D. Dissertation, Zhejiang University, Hangzhou, China, 2019. [[CrossRef](#)]
27. Li, B.; Liu, H.; Su, W. Topology optimization techniques for mobile robot path planning. *Appl. Soft Comput.* **2019**, *78*, 528–544. [[CrossRef](#)]
28. Hao, Y.; Si, B.; Zhao, C. A novel shortest path algorithm with topology transformation for urban rail transit network. *Comput. Ind. Eng.* **2022**, *169*, 108223. [[CrossRef](#)]
29. Candeloro, M.; Lekkas, A.M.; Sørensen, A.J. A Voronoi-diagram-based dynamic path-planning system for underactuated marine vessels. *Control Eng. Pract.* **2017**, *61*, 41–54. [[CrossRef](#)]
30. Ayawli, B.B.K.; Appiah, A.Y.; Nti, I.K.; Kyeremeh, F.; Ayawli, E.I. Path planning for mobile robots using Morphological Dilation Voronoi Diagram Roadmap algorithm. *Sci. Afr.* **2021**, *12*, e00745. [[CrossRef](#)]
31. Wang, Q.; Langerwisch, M.; Wagner, B. Wide Range Global Path Planning for a Large Number of Networked Mobile Robots based on Generalized Voronoi Diagrams. *IFAC Proc. Vol.* **2013**, *46*, 107–112. [[CrossRef](#)]
32. Choset, H.; Pignon, P. Coverage Path Planning: The Boustrophedon Cellular Decomposition. In *Field and Service Robotics*; Zelinsky, A., Ed.; Springer: London, UK, 1998; pp. 203–209. [[CrossRef](#)]
33. He, L.-L.; Liu, X. Complete coverage of mobile robot path planning algorithm study. *J. Mech. Des. Manuf.* **2021**, *3*, 280–284.
34. Tang, D.; Long, Z.; Tang, Y.J.; Pan, F.; You, C.K. Full ergodic path planning for tank Detection wall-climbing Robot. *Chin. J. Eng. Des.* **2020**, *27*, 10. [[CrossRef](#)]
35. Ma, Y.; Ping, P.; Shi, Q. Research on the Path Planning of Unmanned Sweepers Based on a Fusion Algorithm. *Appl. Sci.* **2023**, *13*, 2725. [[CrossRef](#)]
36. Chen, R.; Hu, J.; Zhong, X.; Zhang, M.; Zhu, L. A Cognitive Environment Modeling Approach for Autonomous Vehicles: A Chinese Experience. *Appl. Sci.* **2023**, *13*, 3984. [[CrossRef](#)]
37. Hernandez, E.; Carreras, M.; Ridao, P. A comparison of homotopic path planning algorithms for robotic applications. *Robot. Auton. Syst.* **2015**, *64*, 44–58. [[CrossRef](#)]
38. Galceran, E.; Carreras, M. A survey on coverage path planning for robotics. *Robot. Auton. Syst.* **2013**, *61*, 1258–1276. [[CrossRef](#)]
39. Glorieux, E.; Franciosa, P.; Ceglarek, D. Coverage path planning with targetted viewpoint sampling for robotic free-form surface inspection. *Robot. Comput. Integr. Manuf.* **2020**, *61*, 101843. [[CrossRef](#)]
40. Wang, Y.; He, Z.; Cao, D.; Ma, L.; Li, K.; Jia, L.; Cui, Y. Coverage path planning for kiwifruit picking robots based on deep reinforcement learning. *Comput. Electron. Agric.* **2023**, *205*, 107593. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.