2023 IEEE 3rd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA) | 978-1-6654-9079-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICIBA56860.2023.10165386

Hybrid path planning algorithm for mobile robot based on A* algorithm fused with DWA

Leng Han
Chongqing University
of Posts and Telecommunications
Chongqing, China
Email: hanleng@cqupt.edu.cn

Xuhui Wu
Chongqing University
of Posts and Telecommunications
Chongqing, China
Email: wuxuhui95@gmail.com

Xia Sun*
Chongqing Institute of Engineering
Chongqing, China
Email: sunxia@cqie.edu.cn

***Abstract*—Mobile robots operating in unknown environments and require collision-free real-time path planning. In this paper, a hybrid path planning algorithm is proposed to solve the navigation problem of mobile robots in the presence of unpredictable obstacles. First, to improve the global algorithm's planning efficiency, the heuristic function of the A* algorithm is improved. In addition, this paper forms a hybrid path-planning algorithm by fusing the DWA algorithm. The local path planning is carried out with the intersection of the scope of the local path planning and the global path as the intermediate target point. The trajectory similarity evaluation coefficient is added to the heuristic function. When the original global path needs to be re-opened due to unknown obstacles, the local planning function is then fed back to the improved A* algorithm for dynamic path re-planning. The simulation results show that the hybrid path algorithm can avoid dynamic environmental obstacles. At the same time, when encountering unknown obstacles, it adopts local planning and adjusts the global path to meet the navigation requirements of mobile robots in complex environments.**

***Index Terms*—Mobile robot, Algorithm optimization, Heuristic function, Hybrid path planning.**

I. Introduction

Mobile robots face unfamiliar environments mainly to achieve self-localization and self-map, and the second is to complete self-navigation tasks[1]. The main content of robot navigation is to complete path planning. When the target point is obtained, the robot must quickly plan a safe and effective path with the shortest distance[2]. With the increasing complexity of the offline environment and the emergence of unpredictable obstacles, mobile robots need to cooperate with local path planning to complete the navigation task[3].

Regarding path planning algorithms, scholars at home and abroad have conducted much research and achieved many excellent results. The main path-planning algorithms are divided into two categories[4]. Algorithms used for local path planning include the DWA algorithm[5], algorithms used for global path planning include the A* algorithm[6], fast expanding random tree Algorithm[7], etc.

With the increase of unknown obstacles in the environment, this algorithm cannot adjust the local path in time, and there is a problem that the mobile robot does not run smoothly. Reference modified the heuristic function of the A* algorithm, assigning different gray values to each map grid to calculate the path point. The reference modified the free space relationship between obstacles and robots to improve the DWA algorithm, limiting obstacles. The minimum distance between the object and the robot makes the mobile robot unable to face the narrow corridor environment. A single improvement was made to the DWA algorithm in [8], which enables the improved algorithm to perceive densely distributed obstacles. Once the improved DWA algorithm performs local obstacle avoidance many times, the planned local path deviates from the global path. The optimality of the global path will no longer exist. To sum up, it is necessary to integrate global and local path planning algorithms and adjust the global path in time to deal with the status quo.

The rest work is organized as follows. The second part introduces the map model, and the third part gives the process of A* algorithm optimization. The fourth section improves the DWA algorithm according to the characteristics of local path planning, fuses it with the improved A* algorithm, and proposes a fusion path planning algorithm. The hybrid algorithm is simulated in the fifth section. Finally, a brief conclusion is given in the sixth section.

II. Environment model

Environment modeling is the essential step in path planning. This paper uses a grid map as an environment model for simulating robot motion. The grid method divides the simulation space into continuous non-intersecting squares according to a certain granularity, and these squares form a grid. The state of each grid represents the environmental information of the current location. Assuming that the environment used to simulate the motion of the robot consists of $m \times n$ grids of length a, the size of the environment composed of all grids is $x \times y$, The position corresponding to the i-th grid is represented as follows:

$$\begin{cases} x_i = a \times [\bmod(i, y) - a/2] \\ y_i = a \times [x + a/2 - \mathrm{ceil}(i/x)] \end{cases} \tag{1}$$

The two sides of the grid map are used to establish coordinate axes, the horizontal direction is regarded as the x-axis, and the vertical direction is regarded as the y-axis. Then the coordinates of each grid can be marked with (x, y). The value

