

Article

Path Planning for Wheeled Mobile Robot in Partially Known Uneven Terrain

Bo Zhang ^{1,2} , Guobin Li ^{1,2}, Qixin Zheng ^{1,2}, Xiaoshan Bai ^{1,2,*} , Yu Ding ^{1,2} and Awais Khan ^{1,2} 

¹ College of Mechatronics and Control Engineering, Shenzhen University, Shenzhen 518060, China; zhangbo@szu.edu.cn (B.Z.); 2100292009@email.szu.edu.cn (G.L.); 1810294017@email.szu.edu.cn (Q.Z.); 1910294029@email.szu.edu.cn (Y.D.); awaiskhan@szu.edu.cn (A.K.)

² Shenzhen City Joint Laboratory of Autonomous Unmanned Systems and Intelligent Manipulation, Shenzhen University, Shenzhen 518060, China

* Correspondence: baixiaoshan@szu.edu.cn

Abstract: Path planning for wheeled mobile robots on partially known uneven terrain is an open challenge since robot motions can be strongly influenced by terrain with incomplete environmental information such as locally detected obstacles and impassable terrain areas. This paper proposes a hierarchical path planning approach for a wheeled robot to move in a partially known uneven terrain. We first model the partially known uneven terrain environment respecting the terrain features, including the slope, step, and unevenness. Second, facilitated by the terrain model, we use A* algorithm to plan a global path for the robot based on the partially known map. Finally, the Q-learning method is employed for local path planning to avoid locally detected obstacles in close range as well as impassable terrain areas when the robot tracks the global path. The simulation and experimental results show that the designed path planning approach provides satisfying paths that avoid locally detected obstacles and impassable areas in a partially known uneven terrain compared with the classical A* algorithm and the artificial potential field method.

Keywords: hierarchical path planning; uneven terrain; A* algorithm; Q-learning algorithm



Citation: Zhang, B.; Li, G.; Zheng, Q.; Bai, X.; Ding, Y.; Khan, A. Path Planning for Wheeled Mobile Robot in Partially Known Uneven Terrain. *Sensors* **2022**, *22*, 5217. <https://doi.org/10.3390/s22145217>

Academic Editors: Shuai Li, Dechao Chen, Vasilios N. Katsikis, Predrag Stanimirović, Dunhui Xiao and Mohammed Aquil Mirza

Received: 2 June 2022

Accepted: 9 July 2022

Published: 12 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 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

Mobile robots that are deployed for rescue missions in uneven cluttered terrains generally need to have the ability of autonomous navigation and path planning. However, it is challenging to plan a feasible path efficiently for a robot to move in uneven terrains due to the terrains' slope, step, and unevenness. In [1], a real-time obstacle avoidance method is proposed based on trajectory space, which considers the mobile robot's uncertainty. However, the uneven terrain modeled in [1] does not reflect a realistic terrain environment well. Some research has been conducted for robotic path planning on uneven terrains, such as the path planning for the Chang'e-4 lunar exploration rover Yutu-2 to move through uneven rough terrain [2]. The Yutu-2 lunar rover can passively adapt to the uneven terrain on the moon's far side by using its differential mechanism and rocker arm. This configuration enables the rover to reduce its pitch angle by half compared with other vehicles when clearing an obstacle. The Curiosity rover used in the US Mars exploration mission is a six-wheeled vehicle [3]. It uses a rocker arm steering structure in which two front wheels and two rear wheels are independent such that the rover can pivot steering. The Curiosity rover, meanwhile, relies on a six-wheeled primary and secondary joystick system to navigate over the uneven rocks of Mars. The above work focuses on vehicles' adaptation and safe travel over rough terrain through sophisticated mechanical mechanisms. However, it must be acknowledged that this is a complex and costly approach. The application of path planning techniques would be much more beneficial if they could be used to continually identify whether they can be safely navigated to plan safe and feasible paths in rough environments.

For the path planning of wheeled mobile robots in uneven terrains, a high-quality terrain model is the foundation for path planning as the fidelity of the model affects the applicability of the planned path [4]. A lot of work has been performed on modeling uneven terrain environments. In [5], Dupuis et al. constructed a three-dimensional (3D) terrain by using an ILRIS-3D laser scanner to measure the unevenness of the road surface. Vandapel et al. used a 3D laser radar to obtain the environmental data of uneven terrain, and classified the uneven terrain by learning rocks' characteristics [6]. Agrawal et al. equipped a stereo vision camera on a wheeled mobile robot to map unknown uneven terrain [7]. The wheeled mobile robot moves to a target position in a new environment, where real-time correction is made in the map construction process. Huber et al. modeled uneven terrain by collecting data from three parts: the internal environment map constructed from the depth camera data, the large-scale topographic map constructed from the aerial data, and the map built from the stereo data [8]. The above research work shows that the grid-based 3D elevation map can be used to model an uneven terrain.

The goal of the path planning for a robot operating in a challenging uneven terrain is to find the optimal feasible path that avoids impassable areas, including obstacles, and reduces failure risk due to the terrain's slope, step, and unevenness. Different kinds of algorithms have been designed for robot path planning, including the Dijkstra algorithm [9], A* algorithm [10], RRT algorithm [11], artificial potential field (APF) based algorithms [12], and intelligent algorithms such as genetic algorithm [13], ant colony algorithm [14], particle swarm optimization algorithm [15], and reinforcement learning based algorithms [16]. Pan and Xu used an improved A* algorithm for path planning in a 3D terrain, where a penalty function is used to guarantee the robot's stability [17]. Peng et al. applied an improved APF method in a 3D space and introduced a tangent point for obstacle avoidance [18]. The repulsive force generated by the tangential point can guide a robot to its goal position and reduce the algorithm's running time. Pan et al. proposed a hybrid genetic ant algorithm to improve the efficiency of path planning in a 3D terrain [19]. Josef and Degani proposed a deep reinforcement learning method for local path planning of a robot in unknown uneven terrain in [20]. In [21], a map-based offline path planning approach was designed to construct an initial path for quadrotor UAVs, followed by a vision-based obstacle detection method using optical flow for collision avoidance. Online and real-time processing experiments demonstrate the feasibility of the proposed method for path planning and autonomous navigation. A hybrid navigation system was developed in [22] for a two-wheel differential drive mobile robot that includes static-environment global path planning and dynamic-environment obstacle-avoidance tasks, where a multi-agent A-heuristic algorithm was proposed for finding the optimal obstacle-free path and a weighted-sum model was employed to adapt to the dynamic obstacles.

This paper studies the path planning for a wheeled robot that operates in a partially known uneven terrain with incomplete environmental information such as locally detected obstacles and impassable terrain areas. First, the A* algorithm is used for the robot's global path planning based on partially known environmental information. Then, an adaptable Q-learning method is designed for the robot's local path planning to avoid locally detected obstacles and impassable terrain areas when the robot tracks the global path. The main contributions of this paper are as follows. First, we modeled the partially known uneven terrain considering its slope, step, and unevenness. Second, to avoid the robot colliding with locally detected obstacles and entering impassable terrain areas, we proposed a hierarchical path planning approach by integrating the A* algorithm for the robot's global path planning with the Q-learning algorithm for the robot's local path planning. The proposed approach has satisfying performance compared with the classical A* algorithm and the artificial potential field method.

