sensors

Article

Smart Vehicle Path Planning Based on Modified PRM Algorithm

Qiongqiong Li, Yiqi Xu, Shengqiang Bu and Jiafu Yang *

College of Mechanical and Electronic Engineering, Nanjing Forestry University, Nanjing 210037, China
* Correspondence: jfyang@njfu.edu.cn; Tel.: +86-13951004006

Abstract: Path planning is a very important step for mobile smart vehicles in complex environments. Sampling based planners such as the Probabilistic Roadmap Method (PRM) have been widely used for smart vehicle applications. However, there exist some shortcomings, such as low efficiency, low reuse rate of the roadmap, and a lack of guidance in the selection of sampling points. To solve the above problems, we designed a pseudo-random sampling strategy with the main spatial axis as the reference axis. We optimized the generation of sampling points, removed redundant sampling points, set the distance threshold between road points, adopted a two-way incremental method for collision detections, and optimized the number of collision detection calls to improve the construction efficiency of the roadmap. The key road points of the planned path were extracted as discrete control points of the Bessel curve, and the paths were smoothed to make the generated paths more consistent with the driving conditions of vehicles. The correctness of the modified PRM was verified and analyzed using MATLAB and ROS to build a test platform. Compared with the basic PRM algorithm, the modified PRM algorithm has advantages related to speed in constructing the roadmap, path planning, and path length.

Keywords: smart vehicle; probabilistic roadmap algorithm; pseudo-random sampling; collision detection; path smoothing

check for updates

Citation: Li, Q.; Xu, Y.; Bu, S.; Yang, J. Smart Vehicle Path Planning Based on Modified PRM Algorithm. *Sensors* **2022**, *22*, 6581. https://doi.org/10.3390/s22176581

Academic Editors: Luige Vladareanu, Hongnian Yu, Hongbo Wang and Yongfei Feng

Received: 11 July 2022
Accepted: 29 August 2022
Published: 31 August 2022

1. Introduction

In recent years, smart vehicles have received more attention with the development of emerging technologies such as cloud computing, big data, and the full-scale launch of 5G construction [1,2]. Smart vehicles have significant effects in relieving driving pressure, avoiding traffic jams, and reducing environmental pollution [3] Path planning and motion control are significant and complex navigation tasks in smart vehicles. Path planning technology is the basis of smart vehicles to make motion decisions and navigate positioning [4,5]. To achieve successful path planning and motion control to be able to reach a target safely, smart vehicles must be provided with the ability to perceive and detect obstacles to be avoided [6]. Many sensors are installed on the body of smart vehicles, which ensure that they can perceive and interpret information gathered from the environment to determine position, direction to the target, position of obstacles, and navigation in both structured or unstructured environments [7].A smart vehicle is expected to perform these tasks with the safest and shortest path, reaching the target in the shortest time, and ultimately performing the specified task without the intervention of humans. Path planning in smart vehicles refers to determining how the smart vehicle reaches its target point safely to ensure obstacle avoidance. Smart vehicle path planning is described as a multi-objective optimization problem as it requires the generation of appropriate trajectories as well as obstacle avoidance in the environment [8].

The methods of smart vehicle path planning can be classified in different ways. Ayawli et al. [7] categorized them into nature-inspired computation methods, traditional methods, and hybrid methods. Methods and strategies that imitate natural phenomena are described as nature-inspired computation methods. Meanwhile those that have nothing to do with imitating nature phenomena are described as the conventional method. Approaches that combine two or more strategies are described as hybrid methods. Nature-inspired computing consists of a metaheuristic algorithm that simulates, based on nature phenomena given by natural science [9]. A number of researchers have attempted to solve the problem of mobile robotics path planning by using nature-inspired algorithms including genetic algorithms (GA) [10,11], artificial neural networks (ANN) [12,13], simulated annealing (SA) [14], ant colony optimization (ACO) [15], particle swarm optimization (PSO) [16], and artificial bee colonies (ABC) [17].In order to take advantage of the strengths of some methods while reducing the effects of their disadvantages, some researchers combine two or more methods to provide an efficient hybrid path planning method for controlling smart vehicles. These approaches include APF combined with GA [18], APF combined with PSO [19], and fuzzy logic combined with Kalman filtering [20,21]. Conventional path planning methods have been used for many years. These methods mainly rely on distance information from the object to the smart vehicles, repulsive force and attractive force clustering, or graphical map calculations to determine the path planning of smart vehicles. Even though conventional methods of path planning are computationally expensive, they are easy to implement. Conventional methods mainly consist of the rapidly-exploring random tree (RRT) algorithm [22], probabilistic roadmap algorithm (PRM) [23], artificial potential field (APF) [24,25], sliding mode control (SMC) [11,26], A * algorithm [27], D * algorithm [28,29], and simultaneous localization and mapping (SLAM) [30].

PRM is one of the most popular sampling based planners. PRM is a space planner that uses multiple-query planning. The key idea in PRM is to distribute the nodes across the space and then connect these nodes using simple local planning and straight lines to form a graph roadmap. By connecting the available space, the PRM succeeds in exploring a faster path by reducing the search to a graph [31]. However, PRM has shortcomings, including lack of orientation in the selection of sampling points, low reuse rate of the roadmap, and low search efficiency. Moreover, due to the random sampling of nodes in PRM, there exists a narrow passage problem that generates an unconnected graph. To enhance the efficiency of sampling-based algorithms, Kantaros et al. [32] introduced bias into the sampling process. Vasile et al. [33] maintained sparsity of generated samples. Sparseness was also explored by Dobson and Berkis for PRM using different techniques [34]. Amato et al. [35] proposed parallelizing strategies; the PRM method has massive inherent parallelism, which can be easily and best exploited. Berkis et al. [36] used the probabilistic roadmap method (PRM) with bidirectional rapidly exploring random trees (BI-RRT) as the local planner to solve multiple queries for motion planning problems with single query planners. Kurniawait et al. [37] designed an improved PRM algorithm, which was based on obstacle boundary sampling and evaluated the optimal feasible region to optimize the dispersion of random sampling of the PRM algorithm. Esposito et al. [38] proposed a processing algorithm for optimizing probabilistic roadmaps. Dealing with the format of convex cells in free space with a number of nodes that requires a lot of computation, this algorithm could simplify the computation required for this step by sparse decomposition. Gao Junli et al. [39] proposed to combine the deep reinforcement learning twin-delayed deep deterministic policy gradient algorithm with the traditional PRM algorithm as a new path planner, and the experimental results showed that this incremental training mode could significantly improve search efficiency. Moreover, this new path planner effectively improved the generalization of the model. Chen Gang et al. [40] proposed an improved PRM method. Based on a virtual force field, a new sampling strategy of PRM was proposed to generate a configuration that is more appropriate for practical application in free space.