*Xia Sun is the corresponding author(Email: sunxia@cqie.edu.cn)

of each grid is represented by 0 or 1, with one indicating that the grid is occupied and 0 indicating that the grid is idle. Think of the robot's model as a point on the grid map, with the black grid representing environmental obstacles. The representation of the raster map is displayed as shown below:

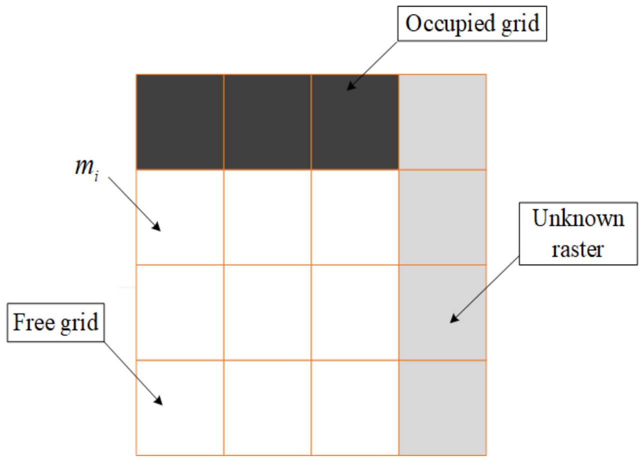

Fig. 1: Schematic diagram of grid utilization

III. Improved A* Algorithm Design

Based on the traditional A* algorithm, the adaptive exponential coefficient is added as the weight of the heuristic function. The weight of the heuristic function is flexibly allocated according to the current environment of the mobile robot, which reduces the traversal of invalid nodes. The improved algorithm obtains all path nodes and connects all nodes to form a global path, and then smoothes the path.

A. Improved heuristic function

In a large-scale environment, as the distance between the starting position and the target point increases, the mobile robot must traverse more nodes when planning the path in the built grid map. The estimated cost value $h(n)$ of the current node n obtained by the traditional A* algorithm may be much smaller than the distance between the current node and the target point. At this time, the search process of the A* algorithm is biased towards the Dijkstra algorithm, and its process traverses many invalid nodes, reducing the efficiency of path planning.

In order to solve the above problems, this paper proposes the adaptive exponential coefficient as the weight of the heuristic function. The adaptive index coefficient integrates the ratio of the number of obstacles around the current node to the total number of obstacles. The calculation of the estimated cost of the parent node of the current node is added to reduce the probability of the node traversing to both sides during the algorithm search process. Finally, the angle information is added while calculating the current nodes estimated cost, making the planned road more directional. The heuristic function of the improved A* algorithm is as follows:

$$h_A(n) = e^{P/Q}(h(n) + h(p) - k\cos\beta) \tag{2}$$

In Equation (2), Define the number of grids between the starting position of the robot and the target point as Q. The number of grid obstacles in the search range from the current node to the target is P. The exponential coefficient $e^{P/Q}$ is the weight coefficient of the heuristic function of the improved A* algorithm. In the process of path planning, the weight value of the estimated cost can be dynamically allocated with the change of the number of obstacles around the current node, which realizes the adaptability of the calculation of the total cost value of the node. Furthermore, $h(n)$ and $h(p)$ are the estimated cost of the current node and the estimated cost of the parent node of the current node, respectively. The direction angle β is defined as the angle between the line between the current node and the starting node and the line between the current node and the target point. The connection between the current node, the target point, and the size range is $[0°, 180°]$. k is the weight coefficient of the cosine of the included angle, which is determined by the range of the environment. Therefore, the calculation formula for the total cost of the improved A* algorithm is as follows.

$$f(n) = g(n) + e^{P/Q}(h(n) + h(p) - k\cos\beta) \tag{3}$$

B. Experiment

The optimized A* algorithm considers the obstacle information of different positions in the environment and the direction information of the current node. In order to verify the feasibility of the improved A* algorithm path planning, the following simulation experiments are carried out. The planning efficiency of the improved A* algorithm is verified in a grid map with a size of 40×40. The simulation software used in this paper is MATLAB R2021, The computer is Windows 10 64-bit operating system, and The processor information is Inter®Corei5-8300H (This device is used in subsequent experiments in this paper). The results are shown in Figure 2 and Figure 3, respectively.