The rest of the paper is organized as follows. In Section 2, we model the uneven terrain environment and formulate the path planning problem. The A* algorithm is presented as the global path planning algorithm, and the Q-learning algorithm is introduced for the

local path planning in Section 3. Section 4 shows the numerical and experimental results of the proposed path planning approach. Section 5 concludes the paper.

2. Environmental Modeling and Problem Formulation

In this section, the 3D digital elevation model (DEM) is first employed to describe the detailed geological surface information of the uneven terrain as in [23]. The 3D DEM can digitally represent the surface information of uneven terrains by considering the terrains' elevation data. The terrain elevation data can be collected from different types of sensors, such as LIDAR, depth camera, ultrasonic, and terahertz [24]. Second, facilitated by the environmental modeling, we formulate the path planning problem and present the kinematic model of the wheeled robot.

2.1. Digital Representation of 3D DEM

The 3D DEM is a raster data model that stores the elevation information of a set of terrain positions in digital form to digitally express the morphology of uneven terrain surfaces. It describes the spatial distribution of an uneven terrain environment by using a set of 3D vectors $\{x, y, z\}$, where x and y represent the plane position of mapping an environmental feature point of the uneven terrain, and z is the height of the point. Therefore, geomorphic characteristics such as the slope aspect, slope change rate, step, and unevenness can be well extracted by 3D DEM [25].

2.2. Information Extraction of a Partially Known Uneven Terrain Environment

This section models the partially known uneven terrain environment considering its geomorphic features such as slope, step, and unevenness. Slope describes the inclining degree of the uneven terrain, where a greater slope value implies a more inclining uneven terrain that is more difficult for a wheeled mobile robot to pass. Step indicates the change rate of the vertical height of the uneven terrain within a certain horizontal distance. The larger the step value is, the more likely the wheeled mobile robot will overturn. The unevenness describes the jitter degree of the terrain's elevation, where a greater unevenness value would lead to a more unstable motion of the wheeled mobile robot.

The terrain's slope actually represents the angle between the terrain plane and the horizontal plane, which can be calculated by the angle between the vertical Z axis and the normal vector of the terrain plane. Assume that N^2 points are sampled, where the sampling point located at the i -th row and j -th column of the raster map is $(x_{i,j}, y_{i,j}, z_{i,j})$, $i, j \in S$, $S = \{1, 2, \dots, N\}$, and $z_{i,j}$ is the point's elevation value. Then, the equation of the terrain plane can be defined as

$$Ax + By + C = z, \quad (1)$$

where A , B , and C are the coefficients to be solved. Meanwhile, the least squares equation is obtained as

$$F = \sum_{i,j \in S} (Ax_{i,j} + By_{i,j} + C - z_{i,j})^2. \quad (2)$$

The partial derivatives of F with respect to A , B , and C are respectively zero, where the equation of the terrain plane can be achieved after calculating A , B , and C using the least square regression. The normal vector of the terrain plane is $\vec{n} = (A, B, -1)$, whereas for the horizontal plane it is $\vec{m} = (0, 0, 1)$. The angle between the normal vector and horizontal plane of grid i is defined as φ , satisfying the angle formula between a line and plane as follows

$$\sin \varphi_i = \frac{|\vec{n} \cdot \vec{m}|}{|\vec{n}| \cdot |\vec{m}|}. \quad (3)$$

The slope angle of the grid i of the terrain plane is

$$\theta_i = \frac{\pi}{2} - \sin^{-1} \frac{|\vec{n} \cdot \vec{m}|}{|\vec{n}| \cdot |\vec{m}|}. \quad (4)$$

The unevenness of the uneven terrain dramatically impacts the state stability of the wheeled robot as the robot might roll over if the terrain's unevenness is too large. In this paper, the mean-square deviation of the road elevation is used to quantify the unevenness information considering the influence of the wheel diameter of the wheeled robot. Then, the unevenness value of the current grid/point i is

$$\omega_i = D^{-1} \sqrt{\frac{1}{9} \sum_{J \in U_i} (z(J) - \bar{z})^2}, \quad (5)$$

where D is the wheel diameter of the wheeled mobile robot, $z(J)$ is the elevation value corresponding to a plane position J in set U_i , and \bar{z} is the mean elevation value of all the plane positions in U_i . U_i contains the plane position information of the current grid i and its eight neighbor grids.

2.3. Objective Function

The traversability of the i -th grid of the map is indicated as $T_i \in [0, 1]$, which represents the easiness for the robot to pass considering the grid's slope, step, and unevenness as

$$T_i = k_1 \cdot \frac{\theta_i}{\theta_{crit}} + k_2 \cdot \frac{\delta_i}{\delta_{crit}} + k_3 \cdot \frac{\omega_i}{\omega_{crit}}. \quad (6)$$

The sum of the three non-negative parameters k_1, k_2, k_3 is 1, and $\theta_{crit}, \delta_{crit}, \omega_{crit}$ respectively represent the maximum unevenness characteristic factors that the robot can pass through each grid. In particular, the step value δ_i is the maximum z-direction elevation difference between the i -th grid and its adjacent eight grids. When any grid i 's slope value θ_i , step value δ_i , or unevenness value ω_i reaches its maximum, set $T_i = 1$, implying that the grid is impassable.

Let the traversability cost for traveling between the two grids i and j be

$$T_{i,j} = \sqrt{(T_i - T_j)^2}. \quad (7)$$

Let the Euclidean distance between two grids i and j be $l_{i,j}$. Then, the objective function of the path planning problem considers both the distance and traversability cost of the planned path as

$$f = \sum_{k=1}^{M-1} (l_{k,k+1} + w \cdot T_{k,k+1}). \quad (8)$$

where w is the parameter used to adjust the weight of the traveled distance and the road traversability, k is the k -th grid on the planned path, and M is the total number of grids traversed by the planned path. The second term of Equation (8) considers the traversability cost of the planned path, which can enable the robot to escape impassable areas. The travel cost f is positive-infinitely large if any grid i of the planned path satisfies $T_i = 1$.

2.4. Robot Kinematic Model

As [26], we assumed that the kinematics of the four-wheel mobile robot is

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta}_a \end{bmatrix} = \begin{bmatrix} \cos(\theta_a) & 0 \\ \sin(\theta_a) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ m \end{bmatrix}, \quad (9)$$

where v, m, x, y , and $\theta_a \in [-\pi, \pi]$ respectively represent the robot's tangential velocity, angular velocity, horizontal displacement, vertical displacement, and yaw angle. In the

simplified differential car model, the right and left wheel velocities are defined as v_R and v_L . Then, v and m in Equation (9) can be calculated as

$$v = \frac{v_L + v_R}{2}, \quad m = \frac{v_R - v_L}{\lambda}, \quad (10)$$

where λ is the distance between the two wheels and $v \leq v_{\max}$, $|m| \leq \frac{2v_{\max}}{\lambda}$. v_{\max} is the robot's maximum speed.

