applied sciences

MDPI

Article

Improved Analytic Expansions in Hybrid A-Star Path Planning for Non-Holonomic Robots

Chien Van Dang [1], **Heungju Ahn** [2], **Doo Seok Lee** [2] **and Sang C. Lee** [1,*]

[1] Division of Intelligent Robot, Convergence Research Institute, DGIST, Daegu 42988, Korea; dvchienbkvn@gmail.com

[2] College of Transdisciplinary Studies, DGIST, Daegu 42988, Korea; heungju@dgist.ac.kr (H.A.); dslee@dgist.ac.kr (D.S.L.)

* Correspondence: sclee@dgist.ac.kr

Abstract: In this study, we concisely investigate two phases in the hybrid A-star algorithm for non-holonomic robots: the forward search phase and analytic expansion phase. The forward search phase considers the kinematics of the robot model in order to plan continuous motion of the robot in discrete grid maps. Reeds-Shepp (RS) curve in the analytic expansion phase augments the accuracy and the speed of the algorithm. However, RS curves are often produced close to obstacles, especially at corners. Consequently, the robot may collide with obstacles through the process of movement at these corners because of the measurement errors or errors of motor controllers. Therefore, we propose an improved RS method to eventually improve the hybrid A-star algorithm's performance in terms of safety for robots to move in indoor environments. The advantage of the proposed method is that the non-holonomic robot has multiple options of curvature or turning radius to move safer on pathways. To select a safer route among multiple routes to a goal configuration, we introduce a cost function to evaluate the cost of risk of robot collision, and the cost of movement of the robot along the route. In addition, generated paths by the forward search phase always consist of unnecessary turning points. To overcome this issue, we present a fine-tuning of motion primitive in the forward search phase to make the route smoother without using complex path smoothing techniques. In the end, the effectiveness of the improved method is verified via its performance in simulations using benchmark maps where cost of risk of collision and number of turning points are reduced by up to around 20%.

Keywords: Reeds-Shepp curves; hybrid A-star; non-holonomic mobile robot; indoor robot applications

check for updates

Citation: Dang, C.V.; 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. https://doi.org/10.3390/app12125999

Academic Editors: Euntai Kim, Zong Woo Geem, Seokwon Yeom, Young-Jae Ryoo and Myung-Geun Chun

Received: 29 April 2022
Accepted: 11 June 2022
Published: 13 June 2022

1. Introduction

Since the emergence of artificial intelligence, path planning has become an attractive topic investigated in many studies. Popular path planning methods such as probabilistic road map (PRM), rapidly-exploring random tree (RRT), and space skeletonization have their advantages because of the countless number of scenarios in occupancy grid maps [1]. Moreover, diverse path planning algorithms [2,3] have emerged to tackle constraints of motions of robots or autonomous vehicles. Path planning algorithms include Dijkstra algorithm [4,5], A-star algorithm [6,7], Genetic algorithm [8,9], Artificial Neural Network (ANN) [10] and a combination of several algorithms [11,12] to name a few. Resulted waypoints of a path planner are connected to form a trajectory from a start configuration to a goal configuration. The trajectory mainly contains many sharp or angular turns that make a mobile robot move awkwardly. Many experts and scholars have studied path smoothing techniques that produce a smooth path and guarantee a kinematically feasible obstacle-free path to deal with this problem [13]. Velocity, acceleration, and jerk have also been optimized to acquire smoothing trajectories for autonomous vehicles [14–16]. While the robot moves on a smooth trajectory, obstacle avoidance has also been considered [17,18].

However, path smoothing methods can not make informed decisions regarding other paths during the search algorithm.

Stanford University researchers have developed a variant of the A-star algorithm, named hybrid A-star, for application to a real car robot in an outdoor racing competition [19]. The hybrid A-star algorithm is a hybrid of discrete state and continuous state. This algorithm consists of two phases: the forward search phase, which takes into account the kinematics of the robot model in order to plan continuous motions of the robot in discrete grid maps, and the analytic expansion phase, which uses the RS curve to augment the accuracy and the speed of the algorithm.

Planning a continuous state in the discrete domain of the hybrid A-star search algorithm has been an inspiration for numerous researchers to investigate and strive to apply it to real applications. Authors in [20] extended the search algorithm to accomplish several challenges, such as planning to a waypoint lying outside the vehicle's local map in outdoor applications. In [21], the authors proposed a method that is a combination of a visibility diagram and the hybrid A-star algorithm for valet parking applications. Firstly, the visibility diagram algorithm was used to find the shortest visible waypoints from a start position to a goal position. Then, the hybrid A-star was implemented to follow these waypoints with non-holonomic constraints of vehicles. The method proved faster than the standard hybrid A-star. However, RS curves' drawbacks when the robot approaches the destination were ignored in these two works. In another work [22], the authors strove to optimize paths generated by the hybrid A-star search. A concept of the artificial potential field was applied to reduce unnecessary steering actions and smooth the paths. However, the proposed method was applied in the hybrid A-star path as a post-smoothing technique. Since the RS curves were re-used from the open motion planning library, the problem in implementing the RS curves has remained.

In this study, we investigate the hybrid A-star algorithm for non-holonomic robots in indoor environments. Through this investigation, issues that occurred in each phase of the algorithm are discovered. First, RS curves in the analytic expansion are often generated closely to obstacles, especially at corners. To overcome this issue, we generate multiple RS curves with different curvature values and propose a cost function to evaluate each curve to select the best one on which the robot moves safely in the presence of obstacles. Second, routes developed by the forward search consist of multiple unnecessary turns. To deal with this issue, without using complex smoothing techniques, we suggest effective fine-tuning to reduce unnecessary turns. By resolving these issues, we eventually improve the algorithm's performance in terms of safety for use in indoor robot applications. Experimental demonstrations are conducted using a non-holonomic robot to highlight the effectiveness of the proposed method in simulations.