RAVANKAR et al. [41] proposed a method for global planning using a hierarchical hybrid PRM and the APF method, using a decomposition method of node distribution that used map segmentation to generate regions of high and low potential, and proposed a method to reduce the dispersion of sample sets during roadmap building. Xu Zhenfan et al. [42] changed the sampling strategy so that nodes were incrementally added and evenly distributed in the exploration region to produce the best viewpoints and PRM enabled the planner to quickly search for alternative paths and avoid dynamic obstacles for safe exploration.

Aiming to improve the shortcomings of the PRM algorithm, the main innovation of this paper is that we propose a pseudo-random sampling strategy with the main spatial axis as the reference axis, set the distance threshold between road points, and adopt a two-way incremental method for collision detections. We aim to find the shortest path between the start point and target point and shorten the time of the planning path. The key road points of the path are extracted as discrete control points of the Bessel curve. We use Bezier curve to make the path smoother, whereas the path is more like the actual driving condition of the smart vehicle.

2. Modified PRM Algorithm

2.1. PRM Algorithm

The PRM algorithm includes sampling and query phases.

Sampling phase: the PRM algorithm randomly samples in the planning space and judges the reasonableness of the sampling points by the local planner. By repeating the sampling times n to generate a collection of valid waypoints V traversing the V, the algorithm connects all the feasible paths between the waypoints to expand to the whole planning space and forms the waypoint graph. $V = \{v_1, .v_2, \ldots, v_n\}$ denotes the set of waypoints; $E = \{v_i, v_j | v_i, v_j \in V\}$ denotes the set of edges between waypoints.

Query phase: the start point q_{init} and target point q_{goal} are put into the wayfinding graph $G(V, E)$, and the algorithm enters the path search phase. We use the graph search algorithm in the wayfinding graph $G(V, E)$ to find a collision-free path connecting the start point q_{init} and target point q_{goal}.

2.2. Pseudo-Random Sampling

In the PRM algorithm, the number of sampling points generated by the random sampling strategy increases with an increase in planning space. It is difficult to achieve a global uniform distribution and easy to create redundancy in sampling points. There is a considerable probability that the shortest path occurs in the area where the starting point and target point connects. This region is regarded as a focused sampling region, referred to as the spatial principal axis region.

To construct the spatial principal axis information, we set the coordinates of the starting point to be $S(x_s, y_s)$ and the coordinates of the target point to be $G(x_g, y_g)$. Length L and declination of the spatial principal axis θ was denoted by:

$$L = \|G - S\|_2 \tag{1}$$

$$\theta = \frac{\pi}{2} - \arctan\frac{|y_g - y_s|}{|x_g - x_s|} \tag{2}$$

We designed the spatial principal axes with the length L, and number of sampling points n, then obtained the longitudinal sampling spacing N_d, as:

$$N_d = \frac{L}{n} \tag{3}$$

Referring to the random sampling method, the sampling points were symmetrically distributed in the sector area near the main axis of space, and sampling points $P_{i,j}(x, y)$ were calculated as follows:

$$x = x_s + r_d \times \cos(\theta + \phi_j) \tag{4}$$

$$y = y_s + r_d \times \sin(\theta + \phi_j) \tag{5}$$

$$r_d = i \times N_d, i = [1, 2, \ldots, n] \tag{6}$$

where (x_s, y_s) indicates the starting point of the intelligent vehicle; r_d indicates the sampling radius; sampling radius is centered on the starting point; $\phi_j \in [-\phi_m, \phi_m]$ indicates the angle of deflection of the sampling point and; ϕ_m indicates the maximum deflection angle. It is used to control the angle of the sector sampling area, that is, the range of lateral sampling.

According to Figure 1a,b, the sampling points are symmetrically distributed on both sides of the main spatial axis, and sampling range is controlled by the maximum deflection angle ϕ_m. With the increase of ϕ_m, the sampling points spread in all directions along the main spatial axis. To make the sampling point distribution more uniform, the lateral sampling range is adjusted along the main axis of space, and sampling range is adjusted in increments using $\Delta\phi = \phi_{\mathrm{m}}/n$. The distribution of sampling points after adjustment is shown in Figure 1c,d.

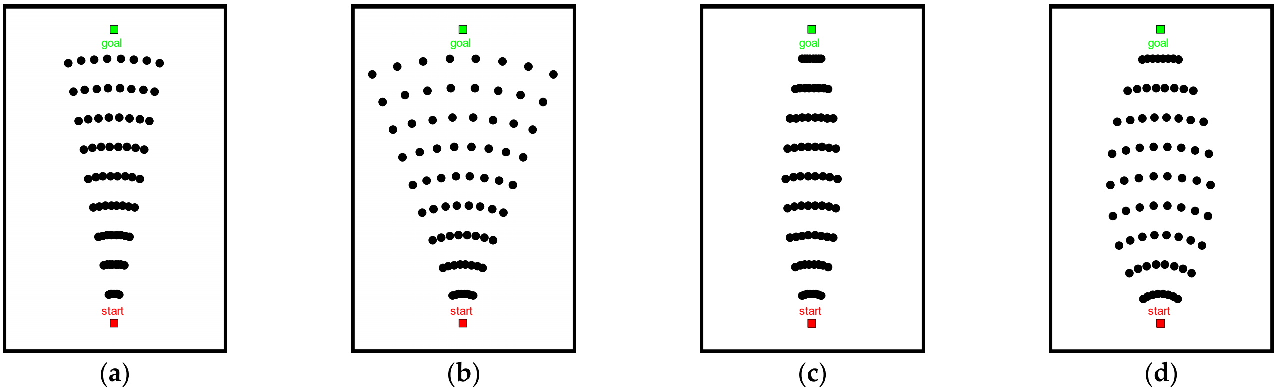


Figure 1. Sampling method based on spatial principal axis: (**a**) $\phi_m = 10$, (**b**) $\phi_m = 20$, (**c**) $\phi_m = 10$, and (**d**) $\phi_m = 20$.