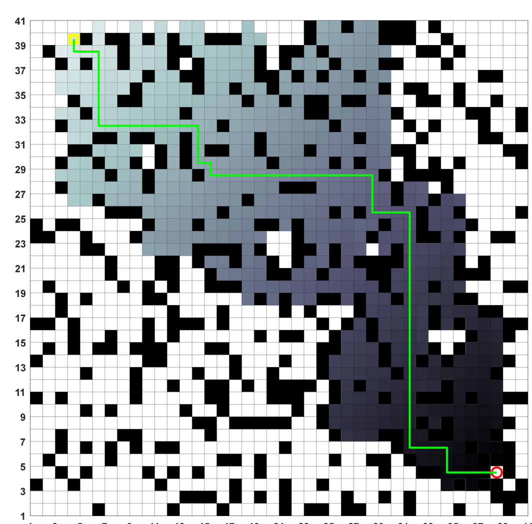

Fig. 2: Simulation results of traditional A* algorithm

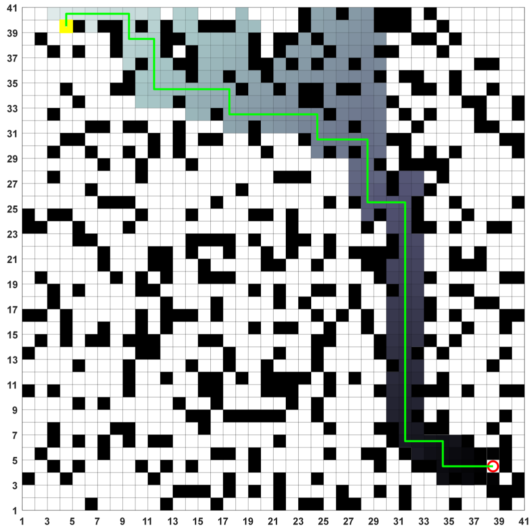

Fig. 3: Simulation results of improved A* algorithm

TABLE I: Simulation results

Simulation results	Search node	Search length	Planning times(ms)
Traditional A*	765	71	131
Improved A*	382	69	40

The improved A* algorithm can plan an optimal path between the starting node and the target node, and fewer nodes are traversed during the search process. The total path length, planning time, and the number of traversed nodes are used as indicators to qualitatively analyze the algorithm's before and after improvement. The improved A* algorithm reduces the traversal of redundant nodes, shortens the path planning time, and dramatically improves the algorithm's efficiency. The parameters of the two paths are shown in Table I.

IV. Optimization of DWA algorithm

The improvement of the execution efficiency and the smooth path of the A* algorithm make the global path planning system of the mobile robot more perfect. At this time, the fusion DWA (Dynamic Window Algorithm) algorithm enables the robot to avoid dynamic obstacles when running on the planned global path and then complete the navigation task safely and efficiently.

A. Mobile Robot Kinematics Modeling

The DWA algorithm selects the speed at the next moment based on the prediction of the trajectory of the mobile robot. The motion model used to control the trajectory of the mobile robot is very important. In this paper, the mobile navigation robot independently developed by the laboratory is selected as the experimental platform for experimental testing. The driving wheels of the mobile robot are two differentially driven wheels, the kinematic model of which is shown in Figure 4.

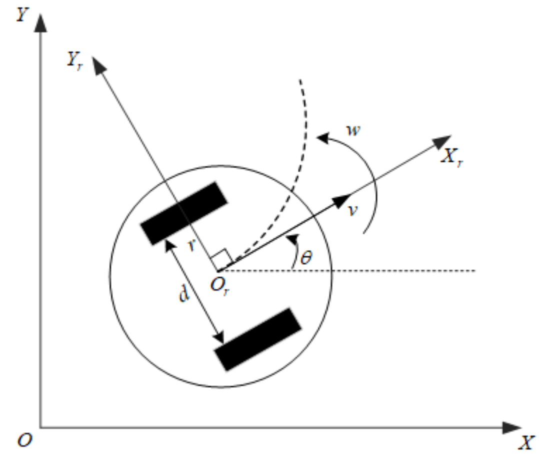

Fig. 4: Differential drive robot kinematic model

Mobile robots can move forward and steer during movement. The speed of moving the robot in unit time is (v_t, ω_t). When the unit time Δt is short, the satisfying arc of the mobile robot can be approximated as a straight line to calculate the running speed from the current time to the next unit time. The mathematical expression of the kinematic model is expressed as follows:

$$\begin{cases} x(t) = x(x-1) + v(t)\Delta t \cos(\theta(t-1)) \\ y(t) = y(t-1) + v(t)\Delta t \sin(\theta(t-1)) \\ \theta(t) = \theta(t-1) + \omega(t)\Delta t \end{cases} \tag{4}$$