The rest of this study is organized as follows. Section 2 briefly describes the hybrid A-star path planning algorithm. Moreover, a problem in the planned path is presented as we implement the bare-bone algorithm in indoor environments. Section 3 presents our proposed solutions to overcome the problem, including a cost function for selecting a safer way to achieve a goal configuration. The proposed solution is verified through intensive tests using benchmark maps in simulations in Section 4. Section 5 presents conclusions and our future work.

2. Statement of the Problem

2.1. Hybrid-State A-Star Algorithm

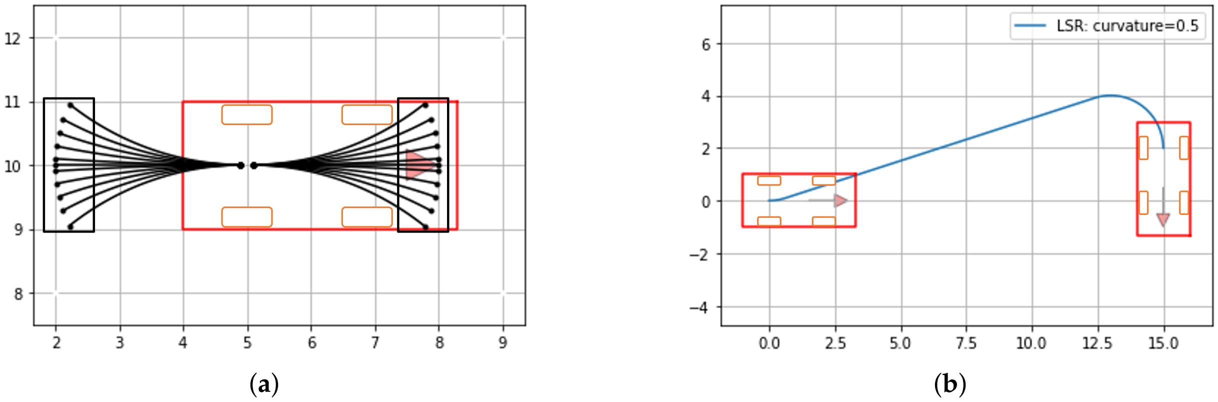
The hybrid-state A-star (or hybrid A-star in short) is known as one of the most efficient path planners for non-holonomic autonomous vehicles. This planner was first implemented in outdoor environments by the Stanford Racing Teams robot, Junior, in the 2007 DARPA urban challenge [19].

The hybrid A-star algorithm includes two phases (the forward search and analytic expansions) to produce an effective path [23,24]. In the forward search, as the conventional A-star algorithm, the hybrid A-star algorithm is guided by a cost-to-go $g(n)$ which is a cost from the start node to the current node (n), and a heuristic function $h_1(n)$. The heuristic function $h_1(n)$ uses the obstacle map to compute the cost of the distance from the current node to the goal in the discrete state. Moreover, like an advanced version of the normal A-star, the hybrid A-star algorithm is also guided by another heuristic function $h_2(n)$ that ignores obstacles but considers the kinematics of a non-holonomic robot (e.g., autonomous vehicles). Further details can be found in [24–26]. The kinematics of the robot model is used to predict motions that depend on the speed, direction, and steering angle of the robot in the continuous state. This way helps the algorithm select the right successor in continuous coordinate, which a non-holonomic robot is able to follow, as shown in Figure 1a.

These continuous coordinate successors have converted approximately to discrete coordinate nodes to compute the total cost of a movement. Equation (1) represents an objective function of the planner at each node n. The successor with the lowest objective value will be chosen until it reaches the goal node.

$$F(n) = g(n) + h_1(n) + h_2(n) \tag{1}$$

However, reaching the exact continuous coordinate goal node is hard due to the dependence of the resolution of the grid map and the smallest motion that the robot can take.



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

Figure 1. Hybrid state A-star method. (**a**) Successor selection in continuous domain. (**b**) Reeds-Shepp curve.

Analytic expansions are the second phase of the hybrid A-star algorithm. This phase plays a role in guaranteeing that the algorithm reaches the exact continuous coordinate of the goal state. In this phase, the RS curves have been utilized since the RS curves algorithm was a well-known algorithm that applies to non-holonomic vehicles that can move forward and backward. An RS curve is calculated to produce the shortest path from a start point to a goal point. Figure 1b depicts a combination of Left, Straight, and Right curves (or LSR curve in short) between two configurations: $[0, 0, 0]$, $[15, 2, -\pi/2]$, where the first two values are position, the third value is an orientation of the robot. The red arrow represents the robot's direction. The curvature value (0.5 in this case) allows the robot to make the most turning angle, or minimum turning radius. The inventors of the hybrid A-star have claimed that the analytic expansion method leads to significant benefits in terms of accuracy and search speed.

Additionally, Voronoi field was proposed to help the vehicle move in narrow opening by adding a path-cost function v as shown in Equation (2) below:

$$v(x,y) = \frac{\alpha}{\alpha + d_0(x,y)} \frac{d_v(x,y)}{d_0(x,y) + d_v(x,y)} \left(\frac{d_0 - d_0^{max}}{d_0^{max}} \right)^2 \tag{2}$$

where, α is a constant falloff rate $(\alpha > 0)$, d_0^{max} is a constant maximum effective range of the field $(d_0^{max} > 0)$, d_0 is a distance to the nearest obstacle $(d_0 < d_0^{max})$, and d_v is a distance to the nearest edge of the generalized Voronoi diagram. The effectiveness of this function is that the cost is higher in the area near obstacles, and decreases to zero at points that are equidistant to obstacles. As a result, the hybrid A-star path will tend to be in the center of the road.

2.2. Our Path Planning Problem

By implementing the bare-bone algorithm on simulations, we have run into a problem that we should deal with as below.

Figure 2 illustrates two simulations of the hybrid A-star in self-created indoor environments of size 50 × 30 m, and a resolution of 1 m. The non-holonomic robot is annotated as a red rectangle. The robot's state is captured along the resultant path, where the blue line represents the results of the forward search, and the green line represents the results of the RS path.