Integrating the characteristics of uniform sampling, we counted the number of sampling points p in free space and the effective sampling rate of the horizontal sampling layer is defined as R:

$$R = \frac{p}{N} \tag{7}$$

where N indicates the total number of samples in the current sampling layer and the size of the effective sampling rate R reflects the connectivity of the current sampling layer. The larger R is, the better the connectivity of the sampling layer. If R is too small, this means that most of the sampling points in the sampling layer have fallen into the obstacle space. If the sampling layer edge subsequently has the same sampling interval, the chance of sampling points falling into the obstacle space will increase.

In order to improve the ability of the sampling points in avoiding obstacles, we introduced random increments Δr to adjust the sampling interval of sampling points. Based on Figure 1d, we adjust the size of the random increment Δr to get Figure 2. As the value of the random increment Δr increases, the sampling points tend to approach random distribution. With a decreasing value of Δr, the sampling points tend to approach uniform distribution.

Referring to Figure 3, hollow dots indicate the sampling points before adjusting the sampling spacing, solid dots indicate the adjusted sampling points, red markers represent the sampling points falling into the obstacle space, and black markers represent the sampling points in the free space. The effective sampling rate of the front sampling layer is low ($R = 0.3$), the radius fluctuation rate ($R = 0.8$) of the subsequent sampling layer is adjusted, and the sampling points avoid the obstacles by using the pseudo-random sampling strategy, which improves the quality of sampling point generation.

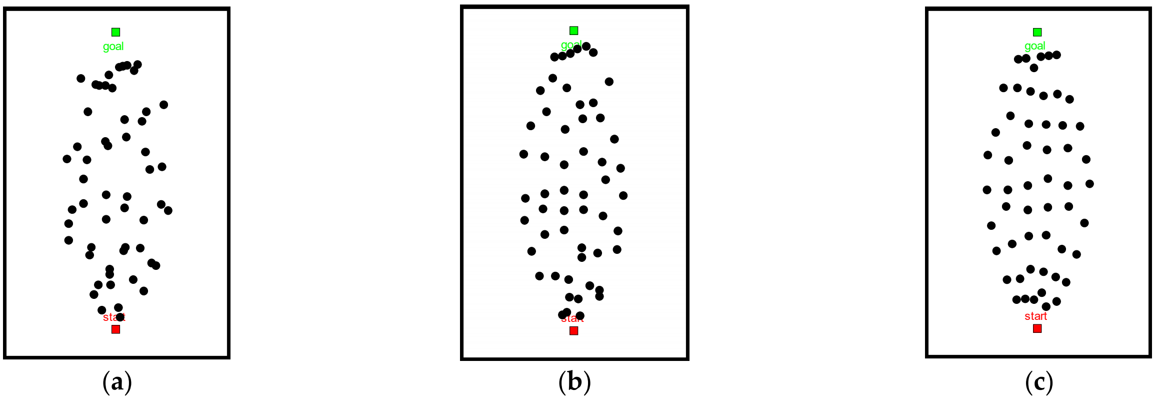


Figure 2. Pseudo-random-based sampling method: (**a**) $\Delta r = r$, (**b**) $\Delta r = 0.5r$, and (**c**) $\Delta r = 0.25r$.

The sampling radius after adding random increments Δr is shown in Equation (8):

$$r'_d = r_d + \Delta r \tag{8}$$

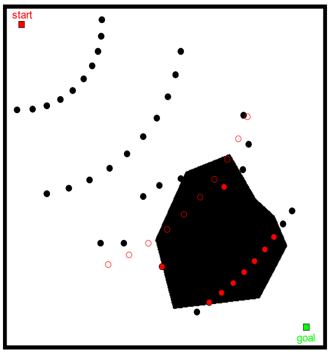


Figure 3. Schematic of sampling point adjustment.

2.3. Bidirectional Incremental Collision Detection

Collision detection is used to determine whether the connected line segments between the sample points intersect with the obstacle space, and the sample points are connected to each other by collision detection to form a roadmap $G(V, E)$. The traditional PRM algorithm usually takes an incremental detection strategy. According to a fixed step size, the planner selects discrete points and detects whether the point falls into the obstacle space. To improve the efficiency of collision detection execution, we combined this incremental detection method with the dichotomous method, proposing a two-way incremental detection strategy.

First, the two-way incremental detection method judges the reasonableness of the first and last connected sample points (Figure 4a). Then, we end the detection if the sample points belong to the obstacle space. If the sample points belong to the self-use space, we select the test point in both directions gradually along the first and last connected sample points and judge the reasonableness of the test point. If the selected test point belongs to the obstacle space, the detection is stopped to discard the path, as shown in Figure 4b. The sample points are connected to each other by collision detection, and finally form a roadmap $G(V, E)$.

2.4. Neighbouring Layer Connection Strategy

In the roadmap $G(V, E)$, the threshold distance between road points is an important factor affecting the efficiency of roadmap construction. The path formed by connecting road points in the same sampling layer is not conducive to shorten the global path length. Taking the distribution characteristics of the longitudinal sampling layer into account, we set the connection threshold of the longitudinal sampling spacing L_{TH} to screen the paths that met the threshold conditions and make the connection between road points from the full connection to adjacent sampling layer connection, improving roadmap construction efficiency.