B. Optimization of DWA algorithm

There are infinitely many velocities in a cluster of velocity sample space (v_t, ω_t). The sampling speed can be controlled within a specific range according to the robot and environmental constraints. The limit speed of the mobile robot itself is as follows:

$$V_m = \{v \in [v_{\min}, v_{\max}], \omega \in [\omega_{\min}, \omega_{\max}]\} \tag{5}$$

Due to the limited torque of the mobile robot motor, there is a maximum acceleration and deceleration limit. In the simulation cycle of the trajectory of the mobile robot, there is a dynamic window, and the speed value in this window is the actual speed V_d that the robot can achieve during operation. Among them, (v_c, ω_c) is the current speed of the robot, and other signs correspond to the maximum acceleration and deceleration.

$$V_d = \left\{ (v, \omega) \mid \begin{array}{c} v \in [v_c - \dot{v}_b \Delta t, v_c + \dot{v}_a \Delta t] \wedge \\ \omega \in [\omega_c - \dot{\omega}_b \Delta t, \omega_c + \dot{\omega}_a \Delta t] \end{array} \right\} \tag{6}$$

For the mobile machine to obtain a suitable deceleration distance under the condition of maximum deceleration, the speed has a range:

$$\begin{aligned} V_a =&((v, \omega) \mid v \leq \sqrt{2 \cdot dist(v, \omega) \cdot \dot{v}_b} \\ &\wedge \omega \leq \sqrt{2 \cdot dist(v, \omega) \cdot \dot{\omega}_b}) \end{aligned} \tag{7}$$

Among them, $dist(v, \omega)$ is the closest distance to the obstacle on the trajectory corresponding to the velocity, and v_a is the sampling speed that can maintain this distance. The closest distance to the obstacle in this speed range is shown in Figure 5.

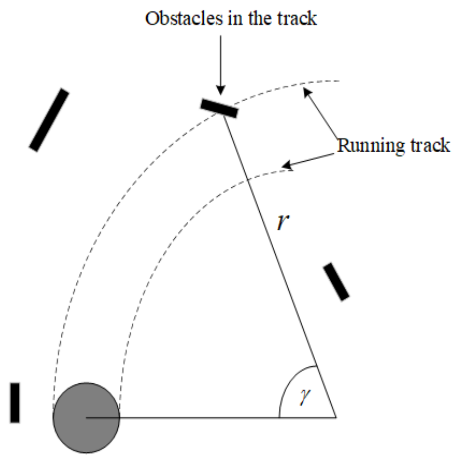

Fig. 5: Schematic diagram of the maximum deceleration of the robot

The heading angle, linear velocity, and the shortest distance between the end of the simulated trajectory and the obstacle of the mobile robot are used as the evaluation function indicators of the DWA algorithm. Since the measurement units of these three indicators are different, to avoid the value of a certain quantity being too large and the proportion being too large, it is necessary to normalize each indicator. The trajectory evaluation function obtained after normalization is as follows:

$$G(v,\omega) = \alpha \cdot heading(v,\omega) + \beta \cdot dist(v,\omega) + \gamma \cdot vel(v,\omega) \quad (8)$$

In Equation (8), α, β, γ are the three weight coefficients, and the sum of the three is 1. *heading* represents the angle between the heading and the target point. *dist* represents the distance between the endpoint of a curve and the nearest obstacle. *vel* represents the velocity of the current curve. In order to ensure that the trajectory simulated by the robot in the current speed space does not deviate from the global path, the trajectory similarity index *simi* is added to the above trajectory evaluation function Its weight coefficient is set to σ. Then the improved evaluation function is as follows:

For the traditional DWA algorithm to solve the optimal trajectory, the most significant factor is the efficiency control of the DWA algorithm. If the efficiency is too high, a locally optimal solution may appear. On the contrary, although it can ensure the safe operation of the robot between obstacles, it will also reduce the robot's sensitivity for dynamic obstacle avoidance.