The blue lines lie in the middle of the corridor. However, these blue lines contain non-smooth segments, such as segments indicated by black ellipses. Most importantly, the green lines are often close to walls, especially at corners indicated by black circles. The robot may collide with obstacles through the process of movement at these corners because of the measurement errors or errors of motor controllers. To overcome this problem, in the next section, we will improve the analytic expansions of the algorithm by using an interesting property of RS curves.

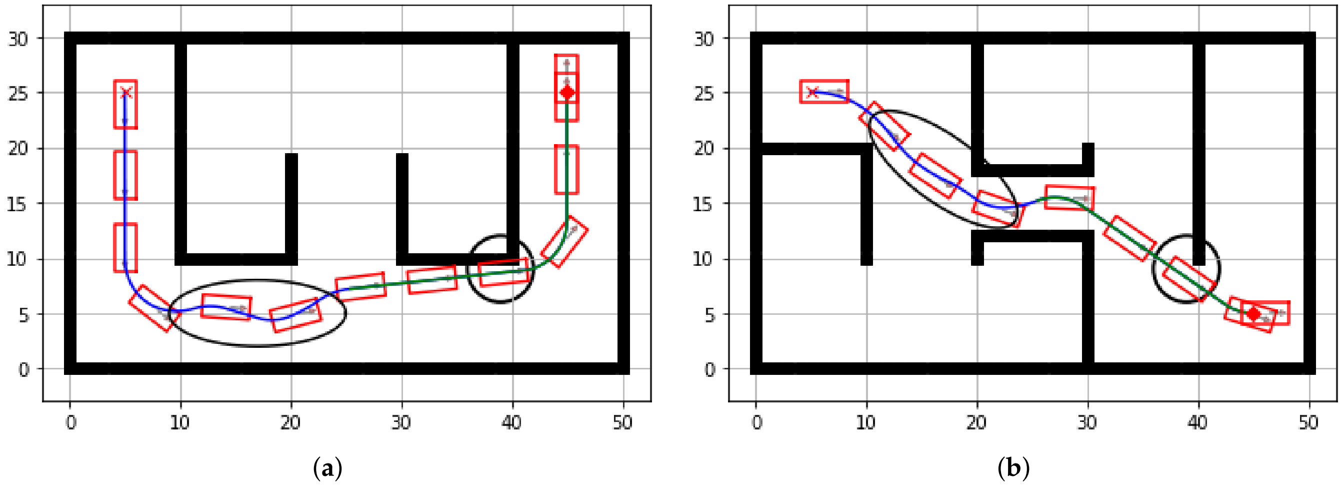


Figure 2. Original Hybrid A-star path planning in simulations. (**a**) Grid map A. (**b**) Grid map B.

3. An Improved Method

The primary purpose of using the RS method in the analytic expansions is to increase the accuracy and speed up the search because the RS method is a well-known method for a non-holonomic vehicle to move from a configuration $[x_0, y_0, \theta_0]$ to another configuration $[x_g, y_g, \theta_g]$ in the shortest path. More details about RS curves can be found in [27].

Figure 3 depicts three different LSR routes using the RS method from an initial state $[x_0, y_0, \theta_0] = [0, 0, 0]$ to a goal state $[x_g, y_g, \theta_g] = [-3, 5, \pi/2]$ according to three different values of curvature. The green route has the biggest curvature, but its length is the shortest of the three. Whereas the blue route corresponds to the smallest curvature, its length is the longest of the three. The turning radius is reversely proportional to curvature. The turning radius in the blue route is much larger than the radius in the orange and green routes. Therefore, it is possible to achieve a safer route for a car-like robot by adjusting curvature value.

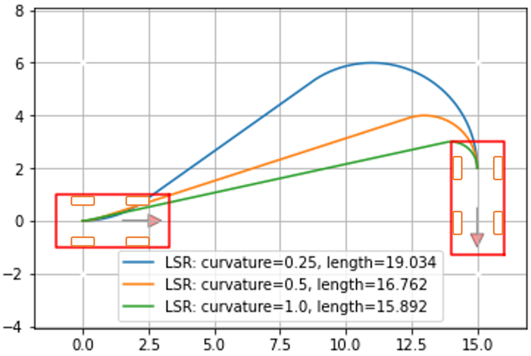


Figure 3. Reed-Shepp paths with different curvature values.

Among different paths generated from different curvatures, a better path is selected in terms of safety from a start state to a goal state. In order to evaluate each path, we propose the following objective function:

$$G = \sigma_1 \cdot v + \sigma_2 \cdot m \tag{3}$$

where, σ_1 and σ_2 are two coefficients normalizing the two related cost functions. The function v represents the cost of the Voronoi field, which is used in the post-processing Equation (2). The farther the robot is from obstacles, the smaller the value of the cost function. Utilizing this cost function, we see the robot prefers relatively equidistant moves to surrounding obstacles.

Meanwhile, the function m is the cost of movement along the RS path which includes straight lines and curves as mentioned above. For a candidate RS path (p), costs of each straight line or curve are computed including its length (l_i), steering angle (s_i), and steer switching (c_i) in functions of (l_p), (s_p), and (c_p) respectively. With three constant weights w_1, w_2, w_3, the cost function m has the following form:

$$\begin{aligned} m &= l_p + s_p + c_p \\ l_p &= w_1 \cdot \sum l_i \\ s_p &= w_2 \cdot \sum s_i \\ c_p &= w_3 \cdot \sum c_i \end{aligned} \tag{4}$$

Because of the safety criteria, a car-like robot or a car does not always make the sharpest turn, or a maximum curvature, but it might choose a less sharp curvature to complete a turn. By combining these two sub-functions (v, m), with appropriate weights σ_1, σ_2, we trade off between selecting safer or faster path. If the coefficient σ_1 is larger than σ_2, then the robot prefers the safe path to the short one.