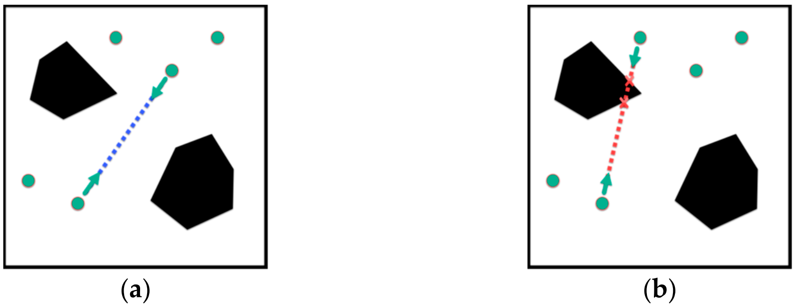


Figure 4. Schematic diagram of two-way incremental detection strategy: (**a**) reasonable path and (**b**) illegal path.

The sampling points generated based on the pseudo-random sampling strategy ($N = 20$) were selected to obtain the roadmap constructed under the drive of two connection strategies, as shown in Figure 5. Figure 5a shows the wayfinding graph generated by the full connectivity strategy, with the red solid line representing the filtered paths. Figure 5b indicates the wayfinding graph generated by the neighbouring layer connectivity strategy. In terms of time consumption, the composition time using these different connection strategies was 0.906 s and 0.437 s, respectively, and the latter optimized composition efficiency by 48.2%.

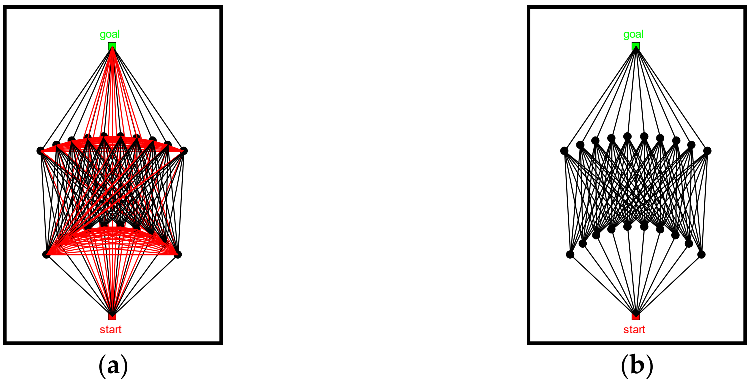


Figure 5. Comparison of road signs: (**a**) full connection and (**b**) neighbouring layer connection.

3. Path Smoothing

In this paper, Bessel curves were chosen to smooth the paths planned by the modified PRM algorithm.

The n order Bessel curve expressions were defined as:

$$B(t) = \sum_{i=0}^{n} P_i b_{i,n}(t),\ (t \in [0,1]) \tag{9}$$

where P_i represents the $n+1$ control point of the Bessel curve and $b_{i,n}(t)$ represents the Bernstein basis function. The value of this function is shown in Equation (10):

$$b_{i,n}(t) = C_n^i t^i (1-t)^{n-i} = \frac{n\,!}{(n-i)\,!i\,!} t^i (1-t)^{n-i}, i = 0,1,2,\ldots,n \tag{10}$$

In this paper, a 4th order Searle curve was chosen, and the formula is as follows:

$$B(t) = (1-t)^4 P_0 + 4P_1(1-t)^3 t + 6P_2(1-t)^2 t^2 + 4P_3(1-t)t^3 + P_4 t^4,\ t \in [0,1] \tag{11}$$

The curvature of the Bessel curve at any point $\kappa(t)$ is:

$$\kappa(t) = \frac{|B'(t) \times B''(t)|}{|B'(t)|^2} \tag{12}$$

Assuming that the planning path $path = \{P_n\}$ consists of a series of discrete points $(n \geq 5)$, the discrete points are used as the control points P_i of the Bessel curve, and the curvature of the Bessel curve $\kappa(P)$ can be obtained according to Equation (12):

$$\kappa(P) = \frac{P'_x P''_y - P'_y P''_x}{(P'^2_x + P'^2_y)^{3/2}} \tag{13}$$

The curvature of the Bessel curve at the starting point is $\kappa(0)$:

$$\kappa(0) = \frac{3|(P_1 - P_0) \times (P_2 - P_1)|}{4(P_1 - P_0)^3} \tag{14}$$

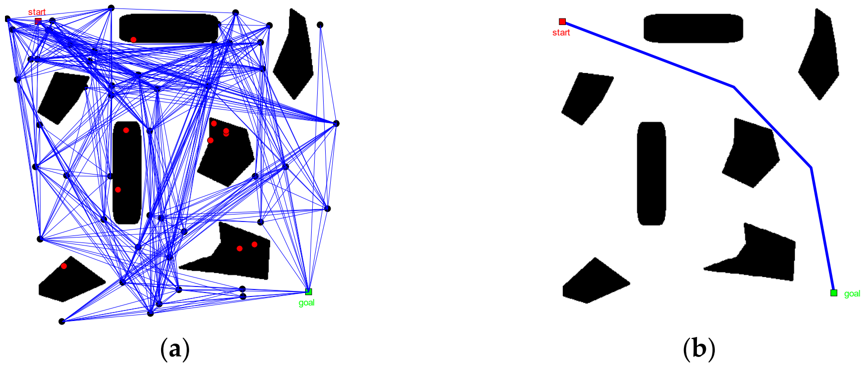
In this specific implementation, the key waypoints of the path searched by the modified PRM algorithm were extracted, discrete control points of the Bessel curve P_i were obtained by discretizing the line between key waypoints, and the discrete points were interpolated and fitted by Equation (9) to realize the smoothing of the path.

4. Simulation Test and Analysis

To verify the composition and path planning efficiency of the modified PRM algorithm, MATLAB (MATLAB2018b, MathWorks. Inc., Natick, MA, USA) was used to build a simulation experiment platform and a ROS (ROS1.0, Willow Garage. Inc., Menlo Park, CA, USA) experimental platform was used to verify the correctness of the modified PRM algorithm. Our computer configurations included: a Windows 10 operating system, 512 GB hard disk, and 8 GB RAM.

4.1. Comparison of Algorithm Composition Efficiency

The planning space of the known map is shown in Figures 6 and 7. The two algorithms kept the same total number $N = m \times n$ of sampling points in the sampling phase, where m and n represent the number of horizontal and vertical sampling points of the algorithm, respectively. We focused on the planning path length and roadmap construction time and repeated the test several times (recorded 10 times). The results are shown in Table 1 in mean values.