3. Path Planning Algorithm for Mobile Robot

Path planning for autonomous mobile robots in a partially known environment generally contains two procedures: global path planning and local path planning. Global path planning is a prior planning for a global path for the robot to move from a start position to a specific goal position based on prior known environmental information. The applicability of the planned global path depends on the accuracy of the environmental information. If some initially unknown obstacles and impassable terrain areas are detected by the robot's on-board sensors when tracking the global path, local path planning is needed to dynamically adjust the robot's path.

3.1. A* Algorithm for Global Path Planning

The classical A* algorithm is a graph search algorithm typically used for global path planning [27]. It can achieve the shortest path for a robot to move between two prescribed positions in a known static environment.

To enable the robot to automatically avoid impassable areas due to slope, step and unevenness, in this paper the cost function of A* considering the traversability cost is defined as

$$F(n) = G_s(n) + H_g(n), \quad (11)$$

where n is the index of the current grid reached by the planned path, $G_s(n)$ is the minimum accumulated cost for the robot to move from its start grid s to the current grid n , and $H_g(n)$ is the estimated minimum cost for the robot to move from n to the goal grid g .

The cost function $F(n)$ is applied to the A* algorithm for the global path planning in a 3D uneven terrain. According Equation (8), the accumulated travel cost $G_s(n)$ considering the influence of the terrain's slope, step, and unevenness is

$$G_s(n) = L_{sn} + w \cdot T_{sn}, \quad (12)$$

where $L_{sn} = \sum_{k=1}^{n-1} l_{k,k+1}$ is the accumulated Euclidean distance for the robot to travel from the start grid s to the current grid n , and $T_{sn} = \sum_{k=1}^{n-1} T_{k,k+1}$ is the accumulated traversability cost for the robot to travel between the two grids.

To guarantee that the A* algorithm is optimal, the heuristic function $H_g(n)$ can be set as the Euclidean distance $l_{n,g}$ between n and g .

3.2. Q-Learning Based Local Path Planning Algorithm

In a known static environment, the wheeled mobile robot can safely track the initially planned global path resulting from the A* algorithm. However, in a partially known uneven environment, the robot might encounter locally detected obstacles and impassable terrain areas that can only be detected within the sense distance of the robot. If the wheeled mobile robot cannot avoid these locally detected obstacles and impassable terrain areas well, it might fail to reach its destination. In [28], a local path planning method was designed based on a Q-learning algorithm for a robot to avoid locally detected obstacles.

According to [29], the Q-learning algorithm can optimize the local path of a wheeled mobile robot whenever an obstacle or impassable area located on the global planned path is detected through the robot's interactive exploration and evaluation. The locally planned path resulting from the collision avoidance maneuver is latched to the nearest state on

the already computed global path. A higher feedback reward of performing an action in the environment implies a better action: the action will be performed more in the future, and vice versa. The feedback of the unknown uneven terrain environment is obtained through constant trial and error. Inspired by [30], we design the schematic diagram of the interaction between the robot and the environment in Figure 1.

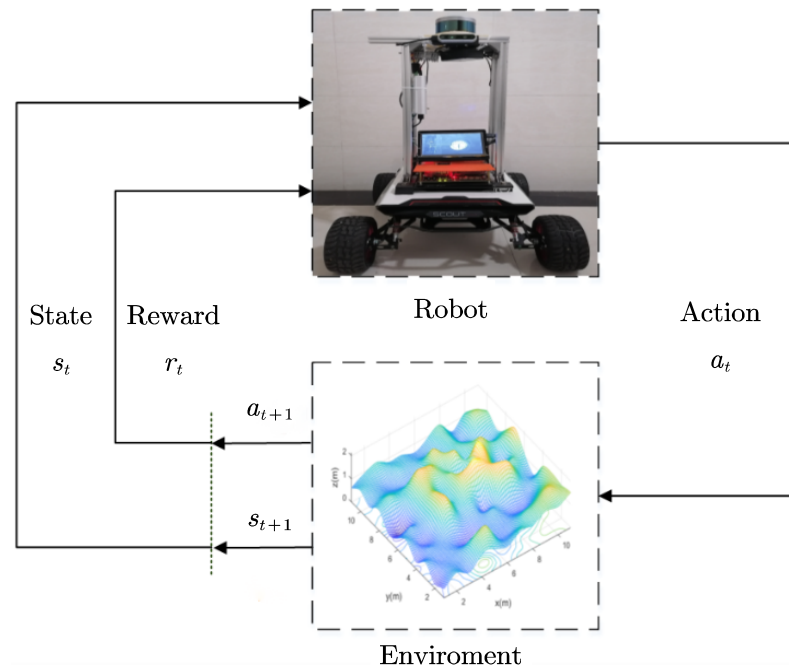


Figure 1. Schematic diagram of interaction between the robot and environment in reinforcement learning.

Q-learning algorithm is a value-based reinforcement learning algorithm, where Q denotes the corresponding action value the robot can obtain by taking a specified action at a certain moment. Inspired by [31], we construct the Q-learning algorithm in Algorithm 1, which uses a matrix Q to store the Q values resulting from the robot's action a_t at each state/grid. The robot's action space at each grid contains eight moving directions: up, down, left, right, upper-left, upper-right, lower-right, and lower-left. The matrix Q is employed to evaluate the corresponding actions to be taken in each current state: if the Q value for performing an action at a state is higher, it is better to take action at this state, and vice versa. For each time step t as shown in Algorithm 1, the robot chooses an action $a_t \in A$ under its current state $s_t \in S$ based on the ϵ -greedy strategy. The strategy maps the relationship between state s_t and action a_t . Meanwhile, the robot obtains a reward r_t according to the reward function and evolves to the next state s_{t+1} based on the current state s_t and action a_t . In an episode of Algorithm 1, the robot continues the cyclic process shown in Figure 1 until one of the robot's following situations happens: (1) getting to the goal position; (2) colliding with an obstacle; and (3) moving into an impassable grid. When an obstacle is detected, the reward for the robot entering the obstacle area is set to -1 , implying a punishment is incurred if the robot encounters the obstacle.