The improved method in the analytic expansion is mainly divided into three steps:

1. First, RS curves are generated based on different curvature values from the input state to the goal state. Collision to obstacles is checked along the curves. Hence the curve which does not collide with obstacles is chosen.
2. Second, the cost of each curve is calculated according to the Equation (3).
3. Third, the curve which has the lowest cost is selected as the best curve. The robot is expected to travel more safely along this curve.

If such an RS curve exists, the method will assign it as the path to the goal and the search ends. If such a curve does not exist, the algorithm will continue with the forward search as in the original method:

1. Computes robot continuous coordinates based on a non-holonomic model,
2. Converts these coordinates to corresponding discrete coordinates in the grid map,
3. Calculates objective costs according to the Equation (1),

4. Selects the best successor for the next search loop.

The flowchart of the overall method is shown in Figure 4. To verify the effectiveness of the proposed solution, we conduct simulations in the next section.

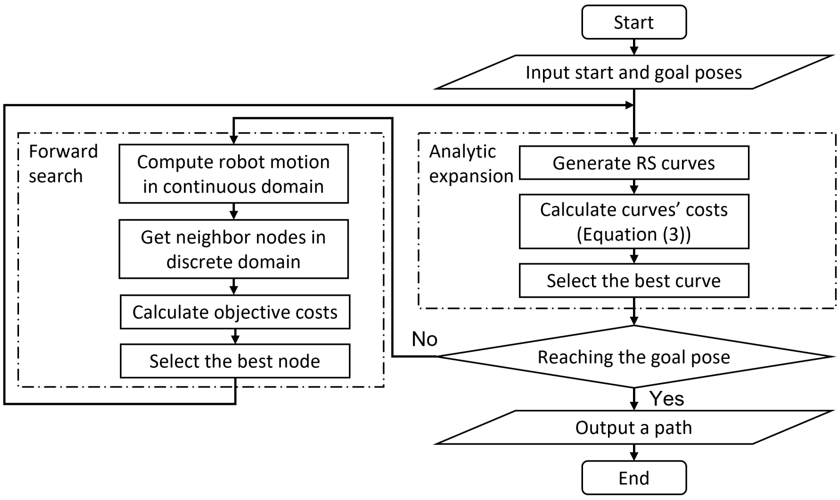
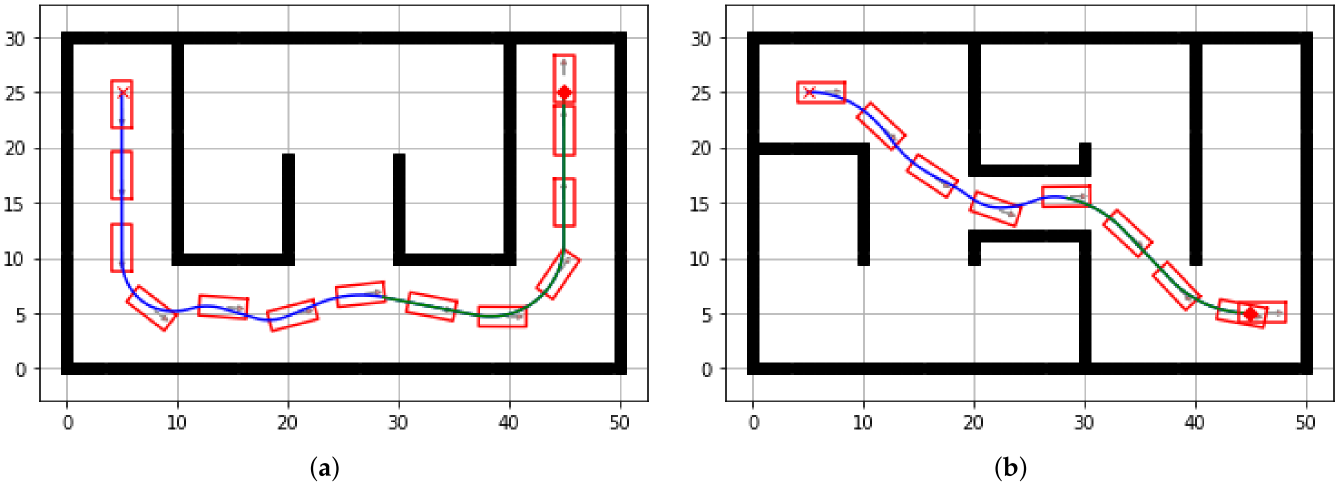

Start
Input start and goal poses
Forward search
Compute robot motion in continuous domain
Get neighbor nodes in discrete domain
Calculate objective costs
Select the best node
Analytic expansion
Generate RS curves
Calculate curves' costs (Equation (3))
Select the best curve
No
Reaching the goal pose
Yes
Output a path
End

Figure 4. Flowchart of the implemented method.

4. Experimental Simulation

4.1. Simulations

We verify the effectiveness of the proposed solution by simulation on the Spyder IDE platform. The algorithm is implemented in Python 3 programming language. The paths generated by the algorithm are visualized by the matplotlib library. The input map is a binary occupancy grid. The robot model is configured with the size of 4.3 × 2.0 m, wheel base of 3.0 m, and maximum steer of 0.6 rad. We vary the robot's curvature in the range of the maximum steering angle with a resolution of 0.05. Figure 5 shows two paths generated by the improved algorithm using the same initial and goal state configurations as in the previous simulations.



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

Figure 5. Improved Hybrid A-star path planning in indoor grid maps. (**a**) Grid map A. (**b**) Grid map B.

The comparison of the two simulated results depicted in Figures 2 and 5 reveals no change in the results of the forward search, represented by the blue line, but a significant improvement in the results of the analytic expansions, represented by the green line. In the original method, the RS paths are close to walls when the robot is trying to make a turn at sharp corners. In contrast, the improved method chooses RS paths that are more in the middle of the road to avoid sharp-angled corners thanks to the cost function.