(**a**) (**b**)

Figure 6. Planning results of the basic PRM algorithm (N = 60): (**a**) roadmap and (**b**) planned path.

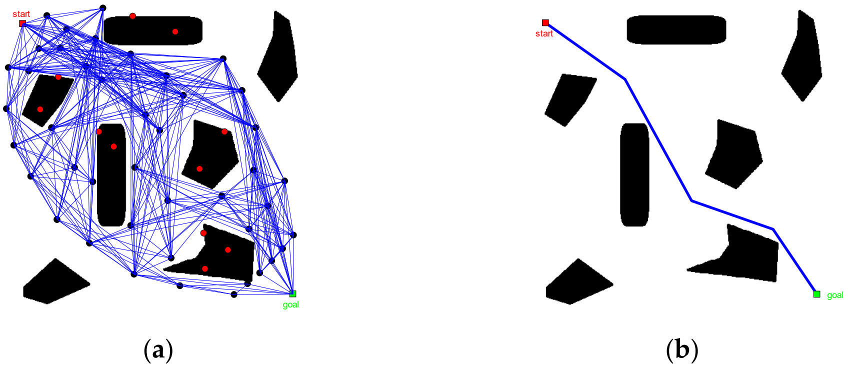


Figure 7. Planning results of the modified PRM algorithm ($N = 60$): (**a**) roadmap and (**b**) planned path.

Table 1. The results of algorithm comparison.

Algorithm Name	Number of Sampling Points N	Path Length/m	Composition Time/s
PRM algorithm	30	582.1	0.958
	60	602.6	3.269
	90	615.4	10.393
Modified PRM algorithm	30	593.3	0.404
	60	590.6	2.056
	90	578.7	5.196

Taking sampling points $N = 60$ as an example, we analyzed the results of the roadmap construction (Figures 6a and 7a). The sampling points were widely distributed in the PRM algorithm and there were many redundant sampling points. On the other hand, for the roadmap constructed by the modified PRM algorithm (Figure 7a), the location selection of the sampling points had a certain orientation, mainly distributed along the main axis of space, and there were fewer redundant sampling points.

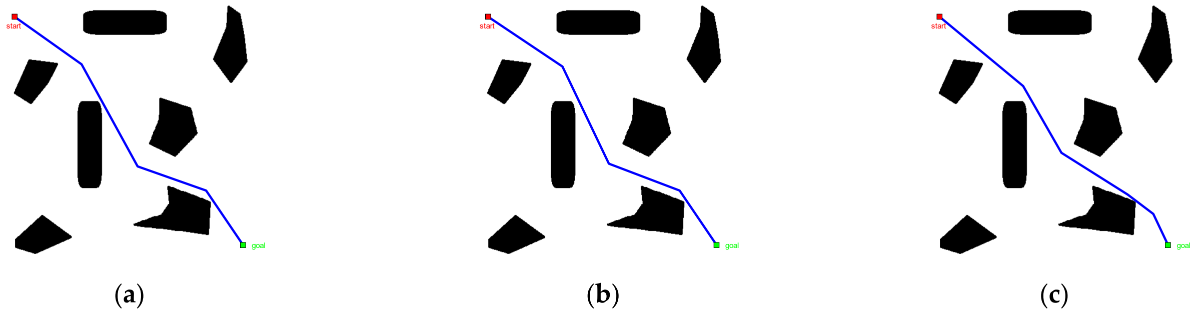
In Figures 6 and 7 and Table 1, it is shown that when the number of sampling points N is 30, the length of the planned path increases by 1.9% and composition time is reduced by 57.8%. When the number of sampling points N is 60, the length of the planned path is reduced by 1.9% and composition time is reduced by 37.1%. When the number of sampling points increase to 90, the length of the planned path is reduced by 5.9% and composition time is reduced 50%. It shows that the changes in path length according to different number N are not consistent. Compared with the PRM algorithm, there is no great advantage in path length for the modified PRM algorithm. However, the modified PRM algorithm showed great advantages in decreasing the construction time of the roadmap; the efficiency of constructing maps was significantly improved.

In Figure 8, keeping all other conditions equal, when the number of fold points of the path increased, path smoothness gradually improved as the number of sampling points increased. The overall trend of the path remains unchanged, indicating that the quality of the path solution solved by the modified PRM algorithm is stable.

To obtain Figure 9, we used the Bessel curve to deal with Figure 8b, the solid blue line indicating the modified PRM algorithm planning path and the black hollow circle logo representing the key road points, used as the Bessel curve control points. The path obtained after the smoothing process (shown by the red line) was more consistent with intelligent vehicle driving road conditions.

4.2. Comparison of Path Planning Efficiency

To verify the path planning efficiency of the modified PRM algorithm, the basic PRM algorithm was used as the comparison algorithm for the case test, where Case A is a square maze and Case B is a narrow channel. The success rate was measured by a ratio of the number of successful path searches to total search number. The results of the case test are shown in Figures 10 and 11 and Table 2.



(**a**) (**b**) (**c**)

Figure 8. Comparison of planning results of modified PRM algorithm: (**a**) $N = 30$, (**b**) $N = 60$, and (**c**) $N = 90$.