Algorithm 1 Q-learning algorithm

Initialize $Q_t(s_t, a_t), \forall s_t \in S, a \in A(s)$, maximum training episode number E ,
 $Q(\text{terminal state}, \cdot) = 0$

```

1: for every episode value of  $\{1, 2, \dots, E\}$  do
2:   Initialize state  $s_{t=0}$ 
3:   while the robot does not reach the goal position and does not collide with an
      obstacle and does not move into an impassable grid do
4:     Use  $\epsilon$ -greedy policy to select an action  $a_t$ 
5:     obtain the corresponding reward  $r_t$  and new state  $s_{t+1}$  based on the current state
       $s_t$  and action  $a_t$ 
6:      $Q_t(s_t, a_t) = (1 - \alpha)Q_t(s_t, a_t) + \alpha \left( r_t + \gamma \max_{a \in A} Q_{t+1}(s_{t+1}, a) \right)$ 
7:      $t \leftarrow t + 1$ 
8:   end while
9: end for

```

After the training by using Q-learning algorithm, the Q-value matrix Q can be obtained. Then, the robot's action with the maximum Q value at each current state can be chosen until reaching the goal position. Based on the Q-learning algorithm designed in [32], when the robot performs an action a_t in state s_t , the corresponding action value function $Q_t(s_t, a_t)$ can be updated as

$$Q_t(s_t, a_t) = (1 - \alpha)Q_t(s_t, a_t) + \alpha \left(r_t + \gamma \max_{a \in A} Q_{t+1}(s_{t+1}, a) \right), \quad (13)$$

where $\gamma \in [0, 1]$ is the attenuation rate representing the attenuation of future rewards, and $\alpha \in (0, 1)$ is the learning rate. The attenuation rate affects the ratio that the robot replaces the original Q value with a new value. In addition, the reward function r_t in Equation (13) is defined as

$$r_t = \begin{cases} v_{\text{reward}}, & \text{if robot gets to the goal position,} \\ v_{\text{penalty}}, & \text{if robot collides to an obstacle,} \\ -G_s(n), & \text{others.} \end{cases} \quad (14)$$

where v_{reward} is a positive value indicating a reward, and v_{penalty} is a negative value indicating a penalty. When the robot cruises on an uneven terrain, the reward $G_s(n)$ is calculated according to Equation (12).

The term $\max_{a \in A} Q_{t+1}(s_{t+1}, a)$ is the optimal action value of $Q_{t+1}(s_{t+1}, a)$ at the next time step corresponding to all possible actions a , which is denoted as

$$Q_{t+1}^*(s_{t+1}, a_{t+1}) = \max_{a \in A} (Q_{t+1}(s_{t+1}, a)). \quad (15)$$

$Q_{t+1}^*(s_{t+1}, a_{t+1})$ can be assumed to remain constant for future determined state s_{t+1} corresponding to the optimal action a_{t+1} .

Through a certain number of episodes of learning and training as shown in Algorithm 1, the robot learns new knowledge by constantly interacting with the environment until the convergence of the Q values $Q_t^*(s_t, a_t)$. Based on Equation (13), we can obtain

$$Q_t^*(s_t, a_t) = (1 - \alpha)^n Q_t(s_t, a_t) + (1 - \alpha)^{n-1} \alpha (r_t + \gamma Q_{t+1}^*(s_{t+1}, a_{t+1})) + \dots + \alpha (r_t + \gamma Q_{t+1}^*(s_{t+1}, a_{t+1})). \quad (16)$$

Then, it is straightforward that

$$Q_t^*(s_t, a_t) = (1 - \alpha)^n Q_t(s_t, a_t) + \alpha (r_t + \gamma Q_{t+1}^*(s_{t+1}, a_{t+1})) \sum_{i=0}^{n-1} (1 - \alpha)^i. \quad (17)$$

When n is an infinite large number, $(1 - \alpha)^n$ approaches to zero, and $\sum_{i=0}^{n-1} (1 - \alpha)^i$ approaches to $1/\alpha$ as $(1 - \alpha) \in (0, 1)$. Then, equation Equation (17) leads to

$$Q_t^*(s_t, a_t) = r_t + \gamma Q_{t+1}^*(s_{t+1}, a_{t+1}). \quad (18)$$

According to Equation (13), we have

$$Q_t(s_t, a_t) = Q_t(s_t, a_t) + \alpha(r_t + \gamma Q_{t+1}^*(s_{t+1}, a_{t+1}) - Q_t(s_t, a_t)). \quad (19)$$

Combining Equations (18) and (19), when the learning process converges, we obtain

$$Q_t(s_t, a_t) = Q_t^*(s_t, a_t). \quad (20)$$

4. Simulation and Experimental Tests

We evaluate the performance of the proposed path planning algorithms through both simulation and experimental tests. First, the simulation tests are performed on an Intel(R) Core (TM) i7-6700HQ CPU 2.60 GHz with 8.00 GB RAM, and the algorithms are compiled and implemented by MATLAB under Windows 10. To distinguish from the classical A* which only focuses on minimizing the length of the planned path, the A* algorithm considering the traversability cost of the planned path is simplified as an improved A* algorithm. The improved A* algorithm is first used to plan a global path for the robot moving in an uneven terrain based on partially known terrain information. The global path acts as a reference path for the robot to track. During the path-tracking process, the Q-learning algorithm is used to avoid locally detected obstacles and impassable terrain areas. The Q-learning algorithm is triggered whenever an obstacle or impassable area on the global planned path is detected. The maximum unevenness characteristic factors that the robot can pass through each grid are respectively $\theta_{crit} = 30^\circ$, $\delta_{crit} = 0.04$ m, and $\omega_{crit} = 0.025$ m according to the robot's physical structure. The range of the whole terrain map is $11 \text{ m} \times 11 \text{ m}$, where the size of the single grid is $0.1 \text{ m} \times 0.1 \text{ m}$. In subsequent experiments, the v_{reward} and $v_{penalty}$ in Equation (14) are set to 10 and -1 , respectively.

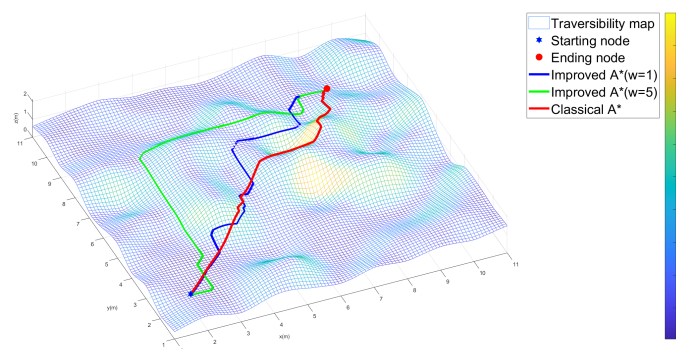
4.1. Global Path Planning Based on A* Algorithm

We first test the performance of the classical A* and the improved A* for path planning in 10 different scenarios of an initially known uneven terrain, where the start position and goal position of the global path planning are randomly generated in each scenario. The improved A* algorithm is performed under different weights w of Equation (8). In Equation (6), we set $k_1 = 0.2$, $k_2 = 0.4$, and $k_3 = 0.4$ since the step and unevenness generally play a more important role than slope.