V. Hybrid Path Planning System

The improved A* algorithm improves the efficiency of global path planning, considering the starting point and static obstacles, but still cannot avoid unknown obstacles appearing in the environment. The DWA algorithm can select the optimal local trajectory, and it is easy to fall into the local optimal due to the limitation of the nearest target point. Therefore, consider combining the two. The execution steps of the hybrid path planning algorithm are as follows.

Step 1 : The hybrid path planning system mainly includes two parts: global planning and local planning. First, the global planner uses the improved A* algorithm to obtain the global path of the robot. The system plan is shown in Figure 8.

Step 2: The intersection of the local map range and the current global path in the forward simulation trajectory time of the DWA algorithm is used as the intermediate target point. Under the local path planner, the DWA algorithm is used to plan the local path with the intermediate target point as the temporary end point.

Step 3: In the local path planner, the DWA algorithm completes the avoidance of dynamic obstacles in the environment. Finally, by continuously updating the intermediate target point until reaching the global target point.

Step 4: When the DWA planner plans a local path, avoiding unknown obstacles on the current global path may cause the local way to deviate from the global direction. The A* algorithm is now improved to perform dynamic path re-planning.

In order to verify the feasibility and superiority of the hybrid path planning algorithm, a simulation experiment was carried out on the hybrid algorithm. The simulation scene is a 20×20 grid map obtained after modeling the natural environment. Dynamic obstacles and unknown static obstacles are added to the trajectory planned by the improved A* algorithm.The starting point coordinates are set as $S(1.5, 1.5)$, and the target point coordinates are set as $G(20.5, 20.5)$. The various coefficients of the evaluation function of the improved DWA algorithm are $\alpha = 0.5$, $\beta = 0.2$, $\gamma = 0.2$, $\sigma = 0.1$. The robot kinematic parameters of the hybrid path planning algorithm are shown in Table II.

TABLE II: Robot kinematic parameters

Kinematic parameters	Numerical value
Maximum line speed	$1.5m/s$
maximum linear acceleration	$0.2m/s^2$
speed resolution	$0.01m/s^2$
maximum angular velocity	$0.5rad/s$
maximum angular acceleration	$0.3rad/s^2$
Angular velocity resolution	$1^\circ/s$

In Figure 6, gray grids represent unknown static obstacles added to the map. Add a yellow dynamic obstacle. The blue trajectory is the global path obtained by the improved A* algorithm. From this path, it can be seen that the improved A* algorithm can find the optimal path but cannot find unknown obstacles in the map, and the path has many turning points.

The final result of the hybrid algorithm is shown in Figure 7. Local dynamic planning is performed when the mobile robot encounters the first static and dynamic obstacles. Global path replanning is performed before the second unknown obstacle.

The gray dotted line on the way is the trajectory of the moving obstacle. The blue dotted line is the path trajectory the improved A* algorithm planned. The solid red line is the final trajectory planned and smoothed by the hybrid programming algorithm.