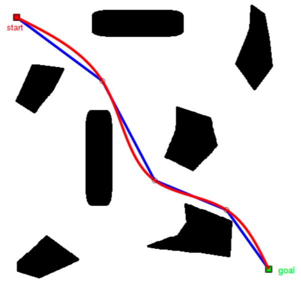
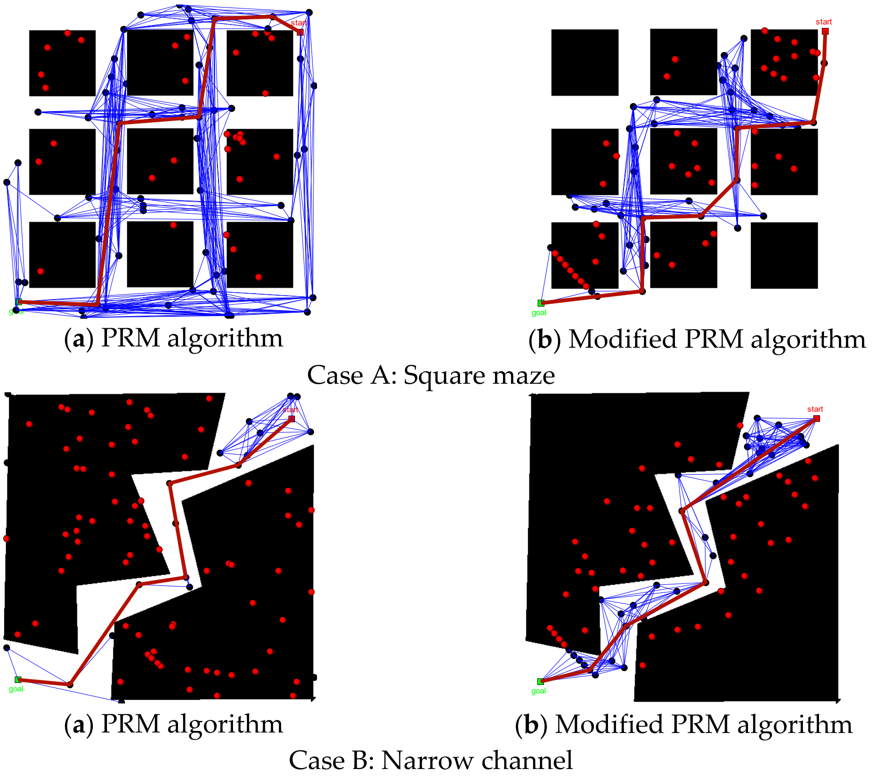


Figure 9. Path smoothing diagram.



(**a**) PRM algorithm (**b**) Modified PRM algorithm

Case A: Square maze

(**a**) PRM algorithm (**b**) Modified PRM algorithm

Case B: Narrow channel

Figure 10. Comparison of algorithm planning results.

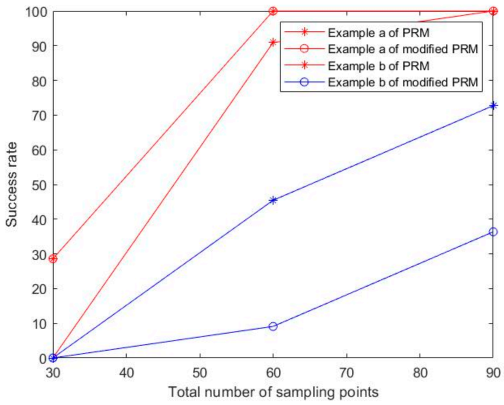


Figure 11. Algorithm success rate comparison.

Table 2. Comparison results of algorithm efficiency.

Algorithm Case	Sampling Points *N*	Basic PRM Algorithm			Modified PRM Algorithm		
		Path Length/m	Running Time/s	Success Rate/%	Path Length/m	Running Time/s	Success Rate/%
A	30	883.14	0.23	28.57	\	\	0
	60	869.63	0.89	100	839	0.62	90.91
	90	861.86	2.18	100	819.91	1.12	100
B	30	\	\	0	\	\	0
	60	812.1	0.28	9.09	735.51	0.29	45.45
	90	734.53	0.43	36.36	729.45	0.73	72.73

Referring to Figure 10, in the experiment of Case A, the number of sampling points falling into the obstacle space was comparable in both algorithms, but the sampling points in the self-use space were widely distributed in the PRM algorithm, which caused redundancy. In the modified PRM algorithm, the sampling points were concentrated on both sides of the main axis of the space, which improved the utilization of sampling points. In the experiment of Case B, most of the sampling points in the PRM algorithm fell into the obstacle space, and there were very few sampling points in the self-use space, which affected the quality of the path solution. In the modified PRM algorithm, the sampling points were distributed along the main axis of the space, and the larger number of sampling points in the self-use space provided the possibility of seeking a better path solution.

In Table 2 and Figure 11, for Case A, the modified PRM algorithm could not successfully plan the path when the number of sampling points was low ($N = 30$). When the number of sampling points increased to 60 ($N = 60$), the differences between the two algorithms in path length, running time, and success rate were not obvious. When the number of sampling points increased to 90 ($N = 90$), the modified PRM algorithm was better than the basic PRM algorithm in path length and running time. For Case B, when the number of sampling points was low ($N = 30$), both algorithms could not successfully plan the path, and as the number of sampling points increased, the modified PRM algorithm had a higher success rate in path planning and the quality of the path solution was more reliable.

4.3. ROS Simulation Test

In order to further verify the implementability of the modified PRM algorithm, simulation tests were designed, based on the ROS experimental platform. The composition of the ROS trolley is shown in Figure 12.

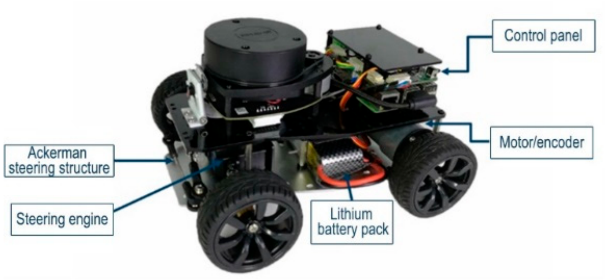


Figure 12. ROS car composition.

We mainly addressed the path planning problem of smart vehicles in a two-dimensional environment, using the function package provided by the ROS experimental platform to implement the LIDAR map building function. The test site is shown in Figure 13, and the SLAM map building effect is shown in Figure 14. Based on this environmental map, we defined the localization result of ROS itself as the starting point and specified the target point. The modified PRM algorithm was executed and the path planning results are shown in Figure 15.

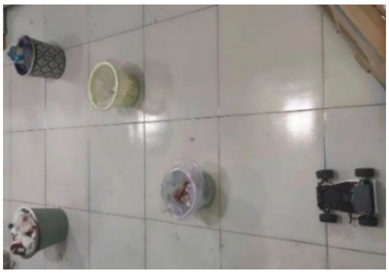


Figure 13. Field map.