Table 1 shows the average performance of the two algorithms, where the improved A* algorithm achieves feasible paths in all 10 scenarios while the classical A* algorithm fails to plan a feasible path in 7 out of 10 scenarios. The average path cost of the 3 feasible paths resulting from the classical A* algorithm is the sum of the initially minimized path length and the traversability cost of the planned path, which is larger than those resulting from the improved A* algorithm when $w = 1$. Furthermore, the travel cost of the planned path resulting from the improved A* algorithm increases with the increase in the weight w . This might be because a larger weight w mainly generates a smoother path while neglecting the planned path as shown in Figure 2, which is consistent with Equation (8). The traversability information of the uneven terrain is reflected by the color bar on the right part of Figure 2: a brighter color implies that it is more difficult for robot to travel on the uneven terrain, and a grid is unfeasible for the robot to move through if its associated color corresponds with value 1.

Table 1. The algorithms' average performance on global path planning in the 10 test scenarios.

| Algorithm | Weight w | Running Time (s) | Path Cost f | Feasible Times |
|--------------|------------|------------------|---------------|----------------|
| Classical A* | 0 | 1.42 | 34.33 | 3 |
| Improved A* | 0.2 | 1.31 | 28.12 | 10 |
| | 0.5 | 1.73 | 31.24 | 10 |
| | 1 | 1.81 | 32.17 | 10 |
| | 2 | 1.89 | 35.27 | 10 |
| | 4 | 1.96 | 38.24 | 10 |
| | 5 | 2.15 | 39.51 | 10 |

**Figure 2.** Global path planned by each algorithm in one scenario.

In Figure 2, the red-colored path results from the classical A* algorithm, which has the shortest length. However, the path is not feasible as it passes through the yellow area (impassable area), where the robot will roll over if tracking the planned global path. The green-colored path, resulting from the improved A* algorithm under $w = 5$, has a relatively smoother path through the uneven terrain compared with the case when $w = 1$. The blue-colored path, resulting from the improved A* algorithm under $w = 1$, has a shorter path length than the case when $w = 5$. However, the path passes over more terrain areas with a higher traversability cost, implying a higher chance of rollover if the robot tracks the planned global path.

4.2. Local Path Planning Based on Q-Learning Algorithm

For the local path planning, we test the algorithms' performance under four different scenarios compared with the artificial potential field method (APF) adapted from [33], the classical A* algorithm, and the improved A* algorithm proposed for global path planning. The start position, the goal position of the global path planning, and the position of an initially unknown obstacle are randomly generated in four scenarios. In Figure 3, the black-colored path is the global path initially planned by the improved A* algorithm with $w = 1$ based on the partially known environmental information in scenario 1. In Figure 3, an initially unknown yellow-colored cylindrical obstacle, locating on the global path, can be locally detected when the robot tracks the global path. The obstacle has a radius of 1 m and a height of 3 m. We assume that the sensing radius of the robot is 0.5 m, and the robot is driven by a constant speed of 1 m/s. It is straightforward to check that the wheeled mobile robot would collide with the locally detected obstacle if just tracking the global path.

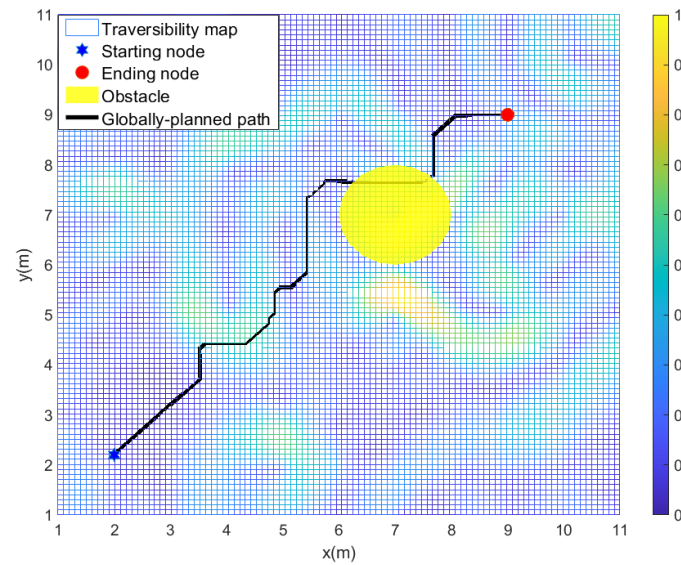


Figure 3. The globally planned path and locally detected obstacle in scenario 1.

Table 2 shows that the total path cost f in objective function Equation (8) corresponding to the local path resulting from the Q-learning algorithm in each scenario is the lowest compared with the APF, classical A* algorithm, and improved A* algorithm under different w . However, the payoff is the algorithm's longer running time. Table 2 also shows that the total path cost of the planned path resulting from the improved A* algorithm increases when increasing the weight of w , which is consistent with Table 1. Figure 4 and Table 2 show that the locally planned path resulting from the Q-learning algorithm has well considered the terrain's unevenness when avoiding the locally detected obstacle.

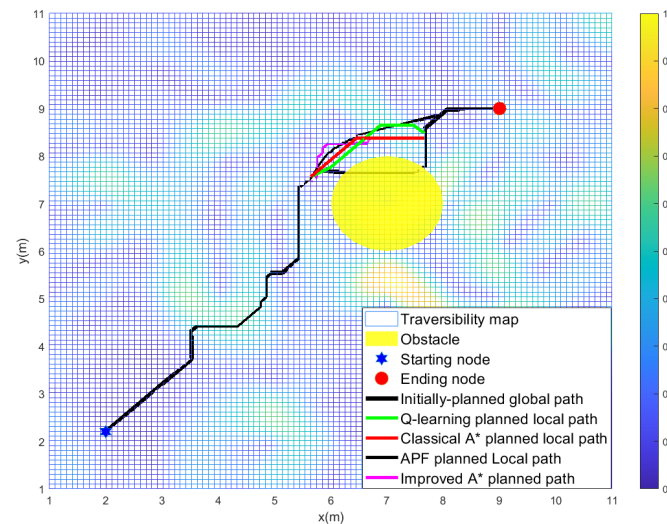


Figure 4. The locally planned path resulting from each algorithm in scenario 1.