Table 1 summarizes the comparison of the original and improved RS paths in terms of curvature, risk of collision (cost), path length (length) and execution time (time). Higher values indicate higher risks of collision, longer paths and slower finding the goal. Generated by smaller curvatures, the proposed method improves the original method in terms of the risk of colliding with obstacles (8.18 versus 15.27, and 5.52 versus 5.85). Besides, the length of the improved path is shorter in the grid map A, but longer in the grid map B, although the difference is not significant (34.05 versus 34.07 and 20.25 versus 19.83, respectively). In both cases, the proposed method requires more execution time in the analytic expansion phase because more candidate RS paths with corresponding curvatures must be computed. The difference in execution time increases as curvature resolution increases.

Table 1. Comparison of original and improved methods.

Maps	**Original RS Path**				**Improved RS Path**			
	Curvature	**Cost**	**Length (m)**	**Time (s)**	**Curvature**	**Cost**	**Length**	**Time**
Map A	0.23	15.27	34.07	0.022	0.15	8.18	34.05	0.1
Map B	0.23	5.85	19.83	0.015	0.1	5.52	20.25	0.139

4.2. Fine Tunings

Observing Figure 5, the improved hybrid A-star paths in the forward search phase are still unrealistic since they deviate substantially from the true shortest paths, or have many turning points. In Figure 6, we illustrate the smallest continuous motions in discrete domain with two different motion primitives, or arcs, defined as the smallest entities of the path [20]. Each arrow from its tail to head depicts the robot's movement from a current pose to a possible successor's pose. Due to different steering angles, the headings of the successors are different in the same motion primitive. Moreover, the headings of the successors also differ by the influence of motion primitives. Compared to Figure 6a with the same steering angle, the headings of successors in Figure 6b are smaller.

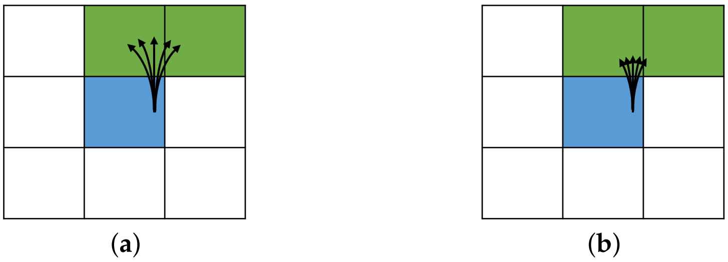


Figure 6. Smallest continuous motions in discrete domain. (**a**) Long motion primitive. (**b**) Short motion primitive.

After tuning the motion primitives without any exhaustive, we found a value of 1.5 for motion primitive to give good results. The resulted routes for grid map A and map B are shown in Figure 7a,b. The number of turning points in these figures is dropped from 5 to 2 points and from 6 to 5 points compared to those in Figure 5a,b, respectively.

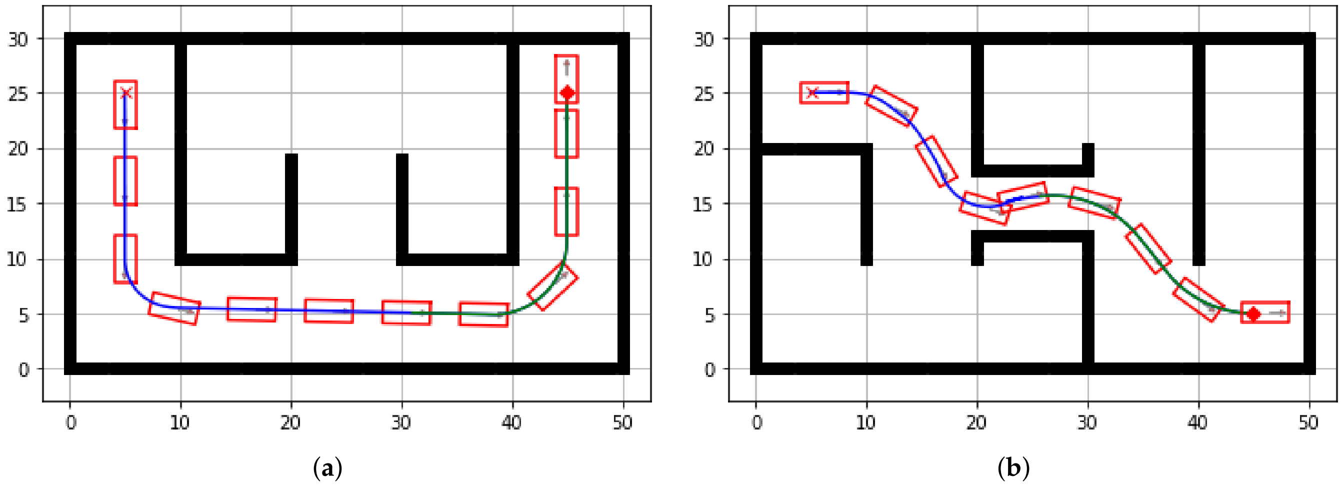


Figure 7. Tuning the improved hybrid A-star path planning. (**a**) Grid map A. (**b**) Gird map B.

4.3. Experiments in Benchmark Maps

In this section, the workspace is standard 2-D grid benchmark maps in common sources [28,29]. We use three different benchmark maps, *map12*, *den520d* and *ost003d*, with the size of 433 × 541 pixel, 1000 × 1000 pixel, and 1000 × 1000 pixel, respectively. These three benchmark maps are re-scaled to maximum 128 pixels per dimension with a resolution of 1 m per pixel. Figures 8a, 9a and 10a show full routes obtained by the improved method in the three benchmark maps. To see clearly the improved paths, we zoom in parts of the forward search phase (blue lines) before tunings and parts of the improved RS paths (green lines) as shown in Figures 8c,e, 9c,e and 10c,e. The corresponding sections of the forward search phase after tunings and the original RS paths are arranged in Figures 8b,d, 9b,d and 10b,d to compare between them.