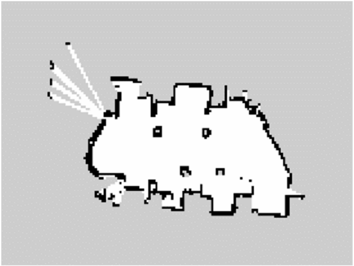
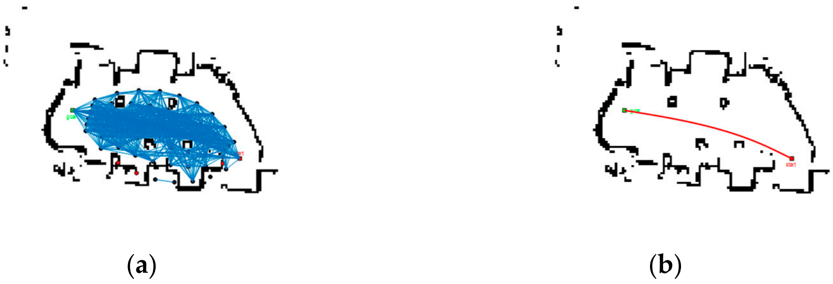


Figure 14. SLAM map.



(**a**) (**b**)

Figure 15. Path planning: (**a**) wayfinding map and (**b**) planning path.

From the simulation results, a road map was established in the SLAM map by the modified PRM algorithm. Meanwhile, the modified PRM algorithm planned a path successfully connecting the starting and target point, verifying the feasibility of the modified PRM algorithm.

5. Conclusions

In order to improve the overall quality of the PRM algorithm in path planning, a pseudo-random sampling method based on uniform sampling was designed to optimize the quality of sampling point generation. Random increments were introduced to adjust the fluctuation range of sampling points to effectively avoid the obstacle space. Due to the disadvantage of a low rate of roadmap construction, a two-way incremental collision detection strategy was used to set the connection threshold between road points to reduce the number of collision detection calls. Finally, the correctness of the modified PRM algorithm was verified and analyzed using MATLAB and ROS test platforms. The test results showed that the modified PRM algorithm has obvious advantages in enhancing the stability of the roadmap, shortening the length of the planned path, and improving the search rate of the algorithm. However, the majority of current algorithms, including the modified PRM algorithm, are model-driven, and face many limitations. These algorithms need to be further researched. Data-driven and cloud-network fusion technologies could be added to these algorithms to achieve better path planning and obstacle avoidance in smart vehicles.

Author Contributions: Conceptualization, S.B. and J.Y.; methodology, Q.L.; software, Q.L. and S.B.; validation, Q.L. and. S.B.; formal analysis, Q.L.; investigation, Y.X.; resources, Q.L.; data curation, Q.L.; writing—original draft preparation, Q.L. and. S.B.; writing—review and editing, Q.L.; visualization, Q.L.; supervision, J.Y.; project administration, J.Y.; funding acquisition, Q.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Graduate Research Innovation Program Project of Jiangsu Province, China, grant number "KYCX22_1059".

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The study did not report any data.

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

References

1. Shi, Y.J.; Han, Q.M.; Shen, W.M.; Wang, X. A Multi-Layer Collaboration Framework for Industrial Parks with 5G Vehicle-to-Everything Networksy. *Engineering* **2021**, *7*, 818–831. [CrossRef]
2. Zheng, Z.G.; Bashir, A.K. Graph-Enabled Intelligent Vehicular Network Data Processing. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 4726–4735. [CrossRef]
3. Sun, N.; Shi, H.Z.; Han, G.J.; Wang, B.; Shu, L. Dynamic Path Planning Algorithms With Load Balancing Based on Data Prediction for Smart Transportation Systems. *IEEE Access* **2020**, *8*, 15907–15922. [CrossRef]
4. Zhu, M.; Liu, X.Y.; Wang, X.D. An Online Ride-Sharing Path-Planning Strategy for Public Vehicle Systems. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 616–627. [CrossRef]
5. Zhu, Y.L.; Chu, K.M.; Chen, X.; Wang, X.; Su, H. Research and application of a multi-degree-of-freedom soft actuator. *Sens. Actuators A Phys.* **2022**, *338*, 113492. [CrossRef]
6. Tian, Y.; Yao, Q.Q.; Hang, P.; Wang, S. Adaptive Coordinated Path Tracking Control Strategy for Autonomous Vehicles with Direct Yaw Moment Control. *Chin. J. Mech. Eng.* **2022**, *35*, 1. [CrossRef]
7. Ayawli, B.K.; Chellali, R.; Appiah, A.Y.; Kyeremeh, F. An Overview of Nature-Inspired, Conventional, and Hybrid Methods of Autonomous Vehicle Path Planning. *J. Adv. Transp.* **2018**, *2018*, 8269698. [CrossRef]
8. Nazarahari, M.; Khanmirza, E.; Doostie, S. Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm. *Expert Syst. Appl.* **2019**, *115*, 106–120. [CrossRef]
9. Siddique, N.; Adeli, H. Nature Inspired Computing: An Overview and Some Future Directions. *Cogn. Comput.* **2015**, *7*, 706–714. [CrossRef] [PubMed]
10. Holland, J.H. Building blocks, cohort genetic algorithms, and hyperplane-defined functions. *Evol. Comput.* **2000**, *8*, 373–391. [CrossRef]
11. Tang, M.; Liu, Y.; Ding, F.; Wang, Z. Solution to Solid Wood Board Cutting Stock Problem. *Appl. Sci.* **2021**, *11*, 7790. [CrossRef]
12. Hoffman, M.R.; Mielens, J.D.; Omari, T.I.; Rommel, N.; Jiang, J.J.; McCulloch, T.M. Artificial neural network classification of pharyngeal high-resolution manometry with impedance data. *Laryngoscope* **2013**, *123*, 713–720. [CrossRef] [PubMed]
13. Yang, J.; Wang, Y.D.; Chen, Y.; Yu, J. Detection of Weeds Growing in Alfalfa Using Convolutional Neural Networks. *Agronomy* **2022**, *12*, 1459. [CrossRef]