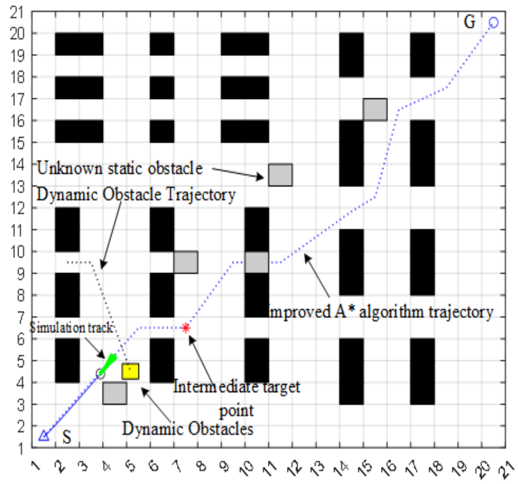

Fig. 6: Improved A* algorithm simulation

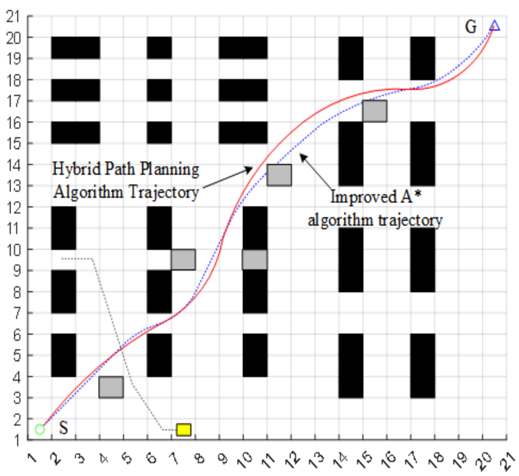

Fig. 7: Simulation of Hybrid Path Planning Algorithm

TABLE III: Comparison of Results between Improved A* Algorithm and Hybrid Path Algorithm

Algorithm	Path length	Number of turning points
Improved A* algorithm	33	7
Hybrid Path Planning Algorithm	29	4

Compared with the improved A* algorithm, the hybrid algorithm reduces the final path length by 13% and the path turning points by 43%. The results are shown in Table III. The hybrid planning algorithm completes the avoidance of dynamic obstacles in the environment. When encountering unknown obstacles, adopt local planning and adjust the global path so that the navigation process of the robot is safe and stable.

VI. Conclusion

This article proposes a hybrid path-planning algorithm. Through the fusion of the improved A* and the improved DWA algorithm, the algorithm obtains the initial optimal path between the starting point and the target point. Combined with the intersection of the global path and the local map as the intermediate target of local planning, global navigation is planned step by step. Through simulation and comparison experiments, the final hybrid path planning algorithm overcomes the problem that dynamic obstacles cannot be avoided and unknown obstacles cannot be distinguished when only global path planning is used. It avoids the problem that the DWA algorithm is easy to fall into the local optimum, and the obtained global path is always processed smoothly so that the hybrid algorithm system can meet the needs of the mobile robot to run smoothly in a complex environment.

VII. Acknowledgments

This work was supported in part by the Science and Technology Research Project of the Chongqing Education Commission under Grant KJZD-M202001901, in part by the General Projects of Chongqing Science and Technology Commission under Grant cstc2020jcyj-msxmX0895, and in part by the Scientific research project of Chongqing Institute of Engineering under Grant 2021xzky02.

References

[1] N. Sariff and N. Buniyamin, ”An Overview of Autonomous Mobile Robot Path Planning Algorithms,” 2006 4th Student Conference on Research and Development, Shah Alam, Malaysia, 2006, pp. 183-188.

[2] Song, Baoye, Zidong Wang, and Lei Zou. ”On global smooth path planning for mobile robots using a novel multimodal delayed PSO algorithm.” Cognitive Computation 9, no. 1 (2017): 5-17.

[3] Guo, Jianming, Liang Liu, Qing Liu, and Yongyu Qu. ”An improvement of D* algorithm for mobile robot path planning in partial unknown environment.” In 2009 Second International Conference on Intelligent Computation Technology and Automation, vol. 3, pp. 394-397. IEEE, 2009.

[4] Gasparetto, Alessandro, Paolo Boscariol, Albano Lanzutti, and Renato Vidoni. ”Path planning and trajectory planning algorithms: A general overview.” Motion and Operation Planning of Robotic Systems: Background and Practical Approaches (2015): 3-27.

[5] Dong, Yujing, Shuang Liu, Changzheng Zhang, and Qigao Zhou. ”Path Planning Research for Outdoor Mobile Robot.” In 2022 12th International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER), pp. 543-547. IEEE, 2022.

[6] Ren, Bin, Jianwei Liu, and Jiayu Chen. ”Simulating human-machine coupled model for gait trajectory optimization of the lower limb exoskeleton system based on genetic algorithm.” International Journal of Advanced Robotic Systems 17, no. 1 (2020): 1729881419893493.

[7] Chen, Tao, Suifan Chen, Kuoran Zhang, Guoting Qiu, Qipeng Li, and Xinmin Chen. ”A jump point search improved ant colony hybrid optimization algorithm for path planning of mobile robot.” International Journal of Advanced Robotic Systems 19, no. 5 (2022): 17298806221127953.

[8] Yu, Junwei, Jing Hou, and Guang Chen. ”Improved safety-first A-star algorithm for autonomous vehicles.” In 2020 5th International Conference on Advanced Robotics and Mechatronics (ICARM), pp. 706-710. IEEE, 2020.