Table 2. Overall performance of the algorithms.

| Scenarios | Local Path Planning Algorithm | Weight w | Running Time (s) | Path Cost f |
|------------|-------------------------------|---------------|------------------|---------------|
| Scenario 1 | Q-learning | 1 | 3.06 | 29.95 |
| | APF | - | 1.45 | 34.76 |
| | Classical A* | 0 | 0.26 | 37.98 |
| | Improved A* | 0.2 | 0.22 | 31.23 |
| | | 0.5 | 0.24 | 31.78 |
| | | 1 | 0.28 | 32.15 |
| | | 2 | 0.33 | 33.45 |
| | | 4 | 0.45 | 34.25 |
| | | 5 | 0.58 | 35.12 |
| Scenario 2 | Q-learning | 1 | 2.31 | 20.36 |
| | APF | - | 1.25 | 28.67 |
| | Classical A* | 0 | 0.23 | 32.82 |
| | Improved A* | 0.2 | 0.16 | 23.51 |
| | | 0.5 | 0.16 | 24.18 |
| | | 1 | 0.17 | 24.97 |
| | | 2 | 0.19 | 25.82 |
| | | 4 | 0.21 | 26.48 |
| | | 5 | 0.25 | 27.69 |
| Scenario 3 | Q-learning | 1 | 3.21 | 35.25 |
| | APF | - | 1.35 | 42.32 |
| | Classical A* | 0 | 0.31 | 43.58 |
| | Improved A* | 0.2 | 0.26 | 39.51 |
| | | 0.5 | 0.26 | 40.24 |
| | | 1 | 0.28 | 40.95 |
| | | 2 | 0.31 | 41.46 |
| | | 4 | 0.42 | 43.65 |
| | | 5 | 0.49 | 44.28 |
| Scenario 4 | Q-learning | 1 | 2.75 | 23.59 |
| | APF | - | 1.14 | 27.18 |
| | Classical A* | 0 | 0.28 | 31.57 |
| | Improved A* | 0.2 | 0.19 | 27.96 |
| | | 0.5 | 0.20 | 28.53 |
| | | 1 | 0.22 | 29.37 |
| | | 2 | 0.24 | 30.92 |
| | | 4 | 0.28 | 32.51 |
| | | 5 | 0.31 | 33.24 |

4.3. Experimental Test

This section experimentally tests the performance of the designed hierarchical path planning approach for guiding a wheeled mobile robot to move in a partially known uneven terrain. Figure 5 shows the Lidar-built DEM of the uneven terrain, where the performance of the algorithms is tested on two experimental scenarios Exp 1 and Exp 2, as shown in Table 3. In the two scenarios, one initially unknown obstacle is locally detected in Exp 1 as shown in Figure 6, whereas two initially unknown obstacles are consecutively detected in Exp 2 as shown in Figure 7.

Table 3. Two experimental scenarios.

| Exp | Start Position | Goal Position | Obstacle 1 | Obstacle 2 |
|-------|----------------|------------------|------------------|----------------|
| Exp 1 | (7.5, 7.5, 0) | (1.5, 7.5, 0.23) | (2.5, 3.5, 0.37) | - |
| Exp 2 | (7.5, 7.5, 0) | (2.5, 3.5, 0.37) | (7, 6.5, 0.02) | (5, 5.5, 0.18) |

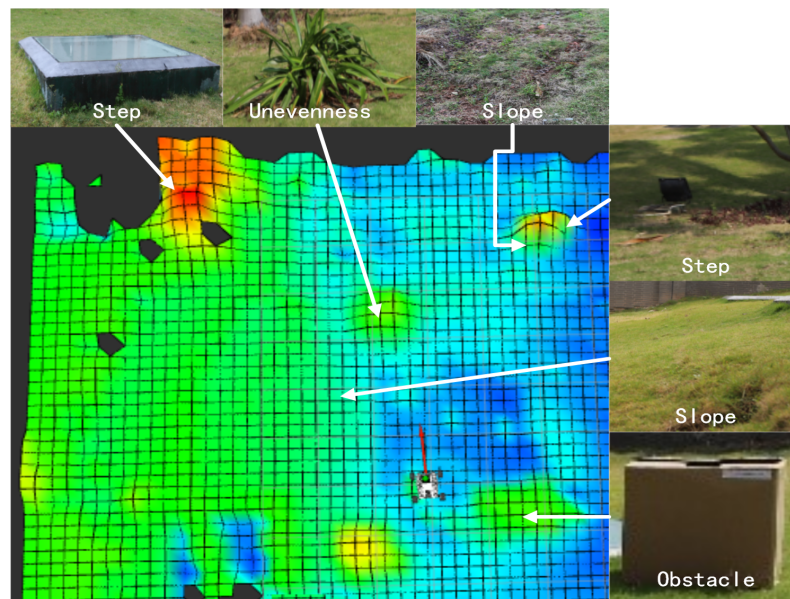


Figure 5. DEM of the experimental uneven terrain.

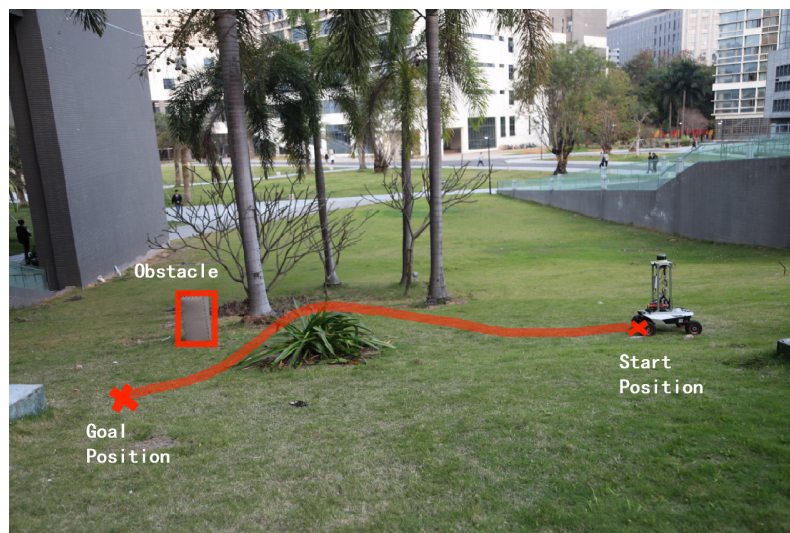


Figure 6. The robot's path in Exp 1 resulting from the integrated improved A* algorithm and Q-learning algorithm.

Figures 6 and 7 respectively show the robot's path guided by the integrated improved A* algorithm and the Q-learning algorithm. The two Supplementary Videos S1 and S2 show how the robot dynamically adjusts the initially planned global path to avoid locally detected obstacles in Exp 1 and Exp 2. It can be seen in the videos that the robot can move smoothly from a given start position to a goal position while dynamically adjusting its path from the initially planned global path to avoid locally detected obstacles.

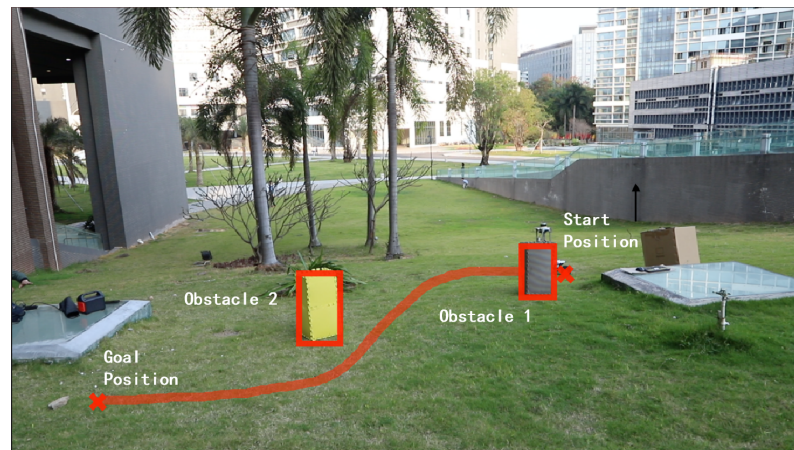


Figure 7. The robot's path in Exp 2 resulting from the integrated improved A* algorithm and Q-learning algorithm.