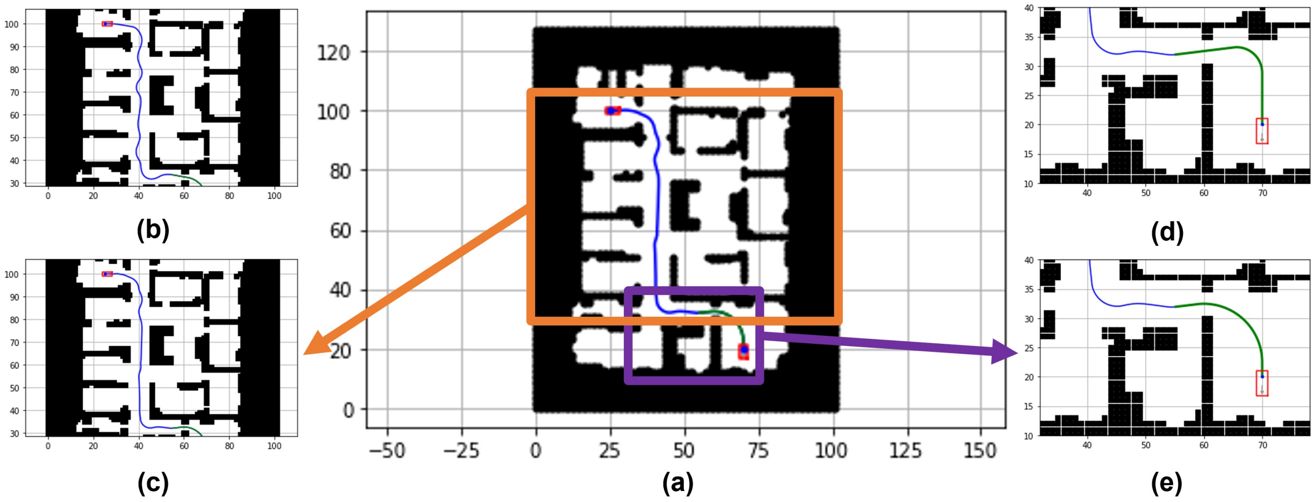


Figure 8. Improved hybrid A-star in a benchmark map *map12*. (**a**) Improved hybrid A-star. (**b**) Before tunings. (**c**) After tunings. (**d**) Original RS path. (**e**) Improved RS path.

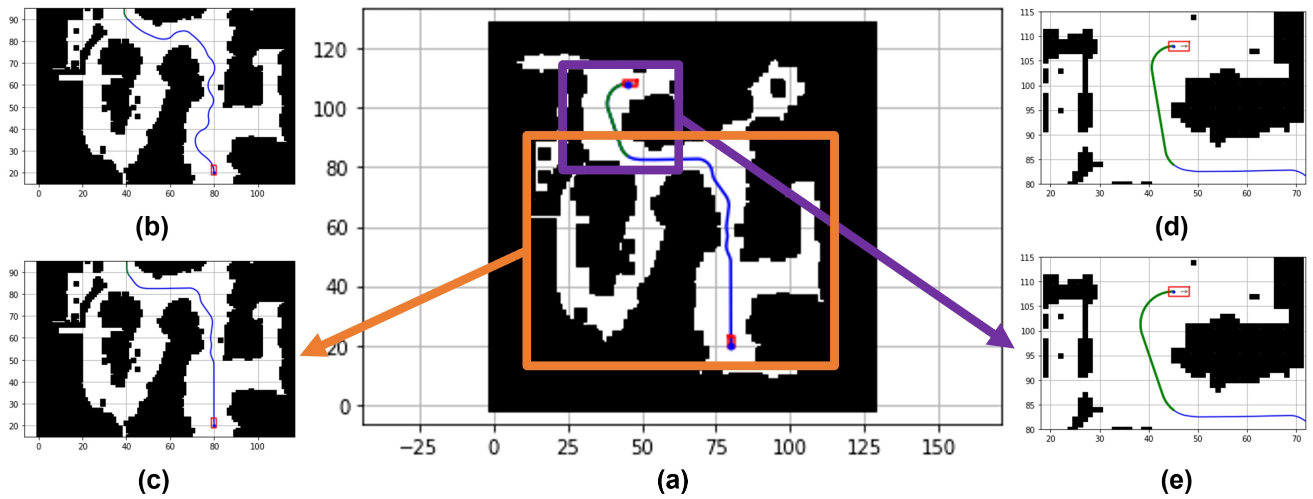


Figure 9. Improved hybrid A-star in a benchmark map *den520d*. (**a**) Improved hybrid A-star. (**b**) Before tunings. (**c**) After tunings. (**d**) Original RS path. (**e**) Improved RS path.