14. Kirkpatrick, S.; Gelatt, C.D., Jr.; Vecchi, M.P. Optimization by Simulated Annealing. *Science* **1983**, *220*, 671–680. [CrossRef]
15. Dorigo, M.; Di Caro, G.; Gambardella, L.M. Ant algorithms for discrete optimization. *Artif. Life* **1999**, *5*, 137–172. [CrossRef]
16. Hu, X.H.; Shi, Y.H.; Eberhart, R. Recent Advances in Particle Swarm. In Proceedings of the 2004 Congress on Evolutionary Computation, Portland, OR, USA, 19–23 June 2004; IEEE: Piscataway, NJ, USA, 2004; pp. 90–97.
17. Karaboga, D.; Basturk, B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *J. Glob. Optim.* **2007**, *39*, 459–471. [CrossRef]
18. Xu, X.Y.; Xie, J.; Xie, K.M. Path Planning and Obstacle-Avoidance for Soccer Robot Based on Artificial Potential Field and Genetic Algorithm. In Proceedings of the 2006 6th World Congress on Intelligent Control and Automation, Dalian, China, 21–23 June 2006; IEEE: Piscataway, NJ, USA, 2006; pp. 3494–3498.
19. Girija, S.; Joshi, A. Fast Hybrid PSO-APF Algorithm for Path Planning in Obstacle Rich Environment. *IFAC Pap.* **2019**, *52*, 25–30. [CrossRef]
20. Dai, L.; Wu, C.; Qi, J.T.; Han, J. Fuzzy Adaptive Kalman Filter Algorithm for RUAV's Integrated Navigation System. In Proceedings of the 2012 24th Chinese Control and Decision Conference (CCDC), Taiyuan, China, 23–25 May 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 2865–2869.
21. Bahari, M.H.; Karsaz, A.; Pariz, N. High maneuvering target tracking using a novel hybrid kalman filter-fuzzy logic architecture. *Int. J. Innov. Comput. Inf. Control* **2011**, *7*, 501–510.
22. Simon, P. Principles of Robot Motion: Theory, Algorithms and Implementations. *Knowl. Eng. Rev.* **2007**, *22*, 209–211.
23. Lydia, E.K.; Petr, S.; Jean-Claude, L.; Overmars, M.H. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Autom.* **1996**, *12*, 566–580.
24. Oussama, K. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *Int. J. Robot. Res.* **1986**, *5*, 396–404.
25. Hwang, Y.K.; Ahuja, N. A potential field approach to path planning. *IEEE Trans. Robot. Autom.* **1992**, *8*, 23–32. [CrossRef]
26. Camacho, O.; Smith, C.A. Sliding mode control: An approach to regulate nonlinear chemical processes. *ISA Trans* **2000**, *39*, 205–218. [CrossRef]
27. Andrew, R.L.; Keith, P. Automated conformational analysis: Directed conformational search using the A* algorithm. *J. Comput. Chem.* **1990**, *11*, 1193–1205.
28. Carsten, J.; Ferguson, D.; Stentz, A. 3D Field D*: Improved Path Planning and Replanning in Three Dimensions. In Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 9–15 October 2006; IEEE: Piscataway, NJ, USA, 2006; p. 3381.
29. Zhou, J.H.; Lin, H.Y. A self-localization and path planning technique for mobile robot navigation. In Proceedings of the 2011 9th World Congress on Intelligent Control and Automation, Taipei, Taiwan, 21–25 June 2011.
30. Cadena, C.; Carlone, L.; Carrillo, H.; Latif, Y.; Scaramuzza, D.; Neira, J.; Reid, I.; Leonard, J.J. Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *IEEE Trans. Robot.* **2016**, *32*, 1309–1332. [CrossRef]
31. Cheng, Q.; Zhang, W.; Liu, H.; Zhang, Y.; Hao, L. Research on the Path Planning Algorithm of a Manipulator Based on GMM/GMR-MPRM. *Appl. Sci.* **2021**, *11*, 7599. [CrossRef]
32. Kantaros, Y.; Zavlanos, M.M. STyLuS*: A Temporal Logic Optimal Control Synthesis Algorithm for Large-Scale Multi-Robot Systems. *Int. J. Robot. Res.* **2020**, *39*, 812–836. [CrossRef]
33. Vasile, C.I.; Belta, C. Sampling-based temporal logic path planning. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013.
34. Dobson, A.; Bekris, K.E. Improving Sparse Roadmap Spanners. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 4106–4111.
35. Amato, N.M.; Dale, L.K. Probabilistic roadmap methods are embarrassingly parallel. In Proceedings of the 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C), Detroit, MI, USA, 10–15 May 1999.
36. Bekris, K.E.; Chen, B.Y.; Ladd, A.M.; Plaku, E.; Kavraki, L.E. Multiple query probabilistic roadmap planning using single query planning primitives. In Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No. 03CH37453), Las Vegas, NV, USA, 27–31 October 2003.
37. Kurniawati, H.; Hsu, D. *Workspace-Based Connectivity Oracle: An Adaptive Sampling Strategy for PRM Planning*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 35–51.
38. Esposito, J.M.; Wright, J.N. Matrix completion as a post-processing technique for probabilistic roadmaps. *Int. J. Robot. Res.* **2019**, *38*, 388–400. [CrossRef]
39. Gao, J.; Ye, W.; Guo, J.; Li, Z. Deep Reinforcement Learning for Indoor Mobile Robot Path Planning. *Sensors* **2020**, *20*, 5493. [CrossRef]
40. Chen, G.; Luo, N.; Liu, D.; Zhao, Z.; Liang, C. Path planning for manipulators based on an improved probabilistic roadmap method. *Robot. Comput.-Integr. Manuf.* **2021**, *72*, 102196. [CrossRef]
41. Ravankar, A.A.; Ravankar, A.; Emaru, T.; Kobayashi, Y. HPPRM: Hybrid Potential Based Probabilistic Roadmap Algorithm for Improved Dynamic Path Planning of Mobile Robots. *IEEE Access* **2020**, *8*, 221743–221766. [CrossRef]
42. Xu, Z.F.; Deng, D.; Shimada, K. Autonomous UAV Exploration of Dynamic Environments Via Incremental Sampling and Probabilistic Roadmap. *IEEE Robot. Autom. Lett.* **2021**, *6*, 2729–2736. [CrossRef]