Figures 8 and 9 respectively show the globally planned paths resulting from the classical A* algorithm, the improved A* algorithm, and the robot's actual path resulting from the integrated improved A* algorithm and the Q-learning algorithm, which are compiled in MATLAB on experimental scenarios Exp 1 and Exp 2. Figures 8 and 9 show that the wheeled robot can successfully escape the locally detected obstacles when tracking the planned global path, which is consistent with Figures 6 and 7, respectively. The above simulation and experimental results demonstrate that the integrated A* algorithm and the Q-learning algorithm can enable the wheeled mobile robot to move smoothly in a partially known uneven terrain while dynamically adjusting its path to avoid locally detected obstacles and impassable terrain areas.

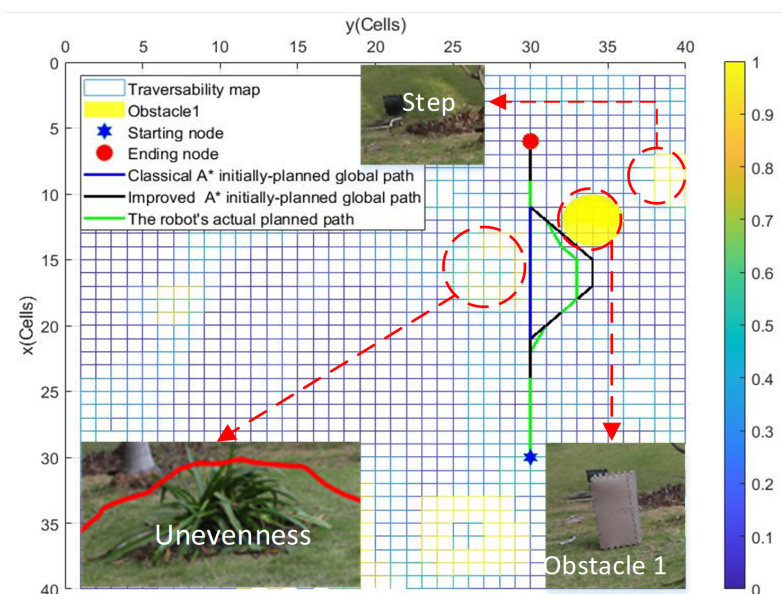


Figure 8. The planned paths in Exp 1.

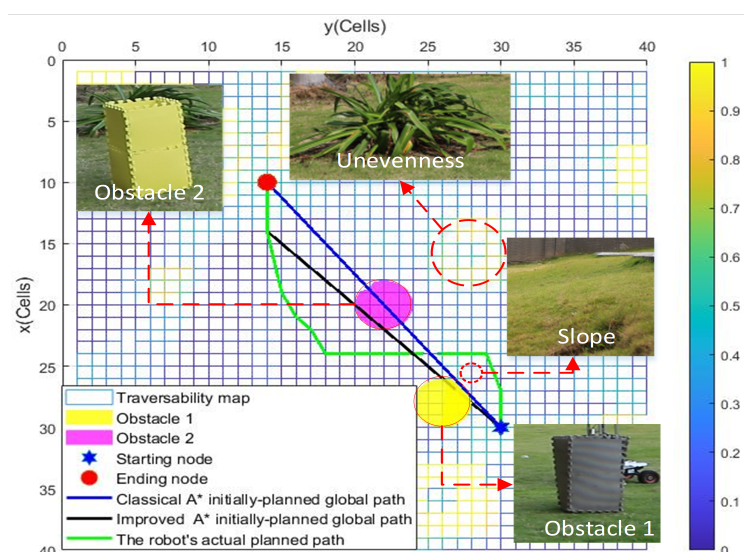


Figure 9. The planned paths in Exp 2.

5. Conclusions

This paper investigates the path planning of a wheeled mobile robot in a partially known uneven terrain. Based on the initial partially known environmental information, an improved A* algorithm is first used to achieve a global path for the robot to track considering the terrain's slope, step, and unevenness. Then, the Q-learning method is applied to adjust the robot's local path dynamically to avoid locally detected obstacles and impassable terrain areas when the robot tracks the global path. The simulation and experimental results show the satisfying performance of the integrated improved A* algorithm and Q-learning method for guiding the robot's movement in a partially known uneven terrain compared with the classical A* algorithm and the artificial potential field method. Future work will focus on improving the Q-learning method's efficiency for online obstacle-avoidance path planning.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/s22145217/s1>, Video S1: Exp 1. Video S2: Exp 2.

Author Contributions: All authors contributed to the study conception and design. Administrative support and provision of study materials were provided by B.Z., X.B. and A.K. Collection and assembly of data were carried out by G.L., Q.Z. and Y.D. Data analysis and interpretation were performed by B.Z., Q.Z., X.B., Y.D. and A.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported in part by the National Natural Science Foundation of China under Grants 62003217 and 62173234.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data generated or analyzed during the study of this manuscript are included in the article.

Conflicts of Interest: Relating to the content of this paper, all authors have no conflict of interest to declare.

References

1. Spenko, M.; Iagnemma, D.; Dubowsky, S. High speed hazard avoidance for mobile robots in rough terrain. *Unmanned Ground Veh. Technol. VI* **2004**, 5422, 439–450. [\[CrossRef\]](#)
2. Wang, Y.; Wan, W.; Gou, S.; Peng, M.; Liu, Z.; Di, K.; Li, L.; Yu, T.; Wang, J.; Cheng, X. Vision-based decision support for rover path planning in the Chang'e-4 Mission. *Remote Sens.* **2020**, *12*, 624. [\[CrossRef\]](#)