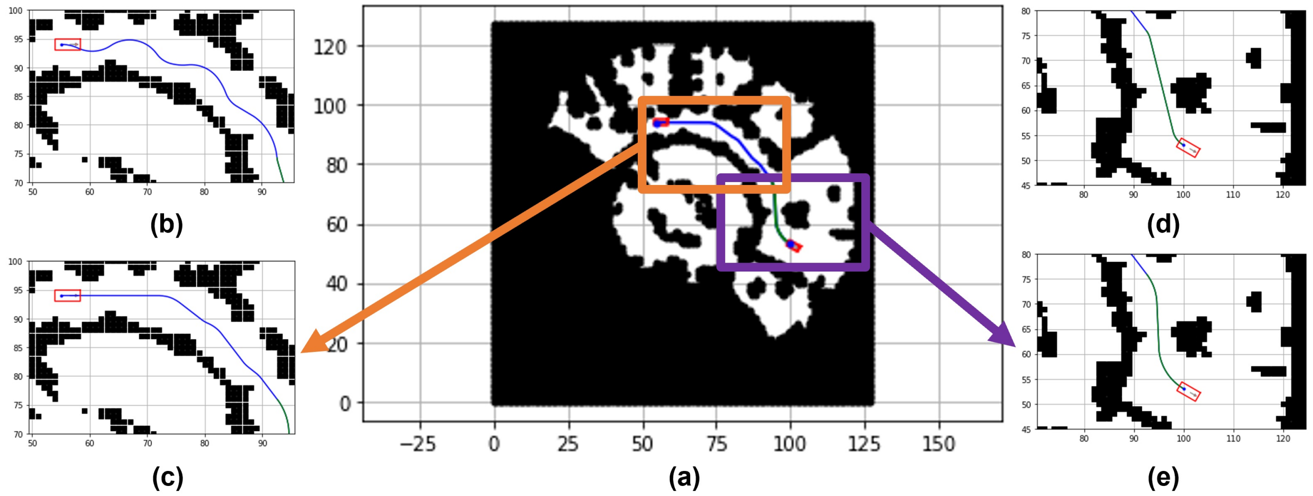


Figure 10. Improved hybrid A-star in a benchmark map *ost003d*. (**a**) Improved hybrid A-star. (**b**) Before tunings. (**c**) After tunings. (**d**) Original RS path. (**e**) Improved RS path.

As it can be seen from the data in Table 2, a smaller curvature value is chosen in both cases in the improved RS path. Consequently, the cost of risk of the improved RS path is reduced by 24% in *map12* and 10% in map *den520d* compared to the original RS path in the hybrid A-star algorithm. The RS paths generated by bigger curvatures in Figures 8d, 9d and 10d are closer to obstacles than those generated by smaller curvatures in Figures 8e, 9e and 10e. On the other hand, the lengths of the RS paths in the improved method are not always shorter than those in the original method when safety is preferentially considered (there are no significant differences in lengths). This result is consistent with the properties of RS curves, as presented in Section 3. However, the computation time is a drawback of this method, and more time is required to compute RS paths as curvature resolution increases. The reason is that the original method produces only RS curves using extreme curvature values. Whereas the proposed method needs to generate multiple RS curves in curvature range with some step size. Therefore, the computation time increases at most linearly proportional to the step size of the curvature. And by the fine-tuning, we reduce the motion primitives, which means the discretization of x and y axes increases, respectively, so that the computation time increases proportionally. Another noticeable point is the number of turning points reduced from 12 to 10, 12 to 9, and 6 to 5 after tunings in the three maps (see Figures 8b,c, 9b,c and 10b,c). These paths nearly do not have unnecessary turning points and keep a comfortable distance from obstacles. In summary, in terms of safety criteria, the results indicate that the improved method can reduce the risk of collision and select a safer path to the destination.

Table 2. Comparison of original and improved methods in benchmark maps.

Maps	Turning Points (before)	Original RS Path				Improved RS Path				Turning Points (after)
		Curvature	Cost	Length (m)	Time (s)	Curvature	Cost	Length (m)	Time (s)	
map12	13	0.23	6.17	26.29	0.014	0.15	4.69	23.10	0.096	10
den520d	14	0.23	5.03	27.40	0.020	0.15	4.03	29.43	0.140	9
ost003d	8	0.23	3.89	24.51	0.018	0.10	3.27	25.26	0.166	6

5. Conclusions

In this paper, we presented an improved path-planning method for autonomous car-like robots. The method was implemented in the analytic expansion phase of a hybrid A-star path planner to improve its performance in terms of safety. Based on the properties of the RS method, different shortest RS paths were generated based on curvature values. Then, an objective function was formulated to evaluate each candidate path and select the best one that minimizes the cost risk of collision. In addition, an effective tuning in the forward search phase was described to reduce unnecessary turning points without using complicated methods. As proved through intensive tests in simulations, the path planner produced the routes that are smoother and have at least equal to or lower risks of collisions (around 20%) with surrounding obstacles compared to the routes that are produced by the conventional hybrid A-star method. Besides, the number of turning points in the improved route is also reduced by up to around 20%. A drawback is that the computation time of the proposed analytic expansion increases less than about ten times.

Having a smoother route with a lower risk of collision helps the mobile robot vehicle to move more safely while working with humans. The results of this work indicate the potential applications of hybrid A-star methods in the indoor environment that require autonomous driving tasks. However, the proposed method needs to be verified in real environments, which we will investigate in our future work.

Author Contributions: Funding acquisition, S.C.L.; Investigation, C.V.D.; Methodology, C.V.D.; Software, C.V.D.; Validation, D.S.L.; Writing—original draft, C.V.D.; Writing—review & editing, H.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the DGIST R&D program 2022010290 (22-IT-03) and 2022010348 (22-DPIC-13) of the Ministry of Science and ICT. This work was also partially supported by the DGIST Education Innovation Grant 2022010262.

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

References

1. Tsardoulias, E.G.; Iliakopoulou, A.; Kargakos, A.; Petrou, L. A Review of Global Path Planning Methods for Occupancy Grid Maps Regardless of Obstacle Density. *Int. J. Intell. Robot. Syst.* **2016**, *84*, 829–858. [CrossRef]
2. Karur, K.; Sharma, N.; Dharmatti, C.; Siegel, J.E. A Survey of Path Planning Algorithms for Mobile Robots. *Vehicles* **2021**, *3*, 448–468. [CrossRef]
3. Sánchez-Ibáñez, J.R.; Pérez-Del-Pulgar, C.J.; García-Cerezo, A. Path Planning for Autonomous Mobile Robots: A Review. *Sensors* **2021**, *21*, 7898. [CrossRef] [PubMed]
4. Fadzli, S.A.; Abdulkadir, S.I.; Makhtar, M.; Jamal, A.A. Robotic Indoor Path Planning Using Dijkstra's Algorithm with Multi-Layer Dictionaries. In Proceedings of the International Conference on Information Science and Security (ICISS), Seoul, Korea, 14–16 December 2015; pp. 1–4
5. Yang, F.; Chakraborty, N. Chance Constrained Simultaneous Path Planning and Task Assignment for Multiple Robots with Stochastic Path Costs. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May 2020–31 August 2020; pp. 6661–6667.

6. Kim, K.; Chen, L.H.; Cera, B.; Daly, M.; Zhu, E.; Despois, J.; Agogino, A.K.; SunSpiral, V.; Agogino, A.M. Hopping and rolling locomotion with spherical tensegrity robots. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 4369–4376.
7. Zhang, J.; Wang, W.; Qi, X.; Liao, Z. Social and Robust Navigation for Indoor Robots Based on Object Semantic Grid and Topological Map. *Appl. Sci.* **2020**, *10*, 8991. [CrossRef]
8. Wei, J.; Liu, J. Mobile robot path planning with η3-splines using spatial-fitness-sharing variable-length genetic algorithm. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, 10–15 October 2009; pp. 978–983.
9. Solak, S.; Yakut, Ö.; Bolat, E.D. Design and Implementation of Web-Based Virtual Mobile Robot Laboratory for Engineering Education. *Symmetry* **2020**, *12*, 906. [CrossRef]
10. Molina-Leal, A.; Gómez-Espinosa, A.; Cabello, J.A.E.; Cuan-Urquizo, E.; Cruz-Ramírez, S.R. Trajectory Planning for a Mobile Robot in a Dynamic Environment Using an LSTM Neural Network. *Appl. Sci.* **2021**, *11*, 10689. [CrossRef]
11. 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]
12. Sun, X.; Deng, S.; Zhao, T.; Tong, B. Motion planning approach for car-like robots in unstructured scenario. *Trans. Inst. Meas. Control* **2022**, *44*, 754–765. [CrossRef]
13. Ravankar, A.; Ravankar, A.A.; Kobayashi, Y.; Hoshino, Y.; Peng, C.C. Path Smoothing Techniques in Robot Navigation: State-of-the-Art, Current and Future Challenges. *Sensors* **2018**, *18*, 3170. [CrossRef] [PubMed]
14. Mazzalai, A.; Biral, F.; Lio, M.D.; Darin, M.; D'Orazio, L. Automated Crossing of Intersections Controlled by Traffic Lights. In Proceedings of the IEEE 18th International Conference on Intelligent Transportation Systems, Gran Canaria, Spain, 15–18 September 2015; pp. 1928–1933.
15. Zhan, W.; Chen, J.; Chan, C.-Y.; Liu, C.; Tomizuka, M. Spatially-partitioned environmental representation and planning architecture for on-road autonomous driving. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, 11–14 June 2017; pp. 632–639.
16. Gutjahr, B.; Gröll, L.; Werling, M. Lateral Vehicle Trajectory Optimization Using Constrained Linear Time-Varying MPC. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 1586–1595. [CrossRef]
17. Zhou, X.; Zhu, J.; Zhou, H.; Xu, C.; Gao, F. EGO-Swarm: A Fully Autonomous and Decentralized Quadrotor Swarm System in Cluttered Environments. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 4101–4107.
18. Mane, S.B.; Vhanale, S. Genetic Algorithm Approach for Obstacle Avoidance and Path Optimization of Mobile Robot. *Adv. Intell. Syst. Comput.* **2019**, *810*, 649–659.
19. Montemerlo, M.; Becker, J.; Bhat, S.; Dahlkamp, H.; Dolgov, D.; Ettinger, S.; Haehnel, D.; Hilden, T.; Hoffmann, G.; Huhnke, B.; et al. Junior: The Stanford Entry in the Urban Challenge. In *The DARPA Urban Challenge*; Buehler, M., Iagnemma, K., Singh, S., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; pp. 91–123.
20. Petereit, J.; Emter, T.; Frey, C.W.; Kopfstedt, T.; Beutel, A. Application of Hybrid A* to an Autonomous Mobile Robot for Path Planning in Unstructured Outdoor Environments. In Proceedings of the ROBOTIK 2012; 7th German Conference on Robotics, Munich, Germany, 21–22 May 2012; pp. 1–6.
21. Sedighi, S.; Nguyen, D.V.; Kuhnert, K.D. Guided Hybrid A-star Path Planning Algorithm for Valet Parking Applications. In Proceedings of the International Conference on Control, Automation and Robotics (ICCAR), Beijing, China, 19–22 April 2019; pp. 570–575.
22. Tang, B.; Hirota, K.; Wu, X.; Dai, Y.; Jia, Z. Path Planning Based on Improved Hybrid A-star Algorithm. *J. Adv. Comput. Intell. Intell. Inform.* **2021**, *25*, 64–72 [CrossRef]
23. Dolgov, D.; Thrun, S.; Montemerlo, M.; Diebel, J. Practical search techniques in path planning for autonomous driving. *Am. Assoc. Artif. Intell.* **2008**, *1001*, 18–80.
24. Dolgov, D.; Thrun, S.; Montemerlo, M.; Diebel, J. Path planning for autonomous vehicles in unknown semi-structured environments. *Int. J. Robot. Res.* **2010**, *29*, 485–501. [CrossRef]
25. Heuristic. Available online: http://theory.stanford.edu/~amitp/GameProgramming/Heuristics.html (accessed on 2 June 2022).
26. A-Star Search Algorithm. Available online: https://isaaccomputerscience.org/concepts/dsa_search_a_star?examBoard=all&stage=all (accessed on 2 June 2022).
27. LaValle, S.M. *Planning Algorithms*; Cambridge University Press: Cambridge, UK, 2006.
28. 2D Pathfinding Benchmarks. Available online: https://www.movingai.com/benchmarks/grids.html (accessed on 25 April 2022).
29. PathBench: Benchmarking Platform for Classic and Learned Path Planning Algorithms. Available online: https://github.com/perfectly-balanced/PathBench (accessed on 25 April 2022).