3. [EB/OL]. Available online: <http://mars.jpl.nasa.gov/MPF/index1.html> (accessed on 18 November 2021).
4. Liang, H.; Bai, H.; Sun, R.; Sun, R.; Li, C. Three-dimensional path planning based on DEM. In Proceedings of the 2017 36th Chinese Control Conference (CCC), Dalian, China, 26–28 July 2017; pp. 5980–5987.
5. Dupuis, E.; Allard, P.; Bakambu, J.; Lamarche, T.; Zhu, W.H.; Rekleitis, I. Towards autonomous long-range navigation. In Proceedings of the 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space, Munich, Germany, 5–8 September 2005; pp. 683–691.
6. Vandapel, N.; Huber, D.F.; Kapuria, A.; Hebert, M. Natural terrain classification using 3-d ladar data. In Proceedings of the IEEE International Conference on Robotics and Automation, ICRA'04, New Orleans, LA, USA, 26 April–1 May 2004; pp. 5117–5122.
7. Agrawal, M.; Konolige, K.; Bolles, R.C. Localization and Mapping for Autonomous Navigation in Outdoor Terrains: A Stereo Vision Approach. In Proceedings of the 2007 IEEE Workshop on Applications of Computer Vision (WACV'07), Austin, TX, USA, 20–21 February 2007; p. 7.
8. Huber, D.; Carmichael, O.; Hebert, M. 3D map reconstruction from range data. In Proceedings of the 2000 ICRA, Millennium Conference, IEEE International Conference on Robotics and Automation, Symposia Proceedings (Cat. No.00CH37065), San Francisco, CA, USA, 24–28 April 2000; pp. 891–897.
9. Stentz, A. Optimal and efficient path planning for partially known environments. In *Intelligent Unmanned Ground Vehicles*; Springer: Boston, MA, USA, 1997; pp. 203–220.
10. Hart, P.E.; Nilsson, N.J.; Raphael, B.A. Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. [[CrossRef](#)]
11. LaValle, S.M.; Kuffner, J.J. Randomized Kinodynamic Planning. *Int. J. Robot. Res.* **2001**, *20*, 378–400. [[CrossRef](#)]
12. Khatib, O. Real-time obstacle avoidance for manipulators and mobile robots. In *Autonomous Robot Vehicles*; Springer: New York, NY, USA, 1986; pp. 396–404.
13. Zhang, X.; Zhao, Y. Dynamic path planning algorithm for a mobile robot based on visible space and an improved genetic algorithm. *Int. J. Adv. Robot. Syst.* **2016**, *13*, 91. [[CrossRef](#)]
14. Liu, J.; Yang, J.; Liu, H.; Tian, X.; Gao, M. An improved ant colony algorithm for robot path planning. *Soft Comput.* **2017**, *21*, 5829–5839. [[CrossRef](#)]
15. Li, G.; Chou, W. Path planning for mobile robot using self-adaptive learning particle swarm optimization. *Sci. China Inf. Sci.* **2018**, *61*, 052204. [[CrossRef](#)]
16. Jaradat, M.A.K.; Al-Rousan, M.; Quadan, L. Reinforcement based mobile robot navigation in dynamic environment. *Robot. Comput.-Integr. Manuf.* **2011**, *27*, 135–149. [[CrossRef](#)]
17. Pan, S.; Xu, X. 2D and 3D Robot path planning based on the A* algorithm. *J. Jinggangshan Univ. (Natural Sci.)* **2015**, *36*, 84–88.
18. Peng, Y.; Guo, W.; Liu, M.; Cui, J.; Xie, S. Obstacle avoidance planning based on artificial potential field optimized by point of tangency in three-dimensional space. *J. Syst. Simul.* **2014**, *26*, 1758–1762.
19. Pan, X.; Wu, X.; Hou, X.; Feng, Y. Global path planning based on genetic-ant hybrid algorithm for AUV. *J. Huazhong Univ. Sci. Technol. (Nat. Sci. Ed.)* **2017**, *45*, 45–49.
20. Josef, S.; Degani, A. Deep Reinforcement Learning for Safe Local Planning of a Ground Vehicle in Unknown Rough Terrain. *IEEE Robot. Autom. Lett.* **2020**, *5*, 6748–6755. [[CrossRef](#)]
21. Lin, H.Y.; Peng, X.Z. Autonomous quadrotor navigation with vision based obstacle avoidance and path planning. *IEEE Access* **2021**, *9*, 102450–102459. [[CrossRef](#)]
22. Luan, P.G.; Thinh, N.T. Real-time hybrid navigation system-based path planning and obstacle avoidance for mobile robots. *Appl. Sci.* **2020**, *10*, 3355. [[CrossRef](#)]
23. Wang, Y.; Dou, W. A parallel algorithm of path planning for DEM terrain data. In Proceedings of the 2018 17th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES), Wuxi, China, 19–23 October 2018; pp. 22–25.
24. Mokrane, A.; Braham, A.C.; Cherki, B. UAV Path Planning Based on Dynamic Programming Algorithm On Photogrammetric DEMs. In Proceedings of the 2020 International Conference on Electrical Engineering (ICEE), Bandung, Indonesia, 23–24 September 2020; pp. 1–5.
25. Wermelinger, M.; Fankhauser, P.; Diethelm, R.; Krüsi, P.; Siegwart, R.; Hutter, M. Navigation planning for legged robots in challenging terrain. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 1184–1189.
26. Rösmann, C.; Hoffmann, F.; Bertram, T. Kinodynamic trajectory optimization and control for car-like robots. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 5681–5686.
27. Xie, W.; Fang, X.; Wu, S. 2.5 D Navigation Graph and Improved A-Star Algorithm for Path Planning in Ship inside Virtual Environment. In Proceedings of the 2020 Prognostics and Health Management Conference (PHM-Besançon), Besançon, France, 4–7 May 2020; pp. 295–299.
28. Sharma, A.; Gupta, K.; Kumar, A.; Sharma, A.; Kumar, R. Model based path planning using Q-Learning. In Proceedings of the 2017 IEEE International Conference on Industrial Technology (ICIT), Toronto, ON, Canada, 22–25 March 2017; pp. 837–842.
29. Song, L.; Li, C.; Wang, X.; Zhang, N.; Fu, H. On Local Path Planning for the Mobile Robot based on QL Algorithm. In Proceedings of the 2018 37th Chinese Control Conference (CCC), Wuhan, China, 25–27 July 2018; pp. 5293–5298.

30. Xin, J.; Zhao, H.; Liu, D.; Li, M. Application of deep reinforcement learning in mobile robot path planning. In Proceedings of the 2017 Chinese Automation Congress (CAC), Jinan, China, 20–22 October 2017; pp. 7112–7116.
31. Liu, X.; Yao, Z.; Wu, B.; Ling, H.; Zhu, L.; Zhang, J. Research On Path Planning Of Hull Decontamination Robot Based On Q-Learning. In Proceedings of the 2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL), Chongqing, China, 10–12 July 2020; pp. 338–341.
32. Gautam, U.; Malmathanraj, R.; Srivastav, C. Simulation for path planning of autonomous underwater vehicle using flower pollination algorithm, genetic algorithm and Q-learning. In Proceedings of the 2015 International Conference on Cognitive Computing and Information Processing (CCIP), Noida, India, 3–4 March 2015; pp. 1–5.
33. Triharminto, H.H.; Wahyunggoro, O.; Adj, T.B.; Cahyadi, A.I. An integrated artificial potential field path planning with kinematic control for nonholonomic mobile robot. *Int. J. Adv. Sci. Eng. Inf. Technol.* **2016**, *6*, 410–418. [[CrossRef](#)]